Exynos 4412 SCP

RISC Microprocessor

Revision 0.10

April 2012

User's Manual

SAMSUNG ELECTRONICS

SAMSUNG

Important Notice

Samsung Electronics Co. Ltd. (“Samsung”) reserves the right to make changes to the information in this publication at any time without prior notice. All information provided is for reference purpose only. Samsung assumes no responsibility for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

This publication on its own does not convey any license, either express or implied, relating to any Samsung and/or third-party products, under the intellectual property rights of Samsung and/or any third parties.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

Customers are responsible for their own products and applications. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by the customer's technical experts.

Samsung products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could reasonably be expected to create a situation where personal injury or death may occur. Customers acknowledge and agree that they are solely responsible to meet all other legal and regulatory requirements regarding their applications using Samsung products notwithstanding any information provided in this publication. Customer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim (including but not limited to personal injury or death) that may be associated with such unintended, unauthorized and/or illegal use.

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu
Yongin-City, Gyeonggi-Do, Korea 446-711

Contact Us: *mobilesol.cs@samsung.com*
Home Page: *http://www.samsungsemi.com*

Trademarks

All brand names, trademarks and registered trademarks belong to their respective owners.

- Exynos, Exynos4210, FlexOneNAND, and OneNAND are trademarks of Samsung Electronics.
- ARM, Jazelle, TrustZone, and Thumb are registered trademarks of ARM Limited. Cortex, ETM, ETB, Coresight, ISA, and Neon are trademarks of ARM Limited.
- Java is a trademark of Sun Microsystems, Inc.
- SD is a registered trademark of Toshiba Corporation.
- MMC and eMMC are trademarks of MultiMediaCard Association.
- JTAG is a registered trademark of JTAG Technologies, Inc.
- Synopsys is a registered trademark of Synopsys, Inc.
- I2S is a trademark of Phillips Electronics.
- I2C is a trademark of Phillips Semiconductor Corp.
- MIPI and Slimbus are registered trademarks of the Mobile Industry Processor Interface (MIPI) Alliance.

All other trademarks used in this publication are the property of their respective owners.

Chip Handling Guide

Precaution against Electrostatic Discharge

When handling semiconductor devices, be sure that the environment is protected against static electricity.

1. Operators should wear anti-static clothing and use earth band.
2. All objects that come in direct contact with devices should be made of materials that do not produce static electricity that would cause damage.
3. Equipment and work table must be earthed.
4. Ionizer is recommended to remove electron charge.

Contamination

Be sure to use semiconductor products in the environment that may not be exposed to dust or dirt adhesion.

Temperature/Humidity

Semiconductor devices are sensitive to environment temperature and humidity. High temperature or humidity may deteriorate semiconductor device's characteristics. Therefore avoid storage or use in such conditions.

Mechanical Shock

Care should be exercised not to apply excessive mechanical shock or force on semiconductor device.

Chemical

Do not expose semiconductor device to chemical because reaction to chemical may cause deterioration of device characteristics.

Light Protection

In case of non-EMC (Epoxy Molding Compound) package, do not expose semiconductor IC to strong light. It may cause device's malfunction. (But, some special products which utilize the light or have security function are excepted from this guide)

Radioactive, Cosmic and X-ray

Semiconductor devices can be influenced by radioactive, cosmic ray or X-ray. Radioactive, cosmic and X-ray may cause soft error during device operation. Therefore semiconductor devices must be shielded under environment that may be exposed to radioactive, cosmic ray or X-ray.

EMS (Electromagnetic Susceptibility)

Note that semiconductor device's characteristics may be affected by strong electromagnetic wave or magnetic field during operation under insufficient PCB circuit design for EMS.

Revision History

Revision No.	Date	Description	Author(s)
0.10	April. 30, 2012	• First Draft	Chongkun Lee

Table of Contents

1 PRODUCT OVERVIEW 1-1

1.1 Overview 1-1
1.2 Block Diagram of Exynos 4412 SCP 1-2
1.3 Features of Exynos 4412 SCP 1-3
1.3.1 Multi-Core Processing Unit 1-5
1.3.2 Memory Subsystem 1-6
1.3.3 Multimedia 1-7
1.3.4 Audio Subsystem 1-10
1.3.5 Image Signal Processing Subsystem 1-10
1.3.6 Security Subsystem 1-10
1.3.7 Connectivity 1-11
1.3.8 System Peripheral 1-13
1.4 Conventions 1-15
1.4.1 Register R/W Conventions 1-15
1.4.2 Register Value Conventions 1-15

2 BALL MAP AND DESCRIPTION 2-1

2.1 Overview 2-1
2.2 Pin Map 2-2
2.2.1 Block Diagram of 786-ball FCFBGA (SCP) 2-2
2.3 Pin Assignments and Pin Number Order 2-3
2.4 I/O Control Type Conventions 2-9
2.5 Pin Description 2-10
2.5.1 Signal Pin 2-10
2.5.2 Digital IO Power Pin 2-28
2.5.3 Analog/HSI Power Pin 2-29
2.5.4 Memory Power Pin 2-30

3 MEMORY MAP 3-1

3.1 Overview 3-1
3.2 SFR Base Address 3-2

4 CHIP ID 4-1

4.1 Overview 4-1
4.2 Register Description 4-2
4.2.1 Register Map Summary 4-2

5 BOOTING SEQUENCE 5-1

5.1 Overview 5-1
5.2 Functional Description 5-1
5.2.1 Block Diagram 5-2
5.2.2 Scenario Description 5-3

6 GENERAL PURPOSE INPUT/OUTPUT (GPIO) CONTROL 6-1

6.1 Overview 6-1

6.1.1 Features of GPIO 6-3
6.1.2 Input/Output Description 6-3
6.2 Register Description 6-5
6.2.1 Registers Summary 6-5
6.2.2 Part 1 6-21
6.2.3 Part 2 6-126
6.2.4 Part 3 6-291
6.2.5 Part 4 6-300

7 CLOCK MANAGEMENT UNIT 7-1

7.1 Clock Domains 7-1
7.2 Clock Declaration 7-4
7.2.1 Clocks from Clock Pads 7-4
7.2.2 Clocks from CMU 7-5
7.3 Clock Relationship 7-6
7.3.1 Recommended PLL PMS Value for APLL and MPLL 7-8
7.3.2 Recommended PLL PMS Value for EPLL 7-9
7.3.3 Recommended PLL PMS Value for VPLL 7-10
7.4 Clock Generation 7-11
7.5 Clock Configuration Procedure 7-16
7.5.1 Clock Gating 7-17
7.5.2 Clock Diving 7-17
7.6 Special Clock Description 7-18
7.6.1 Special Clock Table 7-18
7.7 CLKOUT 7-21
7.8 I/O Description 7-24
7.9 Register Description 7-25
7.9.1 Register Map Summary 7-27

8 POWER MANAGEMENT UNIT 8-1

8.1 Overview 8-1
8.2 Background 8-2
8.3 Power Domains 8-4
8.4 Local Power Control 8-6
8.4.1 Overview 8-6
8.4.2 Sequence 8-7
8.5 System-level Power Control 8-10
8.5.1 Overview on System-level Power Control 8-10
8.5.2 Power-down Sequence 8-14
8.5.3 System-Level Power-down Configuration Registers 8-16
8.5.4 Wake-up Sources 8-20
8.6 Reset 8-22
8.6.1 Hardware Reset 8-24
8.6.2 Watchdog Reset 8-26
8.6.3 Software Reset 8-27
8.6.4 Wakeup Reset 8-27
8.7 IO Description 8-28
8.8 Register Description 8-29
8.8.1 Register Map Summary 8-29

9 INTERRUPT CONTROLLER 9-1

9.1 Overview 9-1
9.1.1 Features 9-1
9.1.2 Security Extensions Support 9-2
9.1.3 Implementation-Specific Configurable Features 9-3
9.1.4 Terminology 9-4
9.2 Interrupt Source 9-8
9.2.1 Interrupt Sources Connection 9-8
9.2.2 GIC Interrupt Table 9-9
9.3 Functional Overview 9-17
9.3.1 Functional Interfaces 9-18
9.3.2 The Distributor 9-22
9.3.3 CPU Interfaces 9-23
9.4 Interrupt Handling and Prioritization 9-24
9.4.1 About Interrupt Handling and Prioritization 9-24
9.4.2 General Handling of Interrupts 9-27
9.4.3 Interrupt Prioritization 9-36
9.4.4 The Effect of the Security Extensions on Interrupt Handling 9-39
9.4.5 The Effect of the Security Extensions on Interrupt Prioritization 9-41
9.5 Register Description 9-42
9.5.1 Register Map Summary 9-42

10 INTERRUPT COMBINER 10-1

10.1 Overview 10-1
10.1.1 Features 10-1
10.1.2 Block Diagram 10-2
10.2 Interrupt Sources 10-4
10.2.1 Interrupt Combiner 10-4
10.3 Functional Description 10-9
10.4 Register Description 10-10
10.4.1 Register Map Summary 10-10
10.4.2 Interrupt Combiner 10-11

11 DIRECT MEMORY ACCESS CONTROLLER (DMAC) 11-1

11.1 Overview of DMA Controller 11-2
11.1.1 Features of DMA Controller 11-3
11.2 Register Description 11-6
11.2.1 Register Map Summary 11-6
11.3 Instruction 11-15
11.3.1 Key Instruction 11-16
11.3.2 USAGE Model 11-18

12 SYSTEM REGISTERS 12-1

12.1 Overview 12-1
12.2 Register Description 12-2
12.2.1 Register Map Summary 12-2

13 CORESIGHT 13-1

13.1 Overview 13-1
13.2 Components of CoreSight 13-3
13.2.1 This Section Includes: 13-3
13.2.2 Debug Access Port (DAP) 13-3

13.2.3 Single Wire and JTAG Debug Port (SWJ-DP) 13-4
13.2.4 APB Multiplexer 13-4
13.2.5 Trace Funnel 13-5
13.2.6 Cross Trigger Interface (CTI) and Cross Trigger Matrix (CTM) 13-5
13.2.7 ROM Table and Multi-Drop ID 13-6
13.3 Clock Description 13-9
13.3.1 PCLKENDBG Generation 13-10
13.4 Reset 13-11
13.5 Power 13-12
13.6 I/O Description 13-13

14 TRUSTZONE PROTECTION CONTROLLER (TZPC) 14-1

14.1 Overview 14-1
14.1.1 Features of TZPC 14-1
14.1.2 Block Diagram 14-1
14.2 Functional Description 14-2
14.3 TZPC Configuration 14-4
14.4 Register Description 14-7
14.4.1 Registers Map Summary 14-7
14.4.2 TZPC0 Registers 14-13
14.4.3 TZPC1 Registers 14-17
14.4.4 TZPC2 Registers 14-21
14.4.5 TZPC3 Registers 14-25
14.4.6 TZPC4 Registers 14-29
14.4.7 TZPC5 Registers 14-33

15 TRUSTZONE ADDRESS ACCESS CONTROLLER (TZASC) 15-1

15.1 Overview 15-1
15.1.1 Features of the TZASC 15-1
15.1.2 Block Diagram 15-1
15.2 Functional Description 15-2
15.2.1 Functional Interfaces 15-2
15.2.2 Functional Operation 15-6
15.2.3 Constraints of Use 15-16
15.3 Register Description 15-17
15.3.1 Registers Map Summary 15-17

16 SYSTEM MEMORY MANAGEMENT UNIT 16-1

16.1 Introduction 16-1
16.1.1 Key Features 16-1
16.1.2 Functional Description 16-2
16.1.3 Functional Overview 16-2
16.1.4 Address Translation 16-4
16.1.5 Page Table Walk 16-6
16.2 Configuration and Programming View 16-8
16.2.1 Disable Mode 16-8
16.2.2 Enable Mode 16-8
16.2.3 Block Mode 16-8
16.2.4 TLB Replacement Policy 16-8
16.2.5 TLB Flush 16-8
16.2.6 Fault Handling 16-9

16.2.7 Miss Rate Measurement 16-11
16.2.8 Register Descriptions 16-13
16.2.9 Register Map Summary 16-13

17 SYSTEM MEMORY MANAGEMENT UNIT 17-1

17.1 Overview 17-1
17.1.1 Features 17-1
17.2 Functional Description 17-2
17.2.1 Functional Overview 17-3
17.2.2 Address Translation 17-5
17.2.3 Page Table Walk 17-8
17.2.4 L2-TLB Access Unit 17-10
17.3 Configuration and Programming View 17-11
17.3.1 Disable Mode 17-11
17.3.2 Enable Mode 17-11
17.3.3 Block Mode 17-11
17.3.4 L1-TLB Replacement Policy 17-11
17.3.5 TLB Flush 17-12
17.3.6 Fault Handling 17-12
17.3.7 Access Protection Fault 17-12
17.3.8 Security Protection Fault 17-12
17.3.9 L1-TLB Multi-hit Fault 17-13
17.3.10 Page Fault 17-13
17.3.11 Bus Error 17-13
17.3.12 Miss Rate Measurement 17-14
17.4 Register Descriptions 17-16
17.4.1 Register Map Summary 17-16

18 DYNAMIC MEMORY CONTROLLER 18-1

18.1 Overview 18-1
18.1.1 Features 18-1
18.2 Block Diagram 18-2
18.3 Initialization 18-4
18.3.1 DDR3 18-4
18.4 Address Mapping 18-6
18.4.1 Linear Mapping 18-6
18.4.2 Interleaved Mapping 18-7
18.5 Low-Power Operation 18-8
18.5.1 AXI Low-Power Channel 18-8
18.5.2 Dynamic Power Down 18-8
18.5.3 Dynamic Self Refresh 18-8
18.5.4 Clock Stop 18-9
18.5.5 Direct Command 18-9
18.6 Precharge Policy 18-10
18.6.1 Bank Selective Precharge Policy 18-10
18.6.2 Timeout Precharge 18-11
18.7 Quality of Service 18-12
18.7.1 AXI_ID-Based QoS 18-12
18.7.2 Fast QoS 18-14
18.7.3 Default QoS 18-15
18.7.4 AxQoS-Based QoS 18-15
18.8 Read Data Capture 18-16

18.9 Memory Channels Interleaving 18-20
18.10 Performance Profiling 18-21
18.11 I/O Description 18-25
18.12 Register Description 18-27
18.12.1 Register Map Summary 18-27

19 SROM CONTROLLER 19-1

19.1 Overview 19-1
19.1.1 Features of SROMC 19-1
19.1.2 Block Diagram 19-1
19.2 Functional Description 19-2
19.2.1 nWAIT Pin Operation 19-2
19.2.2 Programmable Access Cycle 19-3
19.3 I/O Description 19-4
19.4 Register Description 19-5
19.4.1 Register Map Summary 19-5

20 NAND FLASH CONTROLLER 20-1

20.1 Overview 20-1
20.2 Features 20-1
20.2.1 Block Diagram 20-2
20.2.2 NAND Flash Memory Timing 20-3
20.3 Software Mode 20-4
20.3.1 Data Register Configuration 20-4
20.3.2 1/4/8/12/16-bit ECC 20-5
20.3.3 2048 Byte 1-bit ECC Parity Code Assignment Table 20-6
20.3.4 32 Byte 1-bit ECC Parity Code Assignment Table 20-6
20.3.5 1-bit ECC Module Features 20-6
20.3.6 1-bit ECC Programming Guide 20-7
20.3.7 4-bit ECC Programming Guide (ENCODING) 20-8
20.3.8 4-bit ECC Programming Guide (DECODING) 20-9
20.3.9 8/12/16-bit ECC Programming Guide (ENCODING) 20-10
20.3.10 8/12/16-bit ECC Programming Guide (DECODING) 20-11
20.3.11 ECC Parity Conversion Code Guide for 8/12/16-bit ECC 20-12
20.3.12 Lock Scheme for Data Protection 20-13
20.4 Programming Constraints 20-14
20.5 I/O Description 20-14
20.6 Register Description 20-15
20.6.1 Register Map Summary 20-15
20.6.2 NAND Flash Interface and 1/4-bit ECC Registers 20-17
20.6.3 ECC Registers for 8, 12 and 16-bit ECC 20-30

21 EXTERNAL BUS INTERFACE (EBI) 21-1

21.1 Overview of External Bus Interface 21-1
21.2 Features of Exynos 4412 SCP EBI 21-1
21.3 Block Diagram of Memory Interface through EBI 21-2
21.4 Clock Scheme of Memory Controllers and EBI 21-3

22 SECURE DIGITAL/MULTIMEDIACARD MMC CONTROLLER 22-1

22.1 Overview 22-1
22.2 Features 22-1

22.3 Block Diagram 22-2
22.4 Operation Sequence 22-3
22.4.1 SD Card Detection Sequence 22-3
22.4.2 SD Clock Supply Sequence 22-4
22.4.3 SD Clock Stop Sequence 22-5
22.4.4 SD Clock Frequency Change Sequence 22-5
22.4.5 SD Bus Power Control Sequence 22-6
22.4.6 Change Bus Width Sequence 22-7
22.4.7 Timeout Setting for DAT Line 22-8
22.4.8 SD Transaction Generation 22-8
22.4.9 SD Command Issue Sequence 22-9
22.4.10 Command Complete Sequence 22-11
22.4.11 Transaction Control with Data Transfer Using DAT Line 22-13
22.4.12 Sequence without Using DMA 22-14
22.4.13 Sequence Using DMA 22-16
22.5 Abort Transaction 22-18
22.6 DMA Transaction 22-18
22.7 Advanced DMA 22-19
22.7.1 Block Diagram of ADMA 22-19
22.7.2 Example of ADMA Programming 22-20
22.7.3 Data Address and Data Length Requirements 22-20
22.7.4 Descriptor Table 22-21
22.7.5 ADMA States 22-22
22.8 I/O Description 22-24
22.9 Register Description 22-26
22.9.1 Register Map Summary 22-26

23 MOBILE STORAGE HOST 23-1

23.1 Overview 23-1
23.2 Features of Mobile Storage Host 23-2
23.3 Block Diagram of Mobile Storage Host 23-3
23.4 Clock Phase Shifter 23-4
23.5 I/O Description 23-5
23.6 Register Description 23-6
23.6.1 Register Map Summary 23-6

24 PULSE WIDTH MODULATION TIMER 24-1

24.1 Overview 24-1
24.2 Features of PWM Timer 24-4
24.3 PWM Operation 24-5
24.3.1 Prescaler and Divider 24-5
24.3.2 Basic Timer Operation 24-6
24.3.3 Auto-reload and Double Buffering 24-8
24.3.4 Timer Operation Example 24-9
24.3.5 Initialize Timer (Setting Manual-Up Data and Inverter) 24-10
24.3.6 PWM 24-10
24.3.7 During Current ISR. (Interrupt Service Routine) Output Level Control 24-11
24.3.8 Dead Zone Generator 24-12
24.4 I/O Description 24-13
24.5 Register Description 24-14
24.5.1 Register Map Summary 24-14

25 MULTI CORE TIMER (MCT).....25-1

25.1 Overview 25-1
25.2 Features of MCT 25-2
25.3 Internal Function of MCT 25-3
25.3.1 Global Timer 25-4
25.3.2 Local Timer 25-6
25.3.3 Usage Model 25-10
25.4 Register Description 25-14
25.4.1 Register Map Summary 25-14

26 WATCHDOG TIMER.....26-1

26.1 Overview 26-1
26.2 Features of WDT 26-1
26.3 Functional Description 26-2
26.3.1 WDT Operation 26-2
26.3.2 WTDAT and WTCNT 26-3
26.3.3 WDT Start 26-3
26.3.4 Consideration of Debugging Environment 26-3
26.4 Register Description 26-4
26.4.1 Register Map Summary 26-4

27 REAL TIME CLOCK (RTC).....27-1

27.1 Overview 27-1
27.2 Features of RTC 27-1
27.2.1 Block Diagram 27-2
27.3 Leap Year Generator 27-3
27.4 Read/Write Register 27-4
27.4.1 Backup Battery Operation 27-4
27.5 Alarm Function 27-5
27.6 Tick Time Interrupt 27-6
27.7 32.768 kHz X-Tal Connection Example 27-7
27.8 I/O Description 27-8
27.9 Register Description 27-9
27.9.1 Register Map Summary 27-9

28 UNIVERSAL ASYNCHRONOUS RECEIVER AND TRANSMITTER.....28-1

28.1 Overview 28-1
28.2 Features 28-1
28.3 UART Description 28-1
28.3.1 Data Transmission 28-3
28.3.2 Data Reception 28-3
28.3.3 AFC 28-4
28.3.4 Example of Non AFC (Controlling nRTS and nCTS by Software) 28-5
28.3.5 Trigger Level of Tx/Rx FIFO and DMA Burst Size in DMA Mode 28-5
28.3.6 RS-232C Interface 28-5
28.3.7 Interrupt/DMA Request Generation 28-6
28.3.8 UART Error Status FIFO 28-8
28.4 UART Input Clock Description 28-11
28.5 I/O Description 28-12
28.6 Register Description 28-13
28.6.1 Register Map Summary 28-13

29 INTER-INTEGRATED CIRCUIT 29-1

29.1 Overview 29-1
29.2 Features 29-2
29.3 Block Diagram 29-2
29.4 I2C-Bus Interface Operation 29-3
29.4.1 Start and Stop Conditions 29-4
29.4.2 Data Transfer Format 29-5
29.4.3 ACK Signal Transmission 29-6
29.4.4 Read-Write Operation 29-7
29.4.5 Bus Arbitration Procedures 29-7
29.4.6 Abort Conditions 29-7
29.4.7 Configuring I2C-Bus 29-7
29.4.8 Flowcharts of Operations in Each Mode 29-8
29.5 I/O Description 29-12
29.6 Register Description 29-13
29.6.1 Register Map Summary 29-13

30 SERIAL PERIPHERAL INTERFACE 30-1

30.1 Overview 30-1
30.2 Features 30-1
30.2.1 Operation of SPI 30-2
30.3 SPI Input Clock Description 30-5
30.4 IO Description 30-6
30.5 Register Description 30-7
30.5.1 Register Map Summary 30-7

31 USB 2.0 HOST CONTROLLER 31-1

31.1 Overview 31-1
31.2 Block Diagram 31-2
31.3 I/O Description 31-4
31.4 HSIC Enable/Disable Setting 31-4
31.5 Register Description 31-5
31.5.1 Register Map Summary 31-5
31.5.2 Implemented Specific Registers 31-7

32 USB2.0 DEVICE 32-1

32.1 Overview of USB2.0 Device 32-1
32.2 Key Features of USB2.0 Device 32-1
32.3 Block Diagram of USB2.0 Device 32-2
32.4 Modes of Operation 32-3
32.4.1 DMA Mode 32-3
32.4.2 Slave Mode 32-3
32.5 Power Management Unit Setting 32-4
32.6 Register Map 32-5
32.6.1 Overview of Register Map 32-5
32.6.2 Device Link CSR Memory Map 32-5
32.6.3 Device FIFO Address Mapping 32-6
32.6.4 Application Access to the CSRs 32-8
32.7 I/O Description 32-9
32.8 Register Description 32-10
32.8.1 Register Map Summary 32-10

32.8.2 USB PHY Control Register 32-17
32.8.3 OTG LINK Core Register (OTG Global Register) 32-33
32.8.4 Device Mode Register (Device Global Register) 32-59
32.8.5 Device Logical Endpoint-Specific Register 32-73

33 TRANSPORT STREAM INTERFACE 33-1

33.1 Overview 33-1
33.1.1 Features 33-1
33.1.2 Broadcast Mode 33-2
33.1.3 Block Diagram 33-4
33.1.4 I/O Description of TSI 33-5
33.1.5 Functional Description 33-6
33.2 Register Description 33-17
33.2.1 Register Map Summary 33-17

34 AUDIO SUBSYSTEM 34-1

34.1 Overview 34-1
34.2 Features 34-1
34.3 Block Diagram of Audio Block 34-2
34.4 Functional Description 34-6
34.4.1 Reconfigurable Processor 34-6
34.4.2 ASS CLK CON 34-6
34.4.3 Commbox 34-6
34.4.4 I2S 34-7
34.4.5 SRAM 34-7
34.5 Input/Output Description 34-8
34.6 Register Description 34-9
34.6.1 Register Map Summary 34-9
34.6.2 Audio Subsystem CLKCON 34-11
34.6.3 Communication Box (Commbox) 34-13

35 IIS MULTI AUDIO INTERFACE 35-1

35.1 Overview 35-1
35.2 Features 35-1
35.3 Block Diagram 35-2
35.4 Functional Description 35-3
35.4.1 Master/Slave Mode 35-3
35.5 Audio Serial Data Format 35-8
35.5.1 IIS-Bus Format 35-8
35.5.2 MSB (Left) Justified 35-8
35.5.3 LSB (Right) Justified 35-9
35.6 PCM Bit Length (BLC), RFS Divider and BFS Divider for Sampling Frequency (IISLRCLK), Serial bit CLK (IISSCLK), and Root Clock (RCLK) 35-10
35.6.1 PCM Word Length and BFS Divider 35-10
35.6.2 BFS Divider and RFS Divider 35-10
35.6.3 RFS Divider and Root Clock 35-10
35.7 Programming Guide 35-11
35.7.1 Initialization 35-11
35.7.2 Play Mode (Tx Mode) with DMA 35-11
35.7.3 Recording Mode (Rx Mode) with DMA 35-12
35.7.4 Example Code 35-13

35.8 IO Description 35-19
35.9 Register Description 35-20
35.9.1 Register Map Summary 35-20

36 IIS-BUS INTERFACE 36-1

36.1 Overview 36-1
36.2 Features 36-1
36.3 Block Diagram 36-2
36.4 Functional Description 36-3
36.4.1 Master/Slave Mode 36-3
36.4.2 DMA Transfer 36-4
36.4.3 Audio Serial Data Format 36-5
36.4.4 PCM Word Length and BFS Divider 36-8
36.4.5 BFS Divider and RFS Divider 36-8
36.4.6 RFS Divider and Root Clock 36-8
36.5 Programming Guide 36-9
36.5.1 Initialization 36-9
36.5.2 Play Mode (Tx Mode) with DMA 36-9
36.5.3 Recording Mode (Rx Mode) with DMA 36-9
36.5.4 Example Code 36-10
36.6 I/O Description 36-16
36.7 Register Description 36-17
36.7.1 Register Map Summary 36-17

37 AC97 CONTROLLER 37-1

37.1 Overview 37-1
37.2 Features 37-1
37.3 AC97 Controller Operation 37-2
37.3.1 Block Diagram 37-2
37.3.2 Internal Data Path 37-3
37.3.3 Operation Flow Chart 37-4
37.3.4 AC-link Digital Interface Protocol 37-5
37.3.5 AC-link Input Frame (SDATA_IN) 37-7
37.3.6 AC97 Power-Down 37-9
37.4 I/O Description 37-12
37.5 Register Description 37-13
37.5.1 Register Map Summary 37-13

38 PCM AUDIO INTERFACE 38-1

38.1 Overview 38-1
38.2 Features 38-1
38.3 PCM Audio Interface 38-2
38.4 PCM Timing 38-3
38.5 I/O Description 38-5
38.6 Register Description 38-6
38.6.1 Register Map Summary 38-6

39 SPDIF TRANSMITTER 39-1

39.1 Overview 39-1
39.2 Features 39-1
39.3 Block Diagram 39-2

39.4 Functional Description 39-3
39.4.1 Data Format of SPDIF 39-3
39.4.2 Channel Coding 39-6
39.4.3 Preamble 39-6
39.4.4 Non-Linear PCM-Encoded Source (IEC 61937) 39-7
39.4.5 SPDIF Operation 39-9
39.4.6 Shadowed Register 39-10
39.5 I/O Description 39-11
39.6 Register Description 39-12
39.6.1 Register Map Summary 39-12

40 HIGH-SPEED SYNCHRONOUS SERIAL INTERFACE (HSI) 40-1

40.1 Overview 40-1
40.2 Features 40-1
40.3 Functional Description 40-2
40.3.1 Block Diagram 40-2
40.3.2 Interface Port Description 40-4
40.3.3 Programming Basic Sequence Guide 40-5
40.4 Register Description 40-11
40.4.1 Register Map Summary 40-11

41 DISPLAY CONTROLLER 41-1

41.1 Overview 41-1
41.2 Features of Display Controller 41-2
41.3 Functional Description 41-4
41.3.1 Brief Description of the Sub-Block 41-4
41.3.2 Data Flow 41-5
41.3.3 Overview of the Color Data 41-8
41.3.4 Color Space Conversion 41-23
41.3.5 Palette Usage 41-25
41.3.6 Window Blending 41-28
41.3.7 Image Enhancement 41-37
41.3.8 VTIME Controller Operation 41-44
41.3.9 Setting of Commands 41-47
41.3.10 Virtual Display 41-50
41.3.11 RGB Interface Specification 41-51
41.3.12 LCD Indirect i80 System Interface 41-60
41.4 I/O Description 41-64
41.5 Register Description 41-65
41.5.1 Register Map Summary 41-66
41.5.2 Palette Memory 41-72
41.5.3 Control Register 41-73
41.5.4 Gamma Lookup Table 41-145
41.5.5 Shadow Windows Control 41-146
41.5.6 Palette Ram 41-147

42 CAMERA INTERFACE AND SCALER 42-1

42.1 Overview 42-1
42.2 Features of CAMIF 42-3
42.3 External Interface 42-5
42.4 Timing Diagram and Data Alignment of Camera 42-6

42.4.1 Timing Diagram of ITU Camera 42-6
42.4.2 MIPI CSI Data Alignment from MIPI Camera 42-10
42.5 External Connection Guide 42-11
42.6 Camera Interface Operation 42-12
42.6.1 Input/Output DMA Ports 42-12
42.6.2 Clock Domain 42-13
42.6.3 Frame Memory Hierarchy 42-14
42.6.4 Memory Storing Method 42-15
42.6.5 Timing Diagram for Register Setting 42-16
42.6.6 Timing Diagram for last IRQ 42-18
42.6.7 Timing Diagram for IRQ (Memory Data Scaling Mode) 42-20
42.6.8 Input DMA Feature 42-21
42.6.9 Camera Interlace Input Support 42-22
42.7 Input/Output Description 42-23
42.8 Register Description 42-24
42.8.1 Register Map Summary 42-24

43 FIMC_LITE (CAMERA INTERFACE) 43-1

43.1 Overview of Camera Interface in FIMC_LITE 43-1
43.1.1 Block Diagram 43-1
43.2 Timing Diagram and Data Alignment of Camera 43-2
43.2.1 Parallel INTERFACE 43-2
43.2.2 MIPI CSI Slave Interface 43-3
43.2.3 Local Output Interface Data Alignment 43-4
43.3 External Connection Guide 43-5
43.4 Input / Output Path 43-6
43.5 I/O Description 43-7
43.6 Register Description 43-8
43.6.1 Register Map 43-8

44 MIPI-DSI MASTER 44-1

44.1 Overview 44-1
44.2 Features 44-1
44.2.1 Block Diagram 44-2
44.2.2 Interfaces and Protocol 44-6
44.2.3 Configuration 44-16
44.2.4 Dual Display versus Single Display 44-17
44.2.5 PLL 44-18
44.2.6 Buffer 44-18
44.3 I/O Description 44-19
44.4 Register Description 44-20
44.4.1 Register Map Summary 44-20
44.5 DPHY PLL Control 44-39
44.5.1 PMS Setting Sample for MIPI PLL 44-39

45 MIPI-CSI SLAVE (MIPI-CSI) 45-1

45.1 Overview of MIPI CSIS 45-1
45.2 Features 45-1
45.3 Block Diagram 45-2
45.4 Interface and Protocol 45-3
45.5 Data Format 45-4

45.5.1 Data Alignment 45-4
45.5.2 YUV422 8-bit Order 45-4
45.6 I/O Description 45-5
45.7 Register Description 45-6
45.7.1 Register Map Summary 45-6

46 2D GRAPHIC ACCELERATOR 46-1

46.1 Overview 46-1
46.2 Features 46-2
46.2.1 Host Interface 46-2
46.2.2 Primitives 46-2
46.2.3 Per-pixel Operation 46-3
46.2.4 Data Format 46-3
46.3 Host Interface: DMA Mode 46-4
46.4 Color Format Conversion 46-9
46.4.1 RGBA Format 46-9
46.5 Rendering Pipeline 46-11
46.5.1 Primitive Drawing 46-12
46.5.2 Rotation and Addressing Direction (Flip) 46-15
46.5.3 Clipping 46-17
46.5.4 Color Key 46-18
46.5.5 Raster Operation 46-19
46.5.6 Mask Operation 46-21
46.5.7 Alpha Blending 46-22
46.5.8 Fast Solid Color Fill 46-26
46.6 Register Description 46-27
46.6.1 Register Map Summary 46-27
46.6.2 General Registers 46-31
46.6.3 Command Registers 46-37
46.6.4 Parameter Setting Registers 46-43
46.6.5 Source 46-45
46.6.6 Destination 46-50
46.6.7 Pattern 46-54
46.6.8 Mask 46-56
46.6.9 Clipping Window 46-60
46.6.10 ROP & Alpha Setting 46-62
46.6.11 Color 46-63
46.6.12 Color Key 46-65
46.6.13 Gamma Table 46-68

47 3D GRAPHIC ACCELERATOR (G3D) 47-1

47.1 Overview of G3D 47-1
47.2 Features of G3D 47-2
47.3 Architecture Brief of G3D 47-3
47.4 G3D Structure 47-4
47.5 GPU Hardware Architecture 47-5
47.5.1 Top-level System 47-5
47.5.2 Functional Block Diagram of GPU Hardware Architecture 47-6
47.5.3 PMU Hardware Architecture 47-8
47.5.4 Level 2 Cache Controller Hardware Architecture 47-9

48 IMAGE ROTATOR 48-1

48.1 Overview of Image Rotator 48-1
48.2 Features of Image Rotator 48-1
48.3 Block Diagram 48-2
48.4 Supported Image Rotation Functions 48-3
48.5 Image Rotation with Windows Offset 48-4
48.6 Programming Guide 48-5
48.6.1 Resister Setting 48-5
48.6.2 Restrictions on the Image Size 48-5
48.7 Register Description 48-6
48.7.1 Register Map Summary 48-6

49 JPEG CODEC 49-1

49.1 Overview 49-1
49.2 Features 49-2
49.3 Color Component Data Ordering 49-3
49.4 Register Description 49-4
49.4.1 Register Map Summary 49-4
49.5 Programmer's Model 49-18
49.5.1 Encoder Flow Chart 49-18
49.5.2 Programming QUAN_TBL_ENT and HUFF_TBL_ENT 49-19
49.6 Example Codes 49-21
49.7 References 49-22

50 MULTI FORMAT CODEC (MFC) 50-1

50.1 Introduction 50-1
50.1.1 Supported Standards 50-1
50.1.2 Features 50-3
50.1.3 Target Performance and Functions 50-4
50.2 Hardware Overview 50-6
50.2.1 Block Diagram 50-6
50.2.2 Frame Memory 50-8
50.3 Register Description 50-11
50.3.1 Register Map Summary 50-11
50.3.2 Control Registers 50-19
50.3.3 Codec Registers 50-38
50.3.4 Encoding Registers 50-59
50.4 Shared Memory Interface 50-68
50.4.1 Host Interface 50-68
50.4.2 Shared Memory Structure 50-69
50.5 Metadata Interface 50-87
50.5.1 Shared Memory Interface for Decoders 50-87
50.5.2 Shared Memory Interface for Encoders 50-91
50.6 Appendix 50-92
50.6.1 Summary of Buffer Requirements 50-92
50.6.2 Batch Encoding Interface 50-93

51 VIDEO PROCESSOR 51-1

51.1 Overview of Video Processor 51-1
51.1.1 Features 51-1
51.2 Block Diagram 51-2
51.3 Functional Description 51-3

51.3.1 BOB in VP 51-3
51.3.2 Nterlace to Progressive Conversion 51-4
51.4 Register Description 51-5
51.4.1 Register Map Summary 51-5
51.4.2 The Idea of Poly-phase Filtering in Video Processor 51-47

52 MIXER 52-1

52.1 Overview of Mixer 52-1
52.2 Features 52-1
52.3 Block Diagram 52-3
52.3.1 Video Clock Relation 52-4
52.4 Register Description 52-5
52.4.1 Register Map Summary 52-5
52.4.2 Shadow Registers (Read Only) 52-8
52.5 Layers 52-28
52.5.1 Video Layer 52-30
52.5.2 Graphic Layer 52-30
52.5.3 Blank Pixel 52-31
52.5.4 Source Data in Memory 52-32
52.5.5 Background Layer 52-32

53 HIGH-DEFINITION MULTIMEDIA INTERFACE 53-1

53.1 Overview 53-1
53.2 Features 53-1
53.3 Block Diagram of HDMI 53-2
53.4 Block Diagram of HDMI SUB-system in Exynos 4412 SCP 53-3
53.5 Block Diagram of HDCP Key Management 53-4
53.6 Video Input Timing GUIDE for HDMI Timing Generator 53-5
53.7 HDMI PHY Configuration 53-9
53.7.1 Selected i2C Register Control 53-11
53.8 Block Diagram of Clock Strategy for HDMI Tx 53-13
53.9 SPDIF (Auxiliary Information) 53-15
53.9.1 Frame Format 53-15
53.10 Register Description 53-21
53.10.1 Register Map Summary 53-21
53.10.2 Control Registers 53-34
53.10.3 HDMI Core Registers (Control Registers) 53-42
53.10.4 HDMI Core Registers (Video Related Registers) 53-48
53.10.5 HDMI Core Registers (Audio Related Registers) 53-70
53.10.6 HDMI Core Registers (Packet Related Registers) 53-74
53.10.7 HDMI Core Registers (HDCP Related Registers) 53-89
53.10.8 SPDIF Registers 53-103
53.10.9 I2S Registers 53-121
53.10.10 Timing Generator Register 53-135
53.10.11 HDCP eFUSE Registers 53-144
53.10.12 CEC Configure Registers 53-148

54 SECURITY SUB SYSTEM 54-1

54.1 Overview 54-1
54.1.1 Features 54-3
54.2 Functional Description 54-4

54.2.1 Data Flows through SFR (CPU Mode) 54-4
54.2.2 Data Flows through DMA (DMA Mode) 54-5
54.2.3 Byte Swapping Options 54-14
54.3 Programmer's Model 54-18
54.3.1 Feeder 54-18
54.3.2 AES 54-21
54.3.3 DES/3DES 54-24
54.3.4 HASH/PRNG 54-27
54.3.5 PKA 54-47
54.4 Register Description 54-60
54.4.1 Register Map Summary 54-60
54.4.2 Feeder Hardware Register Base 54-66
54.4.3 AES Engine Hardware Register Base 54-73
54.4.4 TDES Engine Hardware Register Base 54-81
54.4.5 HASH Engine Hardware Register Base 54-86
54.4.6 PKA Engine Hardware Register Base 54-99
54.4.7 PKA Engine Local Memory 54-108
54.5 Example Codes 54-109
54.5.1 SFR Definition 54-109
54.5.2 Feeder 54-109
54.5.3 AES 54-112
54.5.4 DES/3DES 54-118
54.5.5 HASH/PRNG 54-124
54.5.6 PKA 54-139
54.5.7 Interrupt Service Routine 54-147
54.5.8 Useful Routines 54-148

55 KEYPAD INTERFACE 55-1

55.1 Overview of Keypad Interface 55-1
55.2 Debouncing Filter 55-3
55.3 Filter Clock 55-3
55.4 Wakeup Source 55-3
55.5 Keypad Scanning Procedure for Software Scan 55-4
55.6 Keypad Scanning Procedure for Hardware Scan 55-9
55.7 I/O Description 55-10
55.8 Register Description 55-12
55.8.1 Register Map Summary 55-12

56 ADC 56-1

56.1 Overview of ADC 56-1
56.2 KEY Features of ADC for motor control 56-1
56.3 ADC Operation 56-3
56.3.1 Block Diagram ADC 56-3
56.4 Function Descriptions 56-4
56.4.1 ADC selection 56-4
56.4.2 A/D Conversion Time 56-4
56.4.3 ADC conversion Mode 56-4
56.4.4 Standby Mode 56-4
56.5 ADC Input Clock Diagram 56-5
56.6 I/O Descriptions 56-5
56.7 Register Description 56-6
56.7.1 Register Map Summary 56-6

57 THERMAL MANAGEMENT UNIT.....57-1

57.1 Overview 57-1
57.2 Temperature Sensing Auto Mode with External Clocks 57-3
57.3 Temperature Code Table 57-4
57.4 I/O Description 57-6
57.5 Register Description 57-7
57.5.1 Register Map Summary 57-7
57.6 Programming Guide 57-19
57.6.1 Calibration Equation 57-19
57.6.2 Software Sequence 57-20
57.6.3 Interrupt Service Routine 57-21
57.6.4 Tracing Past Temperature 57-22

58 FIMC-IS.....58-1

58.1 Exynos 4412 SCP 58-1
58.1.1 Overview 58-1
58.1.2 Features 58-1
58.1.3 FIMC-IS V1.5 Brief Interface 58-2
58.1.4 General Description 58-3
58.1.5 Programmer's View 58-9
58.2 FIMC-ISP 58-13
58.2.1 Overview 58-13
58.2.2 Features 58-13
58.2.3 General Description 58-13
58.2.4 Functional Description 58-14
58.3 FIMC-DRC 58-19
58.3.1 Overview 58-19
58.3.2 Features 58-19
58.3.3 General Description 58-20
58.3.4 Functional Description 58-21
58.4 FIMC-FD 58-23
58.4.1 Overview 58-23
58.4.2 Features 58-23
58.4.3 General Description 58-24
58.4.4 Functional Description 58-24
58.5 MCUCTL 58-25
58.5.1 Overview 58-25
58.5.2 Features 58-25
58.5.3 Functional Description 58-26
58.6 MPWM 58-27
58.6.1 Overview 58-27
58.6.2 Features 58-28
58.6.3 Operation of MPWM 58-29
58.6.4 Interrupt 58-30
58.7 ADC for Motor Control 58-31
58.7.1 Overview of ADC 58-31
58.7.2 Key Features of ADC for Motor Control 58-31
58.7.3 Block Diagram of ADC 58-32
58.7.4 Function Descriptions 58-33

59 ELECTRICAL DATA.....59-1

59.1 Absolute Maximum Ratings 59-1
59.2 Recommended Operating Conditions 59-2
59.3 D.C. Electrical Characteristics 59-6
59.4 IO Driver Strength 59-8
59.4.1 Input/ Output Types 59-8
59.4.2 Type A IO Driver Strength 59-9
59.4.3 Type B IO Driver Strength 59-9
59.5 CLK Alternating Current Electrical Characteristics 59-10
59.6 ROM/SRAM AC Electrical Characteristics 59-12
59.7 NFCON AC Electrical Characteristics 59-14
59.8 LPDDR2 Electrical Characteristics 59-16
59.9 LCD Controller AC Electrical Characteristics 59-17
59.10 Camera Interface AC Electrical Characteristics 59-20
59.11 SDMMC AC Electrical Characteristics 59-23
59.12 SPI AC Electrical Characteristics 59-24
59.13 I2C AC Electrical Characteristics 59-26

60 MECHANICAL DATA 60-1

60.1 Physical Dimension 60-1
60.1.1 Pin Assignment Diagram 786-ball FCFBGA (SCP) 60-2
60.1.2 Package Dimension 60-3
60.2 Package Marking 60-5
60.2.1 Package Marking Line Description 60-5

List of Figures

Figure Number	Title	Page Number
Figure 1-1	Exynos 4412 SCP Block Diagram	1-2
Figure 2-1	Exynos 4412 SCP Pin Map (786-FCFBGA) Top View	2-2
Figure 5-1	Block Diagram of Booting Time Operation	5-2
Figure 6-1	GPIO Block Diagram	6-4
Figure 7-1	Exynos 4412 SCP Clock Domains	7-2
Figure 7-2	Exynos 4412 SCP Clock Generation Circuit (CPU, BUS, DRAM, ISP Clocks)	7-13
Figure 7-3	Exynos 4412 SCP Clock Generation Circuit (Special Clocks)	7-15
Figure 7-4	Exynos 4412 SCP CLKOUT Control Logic	7-21
Figure 7-5	Exynos 4412 SCP Clock Controller Address Map	7-26
Figure 8-1	Procedures for System-Level Power Mode	8-15
Figure 8-2	Power-on/off Reset Sequence	8-25
Figure 9-1	Interrupt Sources Connection	9-8
Figure 9-2	GIC Block Diagram	9-18
Figure 9-3	Distributor	9-20
Figure 9-4	CPU Interface	9-21
Figure 9-5	Interrupt Handling State Machine	9-32
Figure 9-6	ICDISRn Address Map	9-63
Figure 9-7	ICDISERn Address Map	9-65
Figure 9-8	ICDICERn Address Map	9-67
Figure 9-9	ICDISPRn Address Map	9-69
Figure 9-10	ICDICPRn Address Map	9-71
Figure 9-11	Logic of the Pending Status of a Level-sensitive Interrupt	9-72
Figure 9-12	ICDABRn Address Map	9-74
Figure 9-13	ICDIPRn Address Map	9-77
Figure 9-14	ICDIPTRn Address Map	9-81
Figure 9-15	ICDICFRn Bit Assignments	9-83
Figure 9-16	ICDICFRn Address Map	9-83
Figure 9-17	SPI_STATUSn Address Map	9-85
Figure 10-1	Block Diagram of Interrupt Combiner	10-2
Figure 10-2	Internal Operation of Interrupt Combiner	10-3
Figure 11-1	Two DMA Tops	11-2
Figure 12-1	Diagram of the ISPWB_FULL_EN Usage	12-10
Figure 13-1	F4Q and CoreSight	13-1
Figure 13-2	Structure of the CoreSight DAP Components	13-3
Figure 13-3	SWJ-DP External Connections	13-4
Figure 13-4	APB Multiplexer	13-4
Figure 13-5	Trace Funnel	13-5
Figure 13-6	CTI and CTM Block Diagram	13-5
Figure 13-7	ROM Table Entries and Address	13-7
Figure 13-8	CoreSight Clock Domain Interactions	13-10
Figure 14-1	Block Diagram of Access Controller (TZPC)	14-1
Figure 14-2	Secure Configuration of Internal Memory	14-2
Figure 15-1	Block Diagram of Access Controller (TZASC)	15-1
Figure 15-2	Miscellaneous Signals	15-4
Figure 15-3	Functional Operation of TZASC	15-6
Figure 15-4	Region Priority	15-7

Figure 15-5 Subregion Example 15-8
Figure 15-6 Subregion Disable Example 15-9
Figure 16-1 SysMMU High-level Block Diagram 16-2
Figure 16-2 An Example of System Integration 16-3
Figure 16-3 Conceptual Block Diagram of Address Translation Unit 16-4
Figure 16-4 First-level Page Descriptor Format 16-7
Figure 16-5 Second-level Page Descriptor Format 16-7
Figure 17-1 SysMMU High-level Block Diagram 17-3
Figure 17-2 An Example of System Integration 17-4
Figure 17-3 Conceptual Block Diagram of Address Translation Unit 17-5
Figure 17-4 First-Level Page Descriptor Format 17-8
Figure 17-5 Second-Level Page Descriptor Format 17-9
Figure 17-6 L2-TLB Block Diagram 17-10
Figure 18-1 Block Diagram of DMC 18-2
Figure 18-2 Linear Address Mapping 18-6
Figure 18-3 Interleaved Address Mapping 18-7
Figure 18-4 Timing Diagram Of Timeout Precharge 18-11
Figure 18-5 An Adaptive DRAM QoS Scheme Configuration 18-14
Figure 18-6 Timing Diagram Of Read data Capture (DDR3, zero delay, RL=3, rd_fetch=1) 18-16
Figure 18-7 Timing Diagram Of Read Data Capture (DDR3, non-zero delay, RL=3, rd_fetch=2) 18-17
Figure 18-8 Timing Diagram Of Read Data Capture (LPDDR2-S4, Zero Delay, RL=3, rd_fetch=1) 18-17
Figure 18-9 Timing Diagram Of Read Data Capture (LPDDR2-S4, Non-Zero Delay, RL=3, rd_fetch=2) 18-18
Figure 18-10 Timing Diagram Of Read Data Capture (LPDDR2-S4, Low Frequency, RL=3, rd_fetch=0) ... 18-18
Figure 18-11 Memory Channels Interleaving 18-20
Figure 18-12 PHY DLL Lock Procedure 18-41
Figure 18-13 Board Level Connection Diagram for DQS Cleaning 18-44
Figure 18-14 DQS Cleaning for LPDDR2 if tAC Min 18-45
Figure 18-15 DQS Cleaning for LPDDR2 if tAC Max 18-45
Figure 18-16 DQS cleaning for DDR3 18-46
Figure 19-1 Block Diagram of SROMC Introduction 19-1
Figure 19-2 SROMC nWAIT Timing Diagram 19-2
Figure 19-3 SROMC Read Timing Diagram 19-3
Figure 19-4 SROMC Write Timing Diagram 19-3
Figure 20-1 NAND Flash Controller Block Diagram 20-2
Figure 20-2 CLE and ALE Timing (TACLS = 1, TWRPH0 = 0, TWRPH1 = 0) 20-3
Figure 20-3 nWE and nRE Timing (TWRPH0 = 0, TWRPH1 = 0) 20-3
Figure 20-4 Accessibility of NAND Area 20-13
Figure 21-1 Memory Interface Through EBI 21-2
Figure 21-2 Clock Scheme of Memory Controllers and EBI 21-3
Figure 22-1 SDMMC Clock Domain 22-2
Figure 22-2 SD Card Detect Sequence 22-3
Figure 22-3 SD Clock Supply Sequence 22-4
Figure 22-4 SD Clock Stop Sequence 22-5
Figure 22-5 SD Clock Frequency Change Sequence 22-5
Figure 22-6 SD Bus Power Control Sequence 22-6
Figure 22-7 Change Bus Width Sequence 22-7
Figure 22-8 Timeout Setting Sequence 22-8
Figure 22-9 Timeout Setting Sequence 22-9
Figure 22-10 Command Complete Sequence 22-12
Figure 22-11 Transaction Control with Data Transfer Using DAT Line Sequence (Not Using DMA) 22-14
Figure 22-12 Transaction Control with Data Transfer Using DAT Line Sequence (Using DMA) 22-16
Figure 22-13 Block Diagram of ADMA 22-19
Figure 22-14 Example of ADMA Data Transfer 22-20

Figure 22-15 32-bit Address Descriptor Table 22-21
Figure 22-16 State Diagram of the ADMA 22-22
Figure 22-17 Card Detect State 22-44
Figure 22-18 Timing of Command Inhibit (DAT) and Command Inhibit (CMD) with Data Transfer 22-44
Figure 22-19 Timing of Command Inhibit (DAT) for the Case of Response with Busy 22-44
Figure 22-20 Timing of Command Inhibit (CMD) for the Case of No Response Command 22-45
Figure 23-1 Block Diagram of Mobile Storage Host 23-3
Figure 23-2 Clock Phase Shifter 23-4
Figure 24-1 Simple Example of a PWM Cycle 24-2
Figure 24-2 PWM TIMER Clock Tree Diagram 24-3
Figure 24-3 Timer Operations 24-6
Figure 24-4 Example of Double Buffering Feature 24-8
Figure 24-5 Example of a Timer Operation 24-9
Figure 24-6 Example of PWM 24-10
Figure 24-7 Inverter On/Off 24-11
Figure 24-8 Waveform when a Dead Zone Feature is Enabled 24-12
Figure 25-1 Overall Multi Core Timer Block Diagram 25-1
Figure 25-2 Function of Global Timer 25-4
Figure 25-3 Periodically Asserted Interrupt 25-5
Figure 25-4 Two Separate Timers for Various Tick 25-6
Figure 25-5 FRC Free Running Counter (FRC) 25-7
Figure 25-6 Timer Operation with Always on of Auto-reload 25-8
Figure 25-7 Timer Operation (One-Shot) 25-8
Figure 25-8 FRC Operation (Always Auto-reload) 25-9
Figure 26-1 Watchdog Timer Block Diagram 26-2
Figure 27-1 RTC Block Diagram 27-2
Figure 27-2 Main Oscillator Circuit Example 27-7
Figure 28-1 Block Diagram of UART 28-2
Figure 28-2 UART AFC Interface 28-4
Figure 28-3 UART Receives the Five Characters Including Two Errors 28-8
Figure 28-4 IrDA Function Block Diagram 28-9
Figure 28-5 Serial I/O Frame Timing Diagram (Normal UART) 28-9
Figure 28-6 Infra-Red Transmit Mode Frame Timing Diagram 28-10
Figure 28-7 Infra-Red Receive Mode Frame Timing Diagram 28-10
Figure 28-8 Input Clock Diagram for UART 28-11
Figure 28-9 nCTS and Delta CTS Timing Diagram 28-26
Figure 28-10 Block diagram of UINTSP, UINTP, and UINTM 28-30
Figure 29-1 I2C-Bus Block Diagram 29-2
Figure 29-2 Start and Stop Condition 29-4
Figure 29-3 I2C-Bus Interface Data Format 29-5
Figure 29-4 Data Transfer on the I2C-Bus 29-5
Figure 29-5 Acknowledgement on the I2C-Bus 29-6
Figure 29-6 Operations for Master/Transmitter Mode 29-8
Figure 29-7 Operations for Master/Receiver Mode 29-9
Figure 29-8 Operations for Slave/Transmitter Mode 29-10
Figure 29-9 Operations for Slave/Receiver Mode 29-11
Figure 30-1 SPI Transfer Format 30-4
Figure 30-2 Input Clock Diagram for SPI 30-5
Figure 30-3 Auto Chip Select Mode Waveform (CPOL = 0, CPHA = 0, CH_WIDTH = Byte) 30-10
Figure 31-1 USB System Block Diagram 31-2
Figure 31-2 USB 2.0 Host Controller Block Diagram 31-3
Figure 32-1 System Level Block Diagram 32-2
Figure 32-2 Device Link CSR Memory Map 32-5

Figure 32-3 Device FIFO Mapping 32-6
Figure 32-4 USB PHY Clock Path 32-20
Figure 32-5 COMPDISTUNE 32-28
Figure 32-6 OTGTUNE 32-29
Figure 32-7 SQRXTUNE 32-29
Figure 32-8 TXFSLSTUNE 32-30
Figure 32-9 TXPREEMPAMPTUNE 32-30
Figure 32-10 TXPREEMPPULSETUNE 32-31
Figure 32-11 TXRISETUNE 32-31
Figure 32-12 TXVREFTUNE 32-32
Figure 32-13 TXHSXVTUNE 32-32
Figure 33-1 Support TSI in the Broadcasting Mode 33-2
Figure 33-2 TSI Block Diagram 33-4
Figure 33-3 Transport Stream Packet Data Format 33-6
Figure 33-4 Transport Stream Signals 33-7
Figure 33-5 Sync Detection Using TS_SYNC Signal 33-8
Figure 33-6 Sync Detection Using Sync Byte (sync_det_cnt = 3) 33-9
Figure 33-7 TSI Error Cases (with Skip Mode, TS_VALID/TS_SYNC/TS_Error is Active High) 33-12
Figure 33-8 Block Diagram of TS_CLK Filter 33-13
Figure 33-9 TSI Control Finite State Machine (FSM) 33-15
Figure 34-1 Block Diagram of Audio Block 34-2
Figure 34-2 GPIO for Audio Sub-System 34-4
Figure 34-3 JTAG Port of SRP Debugging 34-5
Figure 34-4 Clock Controller in Audio Subsystem 34-6
Figure 35-1 IIS-Bus Block Diagram 35-2
Figure 35-2 Clock Controller in Audio Sub-System 35-3
Figure 35-3 IIS Clock Control Block Diagram 35-4
Figure 35-4 Master/Slave Modes of IIS 35-5
Figure 35-5 Concept of Mixer in IIS 35-7
Figure 35-6 IIS Audio Serial Data Formats 35-9
Figure 35-7 Tx FIFO Structure for BLC = 00 or BLC = 01 35-14
Figure 35-8 Tx FIF0 Structure for BLC = 10 (24-bit/Channel) 35-15
Figure 35-9 Rx FIFO Structure for BLC = 00 or BLC = 01 35-17
Figure 35-10 Rx FIF0 Structure for BLC = 10 (24-bit/Channel) 35-18
Figure 36-1 IIS-Bus Block Diagram 36-2
Figure 36-2 IIS Clock Control Block Diagram 36-3
Figure 36-3 IIS Audio Serial Data Formats 36-6
Figure 36-4 Tx FIFO Structure for BLC = 00 or BLC = 01 36-11
Figure 36-5 Tx FIF0 Structure for BLC = 10 (24-bit/Channel) 36-12
Figure 36-6 Rx FIFO Structure for BLC = 00 or BLC = 01 36-14
Figure 36-7 Rx FIF0 Structure for BLC = 10 (24-bit/Channel) 36-15
Figure 37-1 AC97 Block Diagram 37-2
Figure 37-2 Internal Data Path 37-3
Figure 37-3 AC97 Operation Flow Chart 37-4
Figure 37-4 Bi-Directional AC-link Frame with Slot Assignments 37-5
Figure 37-5 AC-Link Output Frame 37-6
Figure 37-6 AC-Link Input Frame 37-8
Figure 37-7 AC97 Power-Down Timing 37-9
Figure 37-8 AC97 Power Down/Power Up Flow 37-10
Figure 37-9 AC97 State Diagram 37-11
Figure 38-1 PCM Timing, POS_MSB_WR/RD = 0 38-3
Figure 38-2 PCM timing, POS_MSB_WR/RD = 1 38-4
Figure 38-3 Input Clock Diagram for PCM 38-4

Figure 39-1 Block Diagram of SPDIFOUT 39-2
Figure 39-2 SPDIF Frame Format 39-4
Figure 39-3 SPDIF Sub-Frame Format 39-5
Figure 39-4 Channel Coding 39-6
Figure 39-5 Format of Burst-Payload 39-8
Figure 40-1 Functional Block Diagram of HSI Controller 40-2
Figure 40-2 Transmit or Receive Control Flow in PIO Mode 40-5
Figure 41-1 Block Diagram of Display Controller 41-1
Figure 41-2 Block Diagram of the Data Flow 41-6
Figure 41-3 Block Diagram of the Interface 41-7
Figure 41-4 Memory Format of 25 BPP(A888) Display 41-8
Figure 41-5 Memory Format of 32 BPP (8888) Display 41-9
Figure 41-6 Memory Format of 24 BPP(A887) Display 41-10
Figure 41-7 Memory Format of 24 BPP (888) Display 41-11
Figure 41-8 Memory Format of 19 BPP (A666) Display 41-12
Figure 41-9 Memory Format of 18 BPP (666) Display 41-13
Figure 41-10 Memory Format of 16 BPP (A555) Display 41-14
Figure 41-11 Memory Format of 16 BPP (1555) Display 41-15
Figure 41-12 Memory Format of 16 BPP (565) Display 41-16
Figure 41-13 16 BPP (5:6:5) Display Types 41-17
Figure 41-14 Memory Format of 13 BPP (A444) Display 41-18
Figure 41-15 Memory Format of 8 BPP (A232) Display 41-19
Figure 41-16 Memory Format of 8 BPP Display 41-20
Figure 41-17 Memory Format of 4 BPP Display 41-21
Figure 41-18 Memory Format of 2 BPP Display 41-22
Figure 41-19 32 BPP (8:8:8:8) Palette Data Format 41-25
Figure 41-20 25 BPP (A: 8:8:8) Palette Data Format 41-26
Figure 41-21 19 BPP (A: 6:6:6) Palette Data Format 41-26
Figure 41-22 16 BPP (A: 5:5:5) Palette Data Format 41-27
Figure 41-23 Blending Equation 41-30
Figure 41-24 Blending Diagram 41-32
Figure 41-25 Blending Factor Decision 41-33
Figure 41-26 Color-Key Function Configurations 41-34
Figure 41-27 Blending and Color-Key Function 41-35
Figure 41-28 Blending Decision Diagram 41-36
Figure 41-29 Image Enhancement Flow 41-37
Figure 41-30 Image Enhancement Flow 41-38
Figure 41-31 Image Enhancement Flow 41-39
Figure 41-32 Hue Coefficient Decision 41-42
Figure 41-33 Hue Control Block Diagram 41-42
Figure 41-34 Example1. RGBSPSEL == 1'b0, RGB_SKIP == 1'b1, PIXCOMPEN_DIR == 1'b0 41-43
Figure 41-35 Example2. RGBSPSEL == 1'b1, PIXCOMPEN_DIR == 1'b0 41-43
Figure 41-36 Example3. RGBSPSEL == 1'b1, PIXCOMPEN_DIR == 1'b1 41-43
Figure 41-37 Sending Command 41-48
Figure 41-38 Example of Scrolling in Virtual Display 41-50
Figure 41-39 LCD RGB Interface Timing 41-53
Figure 41-40 LCD RGB Interface Timing (RGB Parallel) 41-54
Figure 41-41 LCD RGB Interface Timing (RGB Skip) 41-55
Figure 41-42 LCD RGB Interface Timing (RGB Serial, Dummy Disable) 41-56
Figure 41-43 LCD RGB Interface Timing (RGB serial, Dummy Insertion) 41-57
Figure 41-44 LCD RGB Output Order 41-58
Figure 41-45 Delta Structure and LCD RGB Interface Timing 41-59
Figure 41-46 Indirect i80 System Interface Write Cycle Timing 41-62

Figure 42-1 Subset of Visual System in Exynos 4412 SCP 42-2
Figure 42-2 Camera Interface Overview 42-4
Figure 42-3 ITU-R BT 601 Input Timing Diagram 42-6
Figure 42-4 ITU-R BT 601 Interlace Handling Diagram 42-7
Figure 42-5 ITU-R BT 656 Input Timing Diagram 42-7
Figure 42-6 Sync Signal Timing Diagram 42-8
Figure 42-7 JPEG Input Timing Diagram (ITU 601 and Free run Clock Mode) 42-9
Figure 42-8 MIPI CSI DATA Alignment 42-10
Figure 42-9 IO Connection Guide 42-11
Figure 42-10 Input/Output DMA Ports 42-12
Figure 42-11 CAMIF Clock Generation 42-13
Figure 42-12 Ping-pong Memory Hierarchy 42-14
Figure 42-13 Memory Storing Style 42-15
Figure 42-14 Timing Diagram for Camera Input Register Setting 42-16
Figure 42-15 Timing Diagram for DMA input Register Setting 42-17
Figure 42-16 Timing Diagram for Last IRQ (LastIRQEn is Enabled) 42-18
Figure 42-17 Diagram for Last IRQ (LastIRQEn is Disabled) and Timing Requirement 42-19
Figure 42-18 Timing Diagram for IRQ (Input DMA Path) 42-20
Figure 42-19 Input DMA or External Camera Interface 42-21
Figure 42-20 Frame Buffer Control 42-22
Figure 42-21 Camera Window Offset Scheme 42-30
Figure 42-22 Interrupt Generation Scheme 42-34
Figure 42-23 Image Mirror and Rotation 42-43
Figure 42-24 YCbCr Plane Memory Storing Style 42-45
Figure 42-25 Scaling Scheme 42-46
Figure 42-26 I/O Timing Diagram for Direct Path 42-53
Figure 42-27 Input & Output Modes in CAMIF 42-54
Figure 42-28 Capture Frame Control 42-60
Figure 42-29 Image Effect 42-63
Figure 42-30 ENVID_M SFR Setting When Input DMA Start to Read Memory Data 42-70
Figure 42-31 SFR and Operation (Related Each DMA When Selected Input DMA Path) 42-71
Figure 42-32 Input DMA Address Change Timing (Progressive to Progressive) 42-71
Figure 42-33 Input DMA Address Change Timing (Progressive to Interlace) 42-72
Figure 42-34 Input DMA Address Change Timing (Software Update) 42-72
Figure 42-35 Input/Ouput DMA pingpong Address Change Scheme 42-73
Figure 42-36 Input DMA Progressive-in to Interlace-out (Only Interlace_Out Setting) 42-73
Figure 42-37 Input DMA Progressive-in to Interlace-out (Weave_In and Interlace_Out Setting) 42-73
Figure 42-38 Input DMA Offset and Image Size 42-78
Figure 42-39 Output DMA Offset and Image Size 42-78
Figure 42-40 Interlace YCbCr 4:2:0 Output Chroma Position 42-87
Figure 43-1 FIMC_LITE Block Diagram 43-1
Figure 43-2 Parallel Interface (ITU) Timing Diagram (RAW and YCbCr 4:2:2) 43-2
Figure 43-3 Parallel Interface (ITU) Timing Diagram (JPEG) 43-2
Figure 43-4 PARALLEL I/F Data Alignment 43-3
Figure 43-5 MIPI CSI Slave Data Alignment 43-3
Figure 43-6 Local Out I/F Data Alignment 43-4
Figure 43-7 IO Connection Guide (Parallel Interface) 43-5
Figure 43-8 Input / Output Path 43-6
Figure 43-9 Capture Frame Control 43-12
Figure 43-10 Camera Window Offset Scheme 43-13
Figure 44-1 MIPI DSI System Block Diagram 44-2
Figure 44-2 Rx Data Word Alignment 44-5
Figure 44-3 Signal Converting Diagram in Video Mode 44-6

Figure 44-4 Block Timing Diagram of HSA Mode (HSA Mode Reset: DSIM_CONFIG[20] = 0) 44-8
Figure 44-5 Block Timing Diagram of HSA Mode (HSA Mode Set: DSIM_CONFIG[20] = 1) 44-8
Figure 44-6 Block Timing Diagram of HBP Mode (HBP Mode Reset: DSIM_CONFIG[21] = 0) 44-8
Figure 44-7 Block Timing Diagram of HBP Mode (HBP Mode Set: DSIM_CONFIG[21] = 1) 44-9
Figure 44-8 Block Timing Diagram of HFP Mode (HFP Mode Reset: DSIM_CONFIG[22] = 0)..................... 44-9
Figure 44-9 Block Timing Diagram of HFP Mode (HFP Mode Set: DSIM_CONFIG[22] = 1)......................... 44-9
Figure 44-10 Block Timing Diagram of HSE Mode (HSE Mode Reset: DSIM_CONFIG[23] = 0) 44-10
Figure 44-11 Block Timing Diagram of HSE Mode (HSE Mode Set: DSIM_CONFIG[23] = 1) 44-10
Figure 44-12 Stable VFP Area Before Command Transfer Allowing Area 44-11
Figure 44-13 I80 Interface Timing Diagram 44-13
Figure 44-14 Packetizing for MIPI DSI Command Mode from I80 Interface... 44-14
Figure 45-1 CSIS0 System Block Diagram 45-2
Figure 45-2 Waveform of Output Data 45-3
Figure 45-3 MIPI CSIS Data Alignment 45-4
Figure 46-1 Data Structure of Command List 46-4
Figure 46-2 Detail Description of One Command List 46-5
Figure 46-3 Linked List of Multiple Command Lists 46-6
Figure 46-4 HOLD Scenario Examples 46-7
Figure 46-5 RGBA Format 46-10
Figure 46-6 FIMG-2D Rendering Pipeline.. 46-11
Figure 46-7 Function of Transparent Mode... 46-13
Figure 46-8 Rotation and Flip Example.. 46-16
Figure 47-1 Graphics System 47-4
Figure 47-2 Mali-400 MP GPU Top-level System 47-5
Figure 47-3 Mali-400 MP GPU Architecture... 47-6
Figure 47-4 PMU with Four Devices 47-8
Figure 47-5 Level 2 Cache Controller Hardware Architecture 47-9
Figure 48-1 Image Rotator Block Diagram.. 48-2
Figure 48-2 Ported Image Rotation Functions 48-3
Figure 48-3 Source Image Example (With Window Offset Function) 48-4
Figure 48-4 Destination Image Example (90 Degree Rotated With Window Offset Function) 48-4
Figure 49-1 Codec Structure 49-1
Figure 49-2 JPEG-Encoder Flowchart 49-18
Figure 50-1 MFC Block Diagram... 50-7
Figure 50-2 Luma and Chroma Pixel (8 bytes-aligned) 50-8
Figure 50-3 QCIF Image in 16 pixel × 16 lines (1 × 1) Tiled Mode 50-9
Figure 50-4 QCIF Image in 64 pixel × 32 lines (4 × 2) Tiled Mode 50-10
Figure 50-5 Memory Allocation 50-42
Figure 50-6 Shared Memory Input for Decoders 50-88
Figure 50-7 Shared Memory Output for Decoders.. 50-89
Figure 50-8 VC1 Parameters 50-90
Figure 50-9 Shared Memory Input for Encoders.. 50-91
Figure 50-10 Shared Memory Output for Encoders... 50-91
Figure 51-1 Video Data Path... 51-1
Figure 51-2 Block Diagram of Video Processor 51-2
Figure 51-3 Data Type for BOB 51-3
Figure 51-4 Difference Between IPC and X2 Scale-up... 51-4
Figure 51-5 Examples of Usage Cases 51-13
Figure 51-6 Endian Mode.. 51-16
Figure 51-7 Video Scaling and Positioning on TV Display... 51-19
Figure 51-8 Image Brightness and Contrast Control 51-45
Figure 51-9 4-Tap Vertical Poly-phase Filter 51-47
Figure 51-10 Pixel Repetition at Picture Boundary 51-48

Figure 52-1 Block Diagram of Mixer 52-3
Figure 52-2 TV Sub-System Block Diagram and Usage Frequency 52-4
Figure 52-3 Mixer Horizontal Scale and Blank-key 52-21
Figure 52-4 3D Frame Packing 52-27
Figure 52-5 Mixer Blending 52-29
Figure 52-6 16 bpp ARGB Example 52-30
Figure 52-7 Example of Expanding 52-30
Figure 52-8 Graphic Data Format in Memory 52-32
Figure 53-1 Block Diagram of HDMI 53-2
Figure 53-2 Block Diagram of HDMI SUB-system in Exynos 4412 SCP 53-3
Figure 53-3 Block Diagram of HDCP Key Management 53-4
Figure 53-4 Sequence of I2C Data 53-9
Figure 53-5 Block Diagram of HDMI Tx Clock Scheme in Exynos 4412 SCP 53-13
Figure 53-6 Frame Format 53-15
Figure 53-7 Sub-Frame Format 53-16
Figure 53-8 Channel Status Block Extract from SPDIF Stream 53-17
Figure 53-9 Channel Status Block 53-18
Figure 53-10 Channel Coding 53-19
Figure 53-11 Non-Linear PCM Format 53-20
Figure 53-12 Structure of Power Down Circuit 53-103
Figure 54-1 Block Diagram of SSS 54-2
Figure 54-2 DES or 3DES Only Data Flow 54-5
Figure 54-3 AES and Hash Parallel Data Flow 54-6
Figure 54-4 Data Flow of AES and Hash with Shared Input 54-7
Figure 54-5 Data Flow of Hashing the Output of AES 54-8
Figure 54-6 FIFO and FIFO Interconnections 54-9
Figure 54-7 Interrupt Controller Scheme for One DMA Interrupt Signal 54-12
Figure 54-8 AES Byte Swapping Scheme 54-14
Figure 54-9 DES Byte Swapping Scheme 54-16
Figure 54-10 Hash Byte Swapping Scheme 54-17
Figure 54-11 Flow Chart of FIFO Mode Block Cipher 54-18
Figure 54-12 Flow Chart of FIFO Mode Hashing 54-19
Figure 54-13 Flow Chart of FIFO Mode Block Cipher Combined with Hashing 54-20
Figure 54-14 AES CPU Flow Using One Buffer 54-21
Figure 54-15 AES CPU Flow Using Two Buffer 54-22
Figure 54-16 AES DMA Flow 54-23
Figure 54-17 DES/3DES CPU Flow Using One Buffer 54-24
Figure 54-18 DES/3DES CPU Flow Using Two Buffer 54-25
Figure 54-19 DES/3DES DMA Flow 54-26
Figure 54-20 Hash Operation Flow 54-27
Figure 54-21 Internal Key Storage 54-28
Figure 54-22 HMAC Key Setup 54-29
Figure 54-23 HMAC Operation 54-30
Figure 54-24 Multi-Part Hashing 54-31
Figure 54-25 Multi-Part Hashing: Part 1 54-32
Figure 54-26 Multi-Part Hashing: Part 2 to Part (N-1) 54-33
Figure 54-27 Multi-Part Hashing: Part N 54-34
Figure 54-28 Multi-Part HMAC 54-35
Figure 54-29 Multi-Part HMAC: Part 1 54-36
Figure 54-30 Multi-Part HMAC: Part 2 to Part (N-1) 54-37
Figure 54-31 Multi-Part HMAC: Part N 54-38
Figure 54-32 Seed Initialization 54-39
Figure 54-33 PRNG Operation 54-40

Figure 54-34 Concurrent Encryption/Decryption and Hash 54-41
Figure 54-35 Encryption/Decryption First, Then Hash 54-42
Figure 54-36 Operation Sequence 54-43
Figure 54-37 Operation Sequence (HMAC, Zero length Message) 54-46
Figure 54-38 Load Input A to Memory (CHNK_SZ = 0011, PREC_ID = 00) 54-57
Figure 54-39 Memory Map Definition 54-59
Figure 54-40 HASH_MSG_SIZE_LOW and HASH_MSG_SIZE_HIGH 54-97
Figure 55-1 Key Matrix Interface External Connection Guide 55-2
Figure 55-2 Internal Debouncing Filter Operation 55-3
Figure 55-3 Keypad Scanning Procedure 55-4
Figure 55-4 Keypad Scanning Procedure II 55-5
Figure 55-5 Keypad Scanning Procedure III 55-6
Figure 55-6 Keypad Scanning Procedure when the Two-key Pressed with Different Row 55-7
Figure 55-7 Keypad I/F Block Diagram 55-8
Figure 56-1 ADC Top/Bottom Offset Error Diagram 56-2
Figure 56-2 ADC Functional Block Diagram 56-3
Figure 56-3 ADC selection 56-4
Figure 56-4 Input Clock Diagram for ADC & Touch Screen Interface 56-5
Figure 57-1 Overview of Operation of TMU 57-1
Figure 57-2 Temperature Sensor Timing Diagram 57-3
Figure 57-3 Temperature Profile and Interrupt Generation 57-16
Figure 57-4 Temperature Profile by Using Emulation Mode 57-18
Figure 58-1 Basic Interface of FIMC-IS V1.5 58-2
Figure 58-2 FIMC-IS V1.5 Architecture 58-3
Figure 58-3 ISP Chain 58-4
Figure 58-4 Sync Signal Constraints 58-7
Figure 58-5 Access Sequence for Image Capturing 58-9
Figure 58-6 Interrupts Group of FIMC-IS 58-10
Figure 58-7 Interrupts for Main Host Processor and Cortex-A5 58-11
Figure 58-8 MSM Pattern 58-14
Figure 58-9 Block Diagram of MCU Controller and Interfaces 58-26
Figure 58-10 Block Diagram of MPWM 58-27
Figure 58-11 PWM Cycle Timing 58-28
Figure 58-12 Generation of One PWM Pulse 58-28
Figure 58-13 Free Trigger Mode 58-29
Figure 58-14 In case of Overlapped Interrupt 58-29
Figure 58-15 The Shapes of PWM Interrupt 58-30
Figure 58-16 ADC Functional Block Diagram 58-32
Figure 59-1 X_{TIpll} Clock Timing 59-10
Figure 59-2 EXTCLK Clock Input Timing 59-10
Figure 59-3 Manual Reset Input Timing 59-11
Figure 59-4 Power-On Reset Sequence 59-11
Figure 59-5 ROM/SRAM Timing 59-12
Figure 59-6 NAND Flash Timing 59-14
Figure 59-7 LCD Controller Timing 59-17
Figure 59-8 LCD I80 Interface Timing 59-19
Figure 59-9 Camera Interface VSYNC Timing 59-20
Figure 59-10 Camera Interface HREF Timing 59-20
Figure 59-11 Camera Interface Data Timing 59-21
Figure 59-12 Camera Interface PCLK Timing 59-21
Figure 59-13 High Speed SDMMC Interface Timing 59-23
Figure 59-14 SPI Interface Timing (CPHA = 0, CPOL = 1) 59-24
Figure 59-15 IIC Interface Timing 59-26

Figure 60-1 Exynos 4412 SCP Pin Assignment (784-FCFBGA) Bottom View 60-2
Figure 60-2 Exynos 4412 SCP Package Dimension (804-FCMSP) Top View 60-3
Figure 60-3 Exynos 4412 SCP Package Dimension (784-FCFBGA) Side View 60-4
Figure 60-4 Exynos 4412 SCP Package Marking-Top View 60-5

List of Tables

Table Number	Title	Page Number
Table 2-1	Exynos 4412 SCP 786-FCFBGA Pin Assignment – Pin Number Order	2-3
Table 2-2	Exynos 4412 SCP Signal Pin List	2-10
Table 2-3	Exynos 4412 SCP Digital I/O Power Pin List	2-28
Table 2-4	Exynos 4412 SCP Analog/HSI Power Pin List	2-29
Table 2-5	Exynos 4412 SCP Memory Power Pin List	2-30
Table 5-1	Functions Needed for Various Reset Status	5-3
Table 5-2	iROM's Clock Speed at 24 MHz ext. Crystal	5-4
Table 5-3	OM Pin Setting	5-5
Table 7-1	Operating Frequencies in Exynos 4412 SCP	7-3
Table 7-2	APLL and MPLL PMS Value	7-8
Table 7-3	EPLL PMS Value	7-9
Table 7-4	VPLL PMS Value	7-10
Table 7-5	Special Clocks in Exynos 4412 SCP	7-18
Table 7-6	I/O Clocks in Exynos 4412 SCP	7-20
Table 7-7	CLKOUT Input Clock Selection Information	7-22
Table 7-8	I/O Description	7-24
Table 8-1	Comparison of Power-saving Techniques	8-2
Table 8-2	Exynos 4412 SCP Power Domains	8-4
Table 8-3	System-Level Power Mode Summary	8-11
Table 8-4	C2C-enabled System-level Power Mode Summary	8-12
Table 8-5	System-level Power-down Configuration Registers	8-16
Table 8-6	Wake-up Sources	8-20
Table 8-7	Reset Types and their Coverage/Effect	8-23
Table 8-8	Reset Types and their Effect on PMU	8-23
Table 8-9	I/O Description	8-28
Table 9-1	GIC Configuration Values	9-3
Table 9-2	GIC Interrupt Table (SPI[127:0])	9-9
Table 9-3	GIC Interrupt Table (PPI[15:0])	9-16
Table 9-4	Clock and Reset	9-21
Table 9-5	Effect of not Implementing Some LS Priority Field Bit	9-36
Table 9-6	Priority Grouping by Binary Point	9-38
Table 9-7	Security Extension of the ICCIAR	9-52
Security of Highest Priority Pending Interrupt		9-52
ICCIAR read		9-52
ICCICR.AckCtl		9-52
Returned Interrupt ID		9-52
Table 9-8	Security Extension of the ICCEOIR	9-53
Table 9-9	Security Extension of the ICCHPIR	9-55
Table 9-10	Meaning of CPU Targets Field Bit Values	9-80
Table 9-11	Truth Table for Sending a SGI to a Target Processor	9-87
Table 10-1	Interrupt Groups of Interrupt Combiner	10-4
Table 11-1	Features of DMA Controller	11-3
Table 11-2	DMA Request Mapping Table	11-3
Table 11-3	Instruction Syntax Summary	11-15
Table 13-1	CoreSight Clock	13-9

Table 13-2 CoreSight Reset ... 13-11
Table 14-1 TZPC Table ... 14-4
Table 14-2 TZPC Transfer Attribute ... 14-6
Table 15-1 AXI Slave Interface Attributes ... 15-3
Table 15-2 AXI Master Interface Attributes ... 15-3
Table 15-3 Region Security Permissions when Security Inversion is Disabled ... 15-10
Table 15-4 Region Security Permissions when Security Inversion is Enabled ... 15-11
Table 15-5 Example OF Memory Map Along with the Register Programming ... 15-12
Table 15-6 Region Size ... 15-28
Table 16-1 The Structure of a TLB Entry ... 16-5
Table 16-2 Overview of PPMU_PPC Registers ... 16-11
Table 17-1 Structure of a TLB Entry ... 17-6
Table 17-2 Overview of PPMU_PPC Registers ... 17-14
Table 18-1 QoS Bus Master ID ... 18-12
Table 18-2 List of Performance Events ... 18-21
Table 18-3 List of Performance Information ... 18-22
Table 18-4 Enable/Disable and Start/Stop Control ... 18-23
Table 18-5 PAD Mux for Address Configuration ... 18-26
Table 22-1 ADMA Length Field ... 22-21
Table 22-2 ADMA States ... 22-23
Table 24-1 Minimum and Maximum Resolution Based on Prescaler and Clock Divider Values ... 24-5
Table 25-1 Interrupt in MCT ... 25-3
Table 27-1 Tick Interrupt Resolution ... 27-6
Table 28-1 Interrupts in Connection with FIFO ... 28-7
Table 33-1 Characteristics of Several Mobile-TV Standard Modes ... 33-3
Table 33-2 TSI I/O Description ... 33-5
Table 33-3 Sync Detection Process using Sync Byte (sync_det_cnt = 3) ... 33-10
Table 33-4 Sync Detection Process Using Sync Byte (sync_det_cnt = 3) ... 33-11
Table 35-1 Typical Usage of Master/Slave Modes ... 35-5
Table 35-2 Allowed BFS Value as BLC ... 35-10
Table 35-3 Allowed RFS Value as BFS ... 35-10
Table 35-4 Root Clock Table (MHz) ... 35-10
Table 36-1 Allowed BFS Value as BLC ... 36-8
Table 36-2 Allowed RFS Value as BFS ... 36-8
Table 36-3 Root Clock Table (MHz) ... 36-8
Table 37-1 Input Slot 1-bit Definitions ... 37-7
Table 39-1 Burst Preamble Words ... 39-8
Table 39-2 I/O Description ... 39-11
Table 41-1 Relation 16 BPP between VCLK and CLKVAL (TFT, Frequency of Video Clock Source = 60 MHz) 41-44
Table 41-2 i80 Output Mode ... 41-49
Table 41-3 RGB Interface Signals of Display Controller ... 41-51
Table 41-4 LCD Indirect i80 System Interface Signals of Display Controller ... 41-60
Table 41-5 Timing Reference Code (XY Definition) ... 41-63
Table 41-6 I/O Signals of Display Controller ... 41-64
Table 42-1 Maximum Size ... 42-3
Table 42-2 Video Timing Reference Codes of ITU-656 8-bit Format ... 42-8
Table 42-3 Sync Signal Timing Requirements ... 42-9
Table 42-4 DATA Order of YCbCr422 Align ... 42-10
Table 42-5 ITU Camera Interface Signal Description ... 42-23
Table 42-6 Color Space Conversion Equations ... 42-52
Table 42-7 FIFO Mode Image Format ... 42-53
Table 44-1 Internal Primary FIFO List ... 44-3

Table 44-2 I80 Interface Address Map 44-13
Table 44-3 Relation between Input Transactions and DSI Transactions 44-15
Table 44-4 MIPI-DPHY Interface Slave Signal 44-19
Table 44-5 PMS and Frequency Constraint 44-39
Table 44-6 AFC Codes 44-40
Table 44-7 Band Control Setting 44-40
Table 44-8 PMS value for 80 MHz 44-41
Table 44-9 PMS value for 1000 MHz 44-41
Table 44-10 PMS value for 999 MHz 44-42
Table 45-1 Timing Diagram of Output Data 45-3
Table 45-2 Data Order of YUV422 Alignment 45-4
Table 46-1 Rotation Effect 46-15
Table 46-2 Addressing Direction Effect 46-15
Table 46-3 Truth table of ROP3 46-19
Table 49-1 Color Component Data Ordering 49-3
Table 49-2 Quantization/Huffman Table 49-19
Table 50-1 Payload in the Shared Memory 50-87
Table 52-1 Graphic Blending-factor Alpha in Case of Normal Mode 52-18
Table 52-2 Graphic Blending Method 52-19
Table 53-1 HDMI LINK 2D Timing Generator Configuration Guide 53-5
Table 53-2 HDMI LINK 3D Timing Generator Configuration Guide 53-7
Table 53-3 HDMI PHY Configuration Table for 24 MHz OSC_In 53-9
Table 53-4 Frequency Summary in Use (VCLKH Usage Frequency) 53-14
Table 54-1 Register Description of FCFIFOCTRL 54-10
Table 54-2 Register Values for All Possible Use Cases 54-10
Table 54-3 Comparison of DMA Options 54-11
Table 54-4 Four Interrupt Controller Registers 54-12
Table 54-5 Bit Field Description of FCINTSTAT 54-13
Table 54-6 How to Clear Each Interrupt 54-13
Table 54-7 Change of byte Orders from Memory to AES/DES/Hash 54-15
Table 54-8 Special Cases (Hash) 54-44
Table 54-9 Special Cases (HMAC) 54-45
Table 54-10 PKA Special Function Register (PKA_SFR) List Summary 54-47
Table 54-11 PKA_SFR0 Register (Offset + 0x00) 54-48
Table 54-12 PKA_SFR1 Register (Offset + 0x04) 54-49
Table 54-13 PKA_SFR2 Register (Offset + 0x08) 54-50
Table 54-14 PKA_SFR3 Register (Offset + 0x0c) 54-53
Table 54-15 PKA_SFR4 Register (Offset + 0x10) 54-55
Table 55-1 Keypad Interface I/O Description 55-10
Table 57-1 8-bit Code Map to Temperature 57-4
Table 58-1 Selectable Data-path Configurations 58-4
Table 58-2 Selectable Data Formats 58-5
Table 58-3 Base Address of Blocks in FIMC-IS V1.5 58-5
Table 58-4 Base Address of BUS Components (SysMMU/PPMU) 58-6
Table 58-5 Sync Signal Constraints for Each Sub-IP 58-7
Table 58-6 Interrupt Pin Assignment in the Internal GIC 58-12
Table 59-1 Absolute Maximum Ratings 59-1
Table 59-2 Recommended Operating Conditions (Ta=-25 ~ 85°C and Tj=-25 ~ 125°C, unit = V) 59-2
Table 59-3 I/O DC Electrical Characteristics 59-6
Table 59-4 TC OSC Electrical Characteristics 59-7
Table 59-5 USB DC Electrical Characteristics 59-7
Table 59-6 I/O types 59-8
Table 59-7 Driver Strength of Type A IO 59-9

Table 59-8 Driver Strength of Type B IO 59-9
Table 59-9 Clock Timing Constants 59-10
Table 59-10 Power-On Reset Timing Specifications 59-11
Table 59-11 ROM/SRAM Bus Timing Constants 59-13
Table 59-12 NFCON Bus Timing Constants 59-15
Table 59-13 TFT LCD Controller Module Signal Timing Constants 59-18
Table 59-14 LCD I80 Interface Signal Timing Constants 59-19
Table 59-15 Camera Controller Module Signal Timing Constants 59-22
Table 59-16 High Speed SDMMC Interface Transmit/Receive Timing Constants 59-23
Table 59-17 SPI Interface Transmit/Receive Timing Constants (VDDext = 1.8 V) 59-25
Table 59-18 IIC BUS Controller Module Signal Timing 59-26
Table 60-1 Package Marking Line Description 60-5

List of Examples

Example Number	Title	Page Number
Example 11-1	USAGE Model	11-18
Example 15-1	Example Configuration for Subregion Disable	15-9
Example 41-1	Application	41-28
Example 41-2	Total Five Windows	41-28
Example 41-3	Blending Equation	41-29
Example 41-4	Default Blending Equation	41-30
Example 41-5	Window Blending Factor Decision	41-32
Example 41-6	Delta Structure and RGB Interface Output Order	41-59
Example 41-7	Hue Equation	41-134
Example 41-8	Coefficient Decision	41-134
Example 42-1	Image Pixel Size: 720p (1280 × 720), Format: YCbCr4:2:0 2plane (NV12)	42-84
Example 50-1	Address Calculation	50-41
Example 50-2	Shared Memory Output for Encoders	50-93
Example 50-3	Shared Memory Output for Encoders	50-94

List of Conventions

Register RW Access Type Conventions

Type	Definition	Description
R	Read Only	The application has permission to read the Register field. Writes to read-only fields have no effect.
W	Write Only	The application has permission to write in the Register field.
RW	Read & Write	The application has permission to read and writes in the Register field. The application sets this field by writing 1'b1 and clears it by writing 1'b0.

Register Value Conventions

Expression	Description
x	Undefined bit
X	Undefined multiple bits
?	Undefined, but depends on the device or pin status
Device dependent	The value depends on the device
Pin value	The value depends on the pin status

Reset Value Conventions

Expression	Description
0	Clears the register field
1	Sets the register field
x	Don't care condition

Warning: Some bits of control registers are driven by hardware or write operation only. As a result the indicated reset value and the read value after reset might be different.

1 Product Overview

1.1 Overview

Exynos 4412 SCP is a 32-bit RISC cost-effective, low power, performance optimized and Coretex-A9 Quad Core based micro-processor solution for smart phone applications. To provide optimized hardware performance for the mobile telecommunication services and general applications on smart phone, Exynos 4412 SCP adopts 64-bit/128-bit internal bus architecture and many powerful hardware accelerators for different tasks. These tasks, for example are, motion video processing, Image Signal Processing, display control and scaling. Integrated Multi Format Codec (MFC) supports encoding and decoding of MPEG-2/4, H.263, H.264 and decoding of VC1. This hardware Encoder/Decoder supports real-time video conferencing and digital TV out.

The memory system has dedicated DRAM ports and Static Memory port. The dedicated DRAM ports support DDR3 interface for high bandwidth. Static Memory Port supports NOR Flash and ROM type external memory and components.

To reduce the total system cost and enhance the overall functionality, Exynos 4412 SCP includes many hardware peripherals, such as TFT 24-bit true color LCD controller, Camera Interface, MIPI DSI, CSI-2, System Manager for power management, MIPI slimbus interface, MIPI HSI, four UARTs, 24-channel DMA, Timers, General I/O Ports, three I2S, S/PDIF, eight IIC-BUS interface, three HS-SPI, USB Host 2.0, USB 2.0 Device operating at high speed (480Mbps), two USB HSIC, four SD Host and high-speed Multimedia Card Interface, and four PLLs for clock generation.

Single Chip Package (SCP) option is available.

1.2 Block Diagram of Exynos 4412 SCP

Figure 1-1 illustrates the complete block diagram of Exynos 4412 SCP.

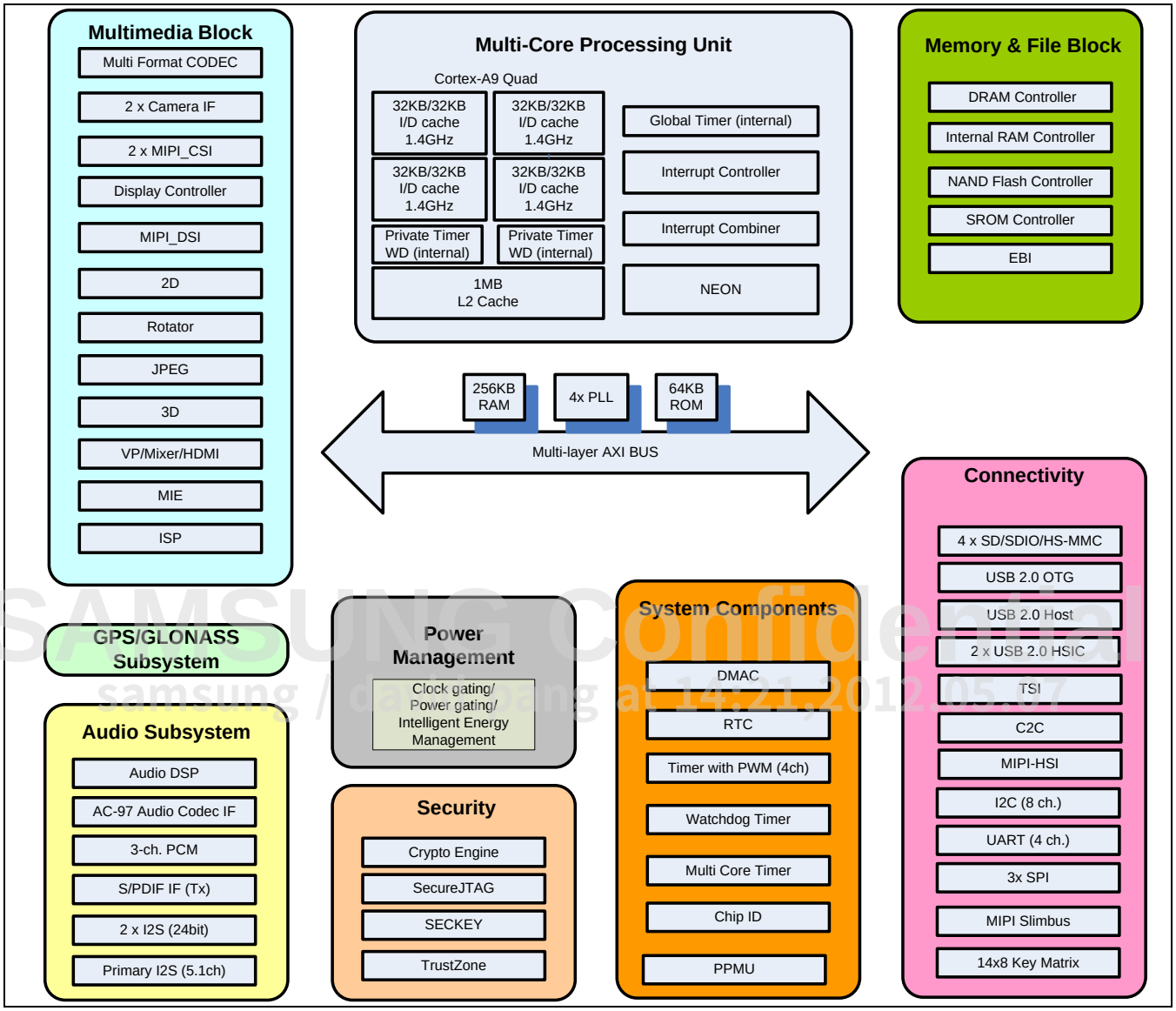


Figure 1-1 Exynos 4412 SCP Block Diagram

1.3 Features of Exynos 4412 SCP

The features of Exynos 4412 SCP are:

- ARM Cortex-A9 based Quad CPU Subsystem with NEON
  - 32/32/32/32 KB I/D Cache, 1 MB L2 Cache
  - Operating frequency up to 800 MHz at 0.9 V, 1 GHz at 1.0 V, and 1.4 GHz at 1.3 V
- 128-bit/64-bit Multi-layer bus architecture
  - Core-D domain for ARM Cortex-A9 Quad, CoreSight, and external memory interface
- Operating frequency up to 200 MHz at 1.0 V
  - Global D- domain mainly for multimedia components and external storage interfaces
- Operating frequency up to 100 MHz at 1.0 V
  - Core-P, Global-P domain mainly for other system component, such as system peripherals, peripheral DMAs, connectivity IPs and Audio interfaces.
- Operating frequency up to 100 MHz at 1.0 V
  - Audio domain for low power audio play
- Advanced power management for mobile applications
- 64 KB ROM for secure booting and 256 KB RAM for security function
- 8-bit ITU 601/656 Camera Interface supports horizontal size up to 4224 pixels for scaled and 8192 pixels for un-scaled resolution
- Multi Format Codec provides encoding and decoding of MPEG-4/H.263/H.264 up to 1080p@30 fps and decoding of MPEG-2/VC1/Xvid video up to 1080p@30fps
- Image Signal Processing subsystem
- JPEG encoder supports various format.
- 3D Graphics Acceleration with scalable Multicore GPU.
- 2D Graphics Acceleration support.
- 1/2/4/ 8bpp Palletized or 8/16/24bpp Non-Palletized Color TFT recommend up to WXGA resolution
- HDMI interface support for NTSC and PAL mode with image enhancer
- MIPI-DSI and MIPI-CSI interface support
- One AC-97 audio codec interface and 3-channel PCM serial audio interface
- Three 24-bit I2S interface support
- One TX only S/PDIF interface support for digital audio
- Eight I2C interface support
- Three SPI support
- Four UART supports three Mbps ports for Bluetooth 2.0
- On-chip USB 2.0 Device supports high-speed (480 Mbps, on-chip transceiver)
- On-chip USB 2.0 Host support
- Two on-chip USB HSIC
- Four SD/ SDIO/ HS-MMC interface support

- 24-channel DMA controller (8 channels for Memory-to-memory DMA, 16 channels for Peripheral DMA)
- Supports 14 × 8 key matrix
- Configurable GPIOs
- Real time clock, PLL, timer with PWM, and watch dog timer
- Multi-core timer support for accurate tick time in power down mode (except sleep mode)
- Memory Subsystem
  - Asynchronous SRAM/ ROM/ NOR interface with x8 or x16 data bus
  - NAND interface with x8 data bus
  - DDR3 interface (800 Mbps/pin DDR)

1.3.1 Multi-Core Processing Unit

The features of main microprocessors are:

- The ARM Cortex-A9 MPCore (quad core) processor integrates the proven and highly successful ARM MPCore technology along with further enhancements to simplify and broaden the adoption of multi-core solutions.
- With the ability to scale in speed from 200 MHz to 1.4 GHz (TBD), the ARM Cortex-A9 MPCore quad processor meets the requirements of power-optimized mobile devices, which require operation in low power and performance-optimized consumer applications. They require over 2000 Dhrystone MIPS.
- Other features of ARM Cortex-A9 MPCore quad core processor are:
  - Thumb-2 technology for greater performance, energy efficiency, and code density
  - NEONTM signal processing extensions
  - Jazelle RCT Java-acceleration technology
  - TrustZone technology for secure transactions and DRM
  - Floating-Point unit for significant acceleration for both single and double precision scalar Floating-Point operations
  - Optimized L1 caches for performance and power
  - Integrated 1 MB L2 Cache using standard compiled RAMs
  - Program Trace Macrocell and CoreSight
- Generic Interrupt Controller
  - Supports three interrupt types
    - Software Generated Interrupt (SGI)
    - Private Peripheral Interrupt (PPI)
    - Shared Peripheral Interrupt (SPI)
  - Programmable interrupts that enable to set the
    - Security state for an interrupt
    - Priority level of an interrupt
    - Enabling or disabling of an interrupt
    - Processors that receive an interrupt
  - Enhanced security features

1.3.2 Memory Subsystem

The features of memory subsystem are:

- High bandwidth Memory Matrix subsystem
- Two independent external memory ports:
  - 1x16 Static Hybrid Memory port
  - 2x32 DRAM port
- Matrix architecture increases the overall bandwidth with simultaneous access capability:
  - SRAM/ROM/NOR Interface
    - x8 or x16 data bus
    - Addresses range support: 23bit
    - Supports asynchronous interface
    - Supports byte and half-word access
  - NAND Interface
    - Supports industry standard NAND interface
    - x8 data bus
  - DDR3 interface
    - x32 data bus up to 800 Mbps/pin
    - 1.5/1.35 V interface voltage
    - Density support up to 4-Gb per port (2CS)

1.3.3 Multimedia

The features of multimedia are:

- Camera Interface
  - Multiple input support
    - ITU-R BT 601/656 mode
    - DMA (AXI 64-bit interface) mode
    - MIPI (CSI) mode
    - Direct FIFO mode (from LCDC)
  - Multiple output support
    - DMA (AXI 64-bit interface) mode
    - Direct FIFO mode (to LCDC)
  - Digital Zoom In (DZI) capability
  - Multiple camera input support
  - Programmable polarity of video sync signals
  - Input horizontal size support up to 4224 pixels for scaled and 8192 pixels for un-scaled resolution
  - Image mirror and rotation (X-axis mirror, Y-axis mirror, 90°, 180°, and 270° rotation)
  - Various image formats generation
  - Capture frame control support
  - Image effect support
- Multi-Format Video Codec (MFC)
  - ITU-T H.264, ISO/IEC 14496-10
    - Decoding supports Baseline/Main/High Profile Level 4.0 (except Flexible Macro-block Ordering (FMO), Arbitrary Slice Ordering (ASO) and Redundant Slice (RS))
    - Encoding supports Baseline/Main/High Profile (except FMO, ASO, and RS)
  - ITU-T H.263 Profile level 3
    - Decoding supports Profile3, restricted up to SD resolution 30 fps (supports H.263)
  - Annex I: Advanced Intra Coding
  - Annex J: De-blocking (in-loop) filter
  - Annex K: Slice Structured Mode without FMO & ASO
  - Annex T: Modified Quantization
  - Annex D: Unrestricted Motion Vector Mode
  - Annex F: Advanced Prediction Mode except overlapped motion compensation for luminance
    - Encoding supports Baseline Profile (supports customer size up to 1920 × 1088)
  - ISO/ IEC 14496-2 MPEG-4
    - Decoding supports MPEG-4 Simple/Advanced Simple Profile Level 5
    - Decoding supports Xvid
  - Encoding supports MPEG-4 Simple/Advanced Simple Profile

  - ISO/ IEC 13818-2 MPEG-2
    - Decoding supports Main Profile High level
    - Decoding supports MPEG-1 except D-picture
  - SMPTE 421M VC-1
    - Decoding supports Simple Profile Medium Level/ Main Profile High Level/Advanced Profile Level
- JPEG Codec supports:
  - Compression/Decompression up to 65536 × 65536
  - Supported format of compression
    - Input raw image: YCbCr4:2:2 or RGB 565
    - Output JPEG file: Baseline JPEG of YCbCr4:2:2 or YCbCr4:2:0
  - General-purpose color-space converter
- 3D Graphic Engine supports:
  - 3D graphics and vector graphics based on programmable processors
  - Tile based pixel processing
  - Scalable multi-core pixel processors
  - Advanced shader feature set and industry stand API support
  - OGL-ES 1.1 and 2.0, Open VG 1.1
  - Full featured MMUs for all processor cores
  - On-chip tile 24-bit fixed point depth buffer
  - 8-bit Stencil with on-chip tile stencil buffer
  - 4-level hierarchical Z and stencil operations
  - Fast dynamic branching
  - Cube map, projected, non square texture support
  - Bi-linear, tri-linear texture filtering
  - Anti-aliasing: penalty-free 4x multi-sampling, up to 512x FSAA (limited to 16x FSAA by driver)
  - Indexed and non-indexed geometry input
- 2D Graphic Engine supports:
  - BitBLT
  - Maximum 8000 × 8000 image size
  - Window clipping, 90°/180°/270°/Rotation, X Flip/Y Flip
  - Totally 4-operand raster operation (ROP4)
  - Alpha blending (user-specified constant alpha value/per-pixel alpha value)
  - 8/16/24/32-bpp. Packed 24-bpp color format, Premultiplied/Non-premultiplied alpha format
  - 1 bpp/4 bpp/8 bpp/16 bpp/32 bpp Mask format, YCbCr format

- Digital TV Interface supports:
  - High-Definition Multimedia Interface (HDMI) 1.4 a
  - Up to 1080 p 60 Hz and 8-channel/112 kHz/24-bit audio
  - 480 p, 576 p, 720 p, 1080i (cannot support 480i)
  - HDCP V1.1
  - 3D support
- Rotator
  - Supported image format: YCbCr422 (Interleave), YCbCr420 (Non-interleave), and RGB565 and RGB888 (unpacked)
  - Supported rotate degree: 90, 180, 270, flip vertical, and flip horizontal
- Video processor: The video processor supports:
  - BOB/ 2D-IPC mode
  - Production of YCbCr 4: 4: 4 output to help the mixer blend video and graphics
  - 1/4X to 16X vertical scaling with 4-tap/16-phase polyphase filter
  - 1/4X to 16X horizontal scaling with 8-tap/16-phase polyphase filter
  - Pan and scan, Letterbox, and NTSC/PAL conversion using scaling
  - Flexible scaled video positioning within display area
  - 1/16 pixel resolution Pan and Scan modes
  - Flexible post video processing
    - Color saturation, brightness/contrast enhancement, edge enhancement
    - Color space conversion between BT.601 and BT.709
  - Video input source size up to 1920 ×1080
- Video Mixer
  The Video Mixer supports:
  - Overlapping and blending input video and graphic layers
  - 480p, 576p, 720p, and 1080i/p display size
  - Four layers (1 video layer, 2 graphic layer, and 1 background layer)
- TFT-LCD Interface
  The TFT-LCD Interface supports:
  - 24/18/16-bpp parallel RGB Interface LCD
  - 8/6 bpp serial RGB Interface
  - Dual i80 Interface LCD
  - 1/2/4/8 bpp Palletized or 8/16/24-bpp Non-Palletized Color TFT
  - Typical actual screen size: 1080 × 1024,1024 × 768, 800 × 480, 640 × 480, 320 × 240, 160 × 160, and so on
  - Virtual image up to 16M pixel (4K pixel × 4K pixel)
  - Five Window Layers for PIP or OSD
  - Real-time overlay plane multiplexing
  - Programmable OSD window positioning
  - 16-level alpha blending

1.3.4 Audio Subsystem

The features of audio subsystem are:

- Reconfigurable Processor (RP) progresses audio processing
- Low power audio subsystem
  - 5.1 channel I2S with 32-bit-width 64-depth FIFO
  - 128 KB audio play output buffer
  - Hardware mixer mixes primary and secondary sounds

1.3.5 Image Signal Processing Subsystem

The features of ISP subsystem are:

- Dual camera input
- Image signal processing
- Dynamic range correction
- Face detection

1.3.6 Security Subsystem

The features of security subsystem are:

- On-chip secure boot ROM
  - 64KB secure boot ROM for secure boot
- On-chip secure RAM
  - 256KB secure RAM for security function
- Hardware Crypto Accelerator
  - Securely integrated DES/TDES, AES, SHA-256, PRNG and PKA
  - Access control (Security Domain Manager with the ARM TrustZone Hardware)
  - Enables enhanced secure platform for separate (Secure/Non-secure) execution environment for security sensitive application
- Secure JTAG
  - Authentication of JTAG user
  - Access control in JTAG mode

1.3.7 Connectivity

The features of connectivity are:

- PCM Audio Interface supports:
  - 16-bit mono audio interface
  - Master mode only
  - 3-port PCM interface
- AC97 Audio Interface supports:
  - Independent channels for stereo PCM In, stereo PCM Out, and mono MIC In
  - 16-bit stereo (2-channel) audio
  - Variable sampling rate AC97 Codec interface (48 kHz and below)
  - AC97 full specification
- SPDIF Interface (TX only) supports:
  - Linear PCM up to 24-bit per sample support
  - Non-Linear PCM formats such as AC3, MPEG1, and MPEG2 support
  - 2x24-bit buffers that are alternately filled with data
- I2S Bus Interface supports:
  - Three I2S-bus for audio-codec interface with DMA-based operation
  - Serial and 8/16/24-bit per channel data transfers
  - I2S, MSB-justified, and LSB-justified data format
  - PCM 5.1 channel
  - Various bit clock frequency and codec clock frequency support
    - 16, 24, 32, and 48fs of bit clock frequency
    - 256, 384, 512, and 768fs of codec clock
  - One port for 5.1 channel I2S (in audio subsystem) and two ports for 2-channel I2S
- I2C Bus Interface supports:
  - Eight Multi-Master IIC-Bus
  - Serial, 8-bit oriented and bi-directional data transfers can be made at up to 100 Kbps in the standard mode
  - Up to 400 Kbps in the fast mode
- MIPI-Slimbus Interface supports:
  - 6 ports. Each port has 16 entry FIFO with 32-bit width
- UART supports:
  - Four UART with DMA-based or interrupt-based operation
  - 5-bit, 6-bit, 7-bit, or 8-bit serial data transmit/ receive
  - Rx/Tx independent 256nbyte FIFO for UART0, 64 byte FIFO for UART1, and 16 byte FIFO for UART2/3/4
  - Programmable baud rate
  - IrDA 1.0 SIR (115.2 Kbps) mode
  - Loop back mode for testing
  - Non-integer clock divides in Baud clock generation

- USB 2.0 Device supports:
  - Complies to USB 2.0 Specification (Revision 1.0a)High-speed up to 480 Mbps
  - On-chip USB transceiver
- USB Host 2.0 supports:
  - With the USB Host 2.0
  - High-speed up to 480 Mbps
  - On-chip USB transceiver
- HS-MMC/ SDIO Interface supports:
  - Multimedia Card Protocol version 4.3 compatible (HS-MMC)
  - SD Memory Card Protocol version 2.0 compatible
  - DMA based or interrupt based operation
  - 128 word FIFO for Tx/Rx
  - Four ports HS-MMC or four ports SDIO
- SPI Interface supports:
  - With three Serial Peripheral Interface Protocol version 2.11
  - Rx/Tx independent 64-Word FIFO for SPI0 and 16-Word FIFO for SPI1
  - DMA-based or interrupt-based operation
- GPIO.

1.3.8 System Peripheral

The features of system peripheral are:

- Real Time Clock
  - Full clock features: sec, min, hour, date, day, month, and year
  - 32.768 kHz operation
  - Alarm interrupt
  - Time-tick interrupt
- PLL
  - Four on-chip PLLs and APLL/MPLL/EPLL/VPLL
  - APLL generates ARM core and MSYS clocks
  - MPLL generates a system bus clock and special clocks
  - EPLL generates special clocks
  - VPLL generates clocks for video interface
- Keypad
  - 14 × 8 Key Matrix support
  - Provides internal de-bounce filter
- Timer with Pulse Width Modulation
  - Five channel 32-bit internal timer with interrupt-based operation
  - Three channel 32-bit Timer with PWM
  - Programmable duty cycle, frequency, and polarity
  - Dead-zone generation
  - Supports external clock source
- Multi-Core timer
  - 64-bit global timer with four independent count comparators
  - Two 31-bit local timers
- It can change interrupt interval without stopping reference tick timer DMA:
  - Micro-code programming based DMA
  - The specific instruction set provides flexibility to program DMA transfers
  - Supports linked list DMA function
  - Supports three enhanced built-in DMA with eight channels per DMA, so the total number of channels it supports are 32
  - Supports one Memory-to-memory type optimized DMA and two Peripheral-to-memory type optimized DMA
  - M2M DMA supports up to 16 burst and P2M DMA supports up to 8 burst
- Watch Dog Timer
  - 16-bit watch dog timer

- Thermal Management Unit (TMU)
- Power Management
  - Clock-gating control for components
  - Various low power modes are available, such as Idle, Stop, Deep Stop, Deep Idle, and Sleep modes
  - Wake up sources in sleep mode are:
    - External interrupts
    - RTC alarm
    - Tick timer
    - Key interface
  - Wake up sources of Stop and Deep Stop mode are:
    - MMC
    - Touch screen interface
    - System timer
    - Entire wake up sources of Sleep mode
  - Wake up sources of Deep Idle mode are:
    - 5.1 channel I2S
    - Wake up source of Stop mode

1.4 Conventions

1.4.1 Register R/W Conventions

Symbol	Definition	Description
R	Read Only	The application has permission to read the register field. Writes to read-only fields have no effect.
W	Write Only	The application has permission to write in the Register field.
R/W	Read and Write	The application has permission to read and writes in the Register field. The application sets this field by writing 1'b1 and clears it by writing 1'b0.
R/WC	Read and Write to clear	The application has permission to read and write in the register field. The application clears this field by writing 1'b1. A register write of 1'b0 has no effect on this field.
R/WS	Read and Write to set	The application has permission to read and write in the register field. The application sets this field by writing 1'b1. A register write of 1'b0 has no effect on this field.

1.4.2 Register Value Conventions

Expression	Description
x	Undefined bit
X	Undefined multiple bits
?	Undefined but depends on the device, or pin status
Device dependent	The value depends on the device
Pin value	The value depends on the pin status

2 Ball Map and Description

2.1 Overview

This chapter describes the Ball Map of Exynos 4412 SCP.

2.2 Pin Map

2.2.1 Block Diagram of 786-ball FCFBGA (SCP)

Figure 2-1 illustrates the Exynos 4412 SCP pin map (786-FCFBGA) top view.

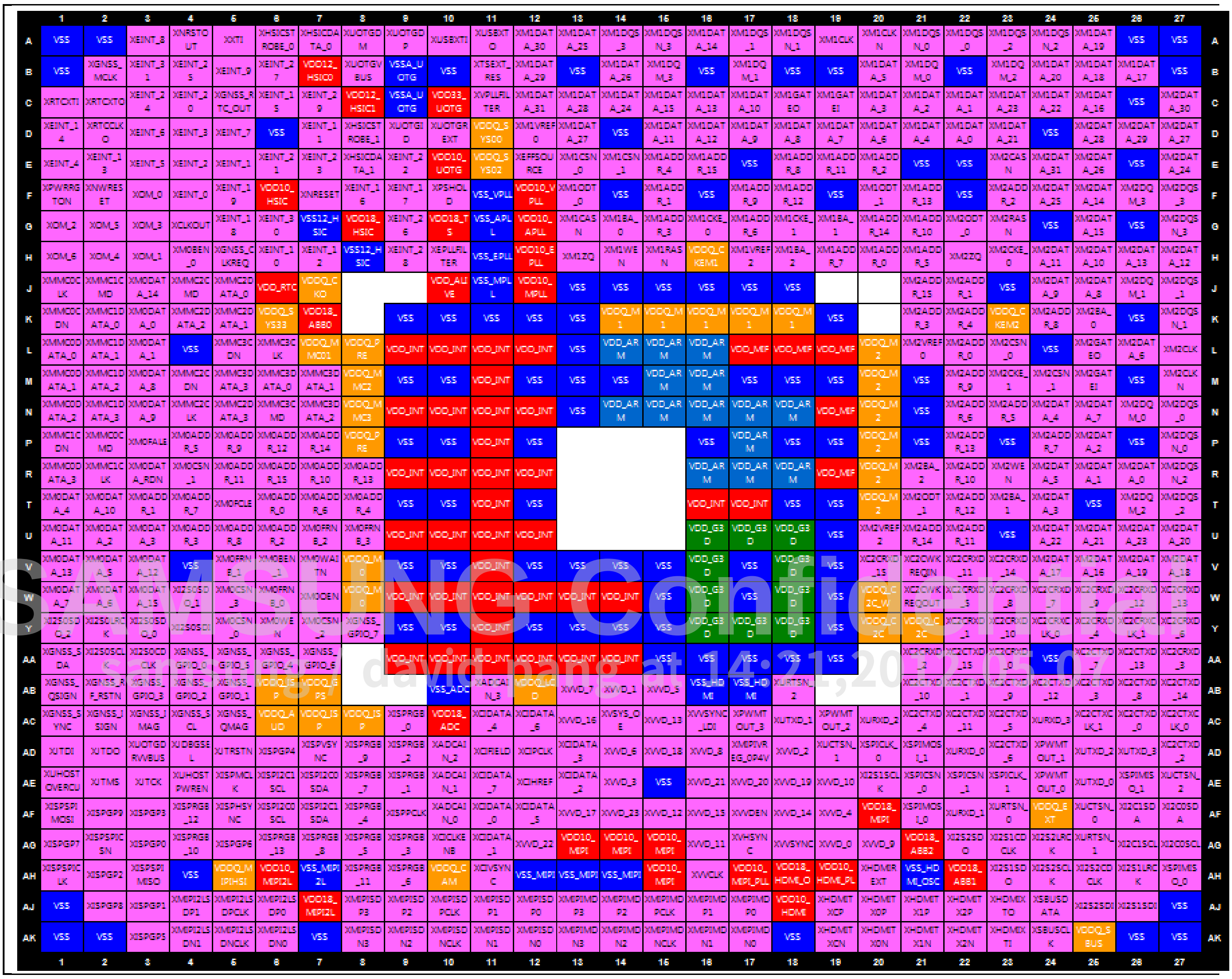


Figure 2-1 Exynos 4412 SCP Pin Map (786-FCFBGA) Top View

2.3 Pin Assignments and Pin Number Order

Table 2-1 describes the Exynos 4412 SCP pin assignment and pin number order.

Table 2-1 Exynos 4412 SCP 786-FCFBGA Pin Assignment – Pin Number Order

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
A1	VSS	H9	XEINT_28	T2	XM0DATA_10	AC22	XC2CTXD_11
A2	VSS	H10	XEPLLFILTER	T3	XM0ADDR_1	AC23	XC2CTXD_5
A3	XEINT_8	H11	VSS_EPLL	T4	XM0ADDR_7	AC24	XURXD_3
A4	XNRSTOUT	H12	VDD10_EPLL	T5	XM0FCLE	AC25	XC2CTXCLK_1
A5	XXTI	H13	XM1ZQ	T6	XM0ADDR_0	AC26	XC2CTXD_0
A6	XHSICSTROBE_0	H14	XM1WEN	T7	XM0ADDR_6	AC27	XC2CTXCLK_0
A7	XHSICDATA_0	H15	XM1RASN	T8	XM0ADDR_4	AD1	XJTDI
A8	XUOTGDM	H16	VDDQ_CKEM1	T9	VSS	AD2	XJTDO
A9	XUOTGDP	H17	XM1VREF2	T10	VSS	AD3	XUOTGDRVVBUS
A10	XUSBXTI	H18	XM1BA_2	T11	VDD_INT	AD4	XJDBGSEL
A11	XUSBXTO	H19	XM1ADDR_7	T12	VSS	AD5	XJTRSTN
A12	XM1DATA_30	H20	XM1ADDR_0	T16	VDD_INT	AD6	XISPGP4
A13	XM1DATA_25	H21	XM1ADDR_5	T17	VDD_INT	AD7	XISPVSYNC
A14	XM1DQS_3	H22	XM2ZQ	T18	VSS	AD8	XISPRGB_9
A15	XM1DQSN_3	H23	XM2CKE_0	T19	VSS	AD9	XISPRGB_2
A16	XM1DATA_14	H24	XM2DATA_11	T20	VDDQ_M2	AD10	XADCAIN_2
A17	XM1DQS_1	H25	XM2DATA_10	T21	XM2ODT_1	AD11	XCIFIELD
A18	XM1DQSN_1	H26	XM2DATA_13	T22	XM2ADDR_12	AD12	XCIPCLK
A19	XM1CLK	H27	XM2DATA_12	T23	XM2BA_1	AD13	XCIDATA_3
A20	XM1CLKN	J1	XMMC0CLK	T24	XM2DATA_3	AD14	XVVD_6
A21	XM1DQSN_0	J2	XMMC1CMD	T25	VSS	AD15	XVVD_18
A22	XM1DQS_0	J3	XM0DATA_14	T26	XM2DQM_2	AD16	XVVD_8
A23	XM1DQS_2	J4	XMMC2CMD	T27	XM2DQS_2	AD17	XMIPIVREG_0P4V
A24	XM1DQSN_2	J5	XMMC2DATA_0	U1	XM0DATA_11	AD18	XVVD_2
A25	XM1DATA_19	J6	VDD_RTC	U2	XM0DATA_2	AD19	XUCTSN_1
A26	VSS	J7	VDDQ_CKO	U3	XM0DATA_3	AD20	XSPICLK_0
A27	VSS	J10	VDD_ALIVE	U4	XM0ADDR_3	AD21	XSPIMOSI_1
B1	VSS	J11	VSS_MPLL	U5	XM0ADDR_8	AD22	XURXD_0
B2	XGNSS_MCLK	J12	VDD10_MPLL	U6	XM0ADDR_2	AD23	XC2CTXD_6
B3	XEINT_31	J13	VSS	U7	XM0FRNB_2	AD24	XPWMTOUT_1
B4	XEINT_25	J14	VSS	U8	XM0FRNB_3	AD25	XUTXD_2
B5	XEINT_9	J15	VSS	U9	VDD_INT	AD26	XUTXD_3
B6	XEINT_27	J16	VSS	U10	VDD_INT	AD27	XC2CTXD_2

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
B7	VDD12_HSIC0	J17	VSS	U11	VDD_INT	AE1	XUHOSTOVERCUR
B8	XUOTGVBUS	J18	VSS	U12	VDD_INT	AE2	XJTMS
B9	VSSA_UOTG	J21	XM2ADDR_15	U16	VDD_G3D	AE3	XJTCK
B10	VSS	J22	XM2ADDR_1	U17	VDD_G3D	AE4	XUHOSTPWREN
B11	XTSEXT_RES	J23	VSS	U18	VDD_G3D	AE5	XISPMCLK
B12	XM1DATA_29	J24	XM2DATA_9	U19	VSS	AE6	XISPI2C1SCL
B13	VSS	J25	XM2DATA_8	U20	XM2VREF2	AE7	XISPI2C0SDA
B14	XM1DATA_26	J26	XM2DQM_1	U21	XM2ADDR_14	AE8	XISPRGB_7
B15	XM1DQM_3	J27	XM2DQS_1	U22	XM2ADDR_11	AE9	XISPRGB_1
B16	VSS	K1	XMMC0CDN	U23	VSS	AE10	XADCAIN_1
B17	XM1DQM_1	K2	XMMC1DATA_0	U24	XM2DATA_22	AE11	XCIDATA_7
B18	VSS	K3	XM0DATA_0	U25	XM2DATA_21	AE12	XCIHREF
B19	VSS	K4	XMMC2DATA_2	U26	XM2DATA_23	AE13	XCIDATA_2
B20	XM1DATA_5	K5	XMMC2DATA_1	U27	XM2DATA_20	AE14	XVVD_3
B21	XM1DQM_0	K6	VDDQ_SYS33	V1	XM0DATA_13	AE15	VSS
B22	VSS	K7	VDD18_ABB0	V2	XM0DATA_5	AE16	XVVD_21
B23	XM1DQM_2	K9	VSS	V3	XM0DATA_12	AE17	XVVD_20
B24	XM1DATA_20	K10	VSS	V4	VSS	AE18	XVVD_19
B25	XM1DATA_18	K11	VSS	V5	XM0FRNB_1	AE19	XVVD_10
B26	XM1DATA_17	K12	VSS	V6	XM0BEN_1	AE20	XI2S1SCLK
B27	VSS	K13	VSS	V7	XM0WAITN	AE21	XSPICSN_0
C1	XRTCXTI	K14	VDDQ_M1	V8	VDDQ_M0	AE22	XSPICSN_1
C2	XRTCXTO	K15	VDDQ_M1	V9	VSS	AE23	XSPICLK_1
C3	XEINT_24	K16	VDDQ_M1	V10	VSS	AE24	XPWMTOUT_0
C4	XEINT_20	K17	VDDQ_M1	V11	VDD_INT	AE25	XUTXD_0
C5	XGNSS_RTC_OUT	K18	VDDQ_M1	V12	VSS	AE26	XSPIMISO_1
C6	XEINT_15	K19	VSS	V13	VSS	AE27	XUCTSN_2
C7	XEINT_29	K21	XM2ADDR_3	V14	VSS	AF1	XISPSPIMOSI
C8	VDD12_HSIC1	K22	XM2ADDR_4	V15	VSS	AF2	XISPGP9
C9	VSSA_UOTG	K23	VDDQ_CKEM2	V16	VDD_G3D	AF3	XISPGP3
C10	VDD33_UOTG	K24	XM2ADDR_8	V17	VSS	AF4	XISPRGB_12
C11	XVPLLFILTER	K25	XM2BA_0	V18	VDD_G3D	AF5	XISPHSYNC
C12	XM1DATA_31	K26	VSS	V19	VSS	AF6	XISPI2C0SCL
C13	XM1DATA_28	K27	XM2DQSN_1	V20	XC2CRXD_15	AF7	XISPI2C1SDA
C14	XM1DATA_24	L1	XMMC0DATA_0	V21	XC2CWKREQIN	AF8	XISPRGB_4
C15	XM1DATA_15	L2	XMMC1DATA_1	V22	XC2CRXD_11	AF9	XISPPCLK
C16	XM1DATA_13	L3	XM0DATA_1	V23	XC2CRXD_14	AF10	XADCAIN_0

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
C17	XM1DATA_10	L4	VSS	V24	XM2DATA_17	AF11	XCIDATA_0
C18	XM1GATEO	L5	XMMC3CDN	V25	XM2DATA_16	AF12	XCIDATA_5
C19	XM1GATEI	L6	XMMC3CLK	V26	XM2DATA_19	AF13	XVVD_17
C20	XM1DATA_3	L7	VDDQ_MMC01	V27	XM2DATA_18	AF14	XVVD_23
C21	XM1DATA_2	L8	VDDQ_PRE	W1	XM0DATA_7	AF15	XVVD_12
C22	XM1DATA_1	L9	VDD_INT	W2	XM0DATA_6	AF16	XVVD_15
C23	XM1DATA_23	L10	VDD_INT	W3	XM0DATA_15	AF17	XVVDEN
C24	XM1DATA_22	L11	VDD_INT	W4	XI2S0SDO_1	AF18	XVVD_14
C25	XM1DATA_16	L12	VDD_INT	W5	XM0CSN_3	AF19	XVVD_4
C26	VSS	L13	VSS	W6	XM0FRNB_0	AF20	VDD18_MIPI
C27	XM2DATA_30	L14	VDD_ARM	W7	XM0OEN	AF21	XSPIMOSI_0
D1	XEINT_14	L15	VDD_ARM	W8	VDDQ_M0	AF22	XURXD_1
D2	XRTCCLKO	L16	VDD_ARM	W9	VDD_INT	AF23	XURTSN_0
D3	XEINT_6	L17	VDD_MIF	W10	VDD_INT	AF24	VDDQ_EXT
D4	XEINT_3	L18	VDD_MIF	W11	VDD_INT	AF25	XUCTSN_0
D5	XEINT_7	L19	VDD_MIF	W12	VDD_INT	AF26	XI2C1SDA
D6	VSS	L20	VDDQ_M2	W13	VDD_INT	AF27	XI2C0SDA
D7	XEINT_11	L21	XM2VREF0	W14	VDD_INT	AG1	XISPGP7
D8	XHSICSTROBE_1	L22	XM2ADDR_0	W15	VSS	AG2	XISPSPICSN
D9	XUOTGID	L23	XM2CSN_0	W16	VDD_G3D	AG3	XISPGP0
D10	XUOTGREXT	L24	VSS	W17	VSS	AG4	XISPRGB_10
D11	VDDQ_SYS00	L25	XM2GATEO	W18	VDD_G3D	AG5	XISPGP6
D12	XM1VREF0	L26	XM2DATA_6	W19	VSS	AG6	XISPRGB_13
D13	XM1DATA_27	L27	XM2CLK	W20	VDDQ_C2C_W	AG7	XISPRGB_8
D14	VSS	M1	XMMC0DATA_1	W21	XC2CWKREQOUT	AG8	XISPRGB_5
D15	XM1DATA_11	M2	XMMC1DATA_2	W22	XC2CRXD_5	AG9	XISPRGB_3
D16	XM1DATA_12	M3	XM0DATA_8	W23	XC2CRXD_8	AG10	XCICLKENB
D17	XM1DATA_9	M4	XMMC2CDN	W24	XC2CRXD_7	AG11	XCIDATA_1
D18	XM1DATA_8	M5	XMMC3DATA_3	W25	XC2CRXD_9	AG12	XVVD_22
D19	XM1DATA_7	M6	XMMC3DATA_0	W26	XC2CRXD_12	AG13	VDD10_MIPI
D20	XM1DATA_6	M7	XMMC3DATA_1	W27	XC2CRXD_13	AG14	VDD10_MIPI
D21	XM1DATA_4	M8	VDDQ_MMC2	Y1	XI2S0SDO_2	AG15	VDD10_MIPI
D22	XM1DATA_0	M9	VSS	Y2	XI2S0LRCK	AG16	XVVD_11
D23	XM1DATA_21	M10	VSS	Y3	XI2S0SDO_0	AG17	XVHSYNC
D24	VSS	M11	VDD_INT	Y4	XI2S0SDI	AG18	XVVSYNC
D25	XM2DATA_28	M12	VSS	Y5	XM0CSN_0	AG19	XVVD_0
D26	XM2DATA_29	M13	VSS	Y6	XM0WEN	AG20	XVVD_9

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
D27	XM2DATA_27	M14	VSS	Y7	XM0CSN_2	AG21	VDD18_ABB2
E1	XEINT_4	M15	VDD_ARM	Y8	XGNSS_GPIO_7	AG22	XI2S2SDO
E2	XEINT_13	M16	VDD_ARM	Y9	VSS	AG23	XI2S1CDCLK
E3	XEINT_5	M17	VSS	Y10	VSS	AG24	XI2S2LRCK
E4	XEINT_2	M18	VSS	Y11	VDD_INT	AG25	XURTSN_1
E5	XEINT_1	M19	VSS	Y12	VSS	AG26	XI2C1SCL
E6	XEINT_21	M20	VDDQ_M2	Y13	VSS	AG27	XI2C0SCL
E7	XEINT_23	M21	VSS	Y14	VSS	AH1	XISPSPICLK
E8	XHSICDATA_1	M22	XM2ADDR_9	Y15	VSS	AH2	XISPGP2
E9	XEINT_22	M23	XM2CKE_1	Y16	VDD_G3D	AH3	XISPSPIMISO
E10	VDD10_UOTG	M24	XM2CSN_1	Y17	VDD_G3D	AH4	VSS
E11	VDDQ_SYS02	M25	XM2GATEI	Y18	VDD_G3D	AH5	VDDQ_MIPIHSI
E12	XEFFSOURCE	M26	VSS	Y19	VSS	AH6	VDD10_MIPI2L
E13	XM1CSN_0	M27	XM2CLKN	Y20	VDDQ_C2C	AH7	VSS_MIPI2L
E14	XM1CSN_1	N1	XMMC0DATA_2	Y21	VDDQ_C2C	AH8	XISPRGB_11
E15	XM1ADDR_4	N2	XMMC1DATA_3	Y22	XC2CRXD_1	AH9	XISPRGB_6
E16	XM1ADDR_15	N3	XM0DATA_9	Y23	XC2CRXD_10	AH10	VDDQ_CAM
E17	VSS	N4	XMMC2CLK	Y24	XC2CRXCLK_0	AH11	XCIVSYNC
E18	XM1ADDR_8	N5	XMMC2DATA_3	Y25	XC2CRXD_4	AH12	VSS_MIPI
E19	XM1ADDR_11	N6	XMMC3CMD	Y26	XC2CRXCLK_1	AH13	VSS_MIPI
E20	XM1ADDR_2	N7	XMMC3DATA_2	Y27	XC2CRXD_6	AH14	VSS_MIPI
E21	VSS	N8	VDDQ_MMC3	AA1	XGNSS_SDA	AH15	VDD10_MIPI
E22	VSS	N9	VDD_INT	AA2	XI2S0SCLK	AH16	XVVCLK
E23	XM2CASN	N10	VDD_INT	AA3	XI2S0CDCLK	AH17	VDD10_MIPI_PLL
E24	XM2DATA_31	N11	VDD_INT	AA4	XGNSS_GPIO_0	AH18	VDD18_HDMI_OSC
E25	XM2DATA_26	N12	VDD_INT	AA5	XGNSS_GPIO_5	AH19	VDD10_HDMI_PLL
E26	VSS	N13	VSS	AA6	XGNSS_GPIO_4	AH20	XHDMIREXT
E27	XM2DATA_24	N14	VDD_ARM	AA7	XGNSS_GPIO_6	AH21	VSS_HDMI_OSC
F1	XPWRRGTON	N15	VDD_ARM	AA9	VDD_INT	AH22	VDD18_ABB1
F2	XNWRESET	N16	VDD_ARM	AA10	VDD_INT	AH23	XI2S1SDO
F3	XOM_0	N17	VDD_ARM	AA11	VDD_INT	AH24	XI2S2SCLK
F4	XEINT_0	N18	VDD_ARM	AA12	VDD_INT	AH25	XI2S2CDCLK
F5	XEINT_19	N19	VDD_MIF	AA13	VDD_INT	AH26	XI2S1LRCK
F6	VDD10_HSIC	N20	VDDQ_M2	AA14	VDD_INT	AH27	XSPIMISO_0
F7	XNRESET	N21	VSS	AA15	VSS	AJ1	VSS
F8	XEINT_16	N22	XM2ADDR_6	AA16	VSS	AJ2	XISPGP8
F9	XEINT_17	N23	XM2ADDR_5	AA17	VSS	AJ3	XISPGP1

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
F10	XPSHOLD	N24	XM2DATA_4	AA18	VSS	AJ4	XMIPI2LSDP1
F11	VSS_VPLL	N25	XM2DATA_7	AA19	VSS	AJ5	XMIPI2LSDPCLK
F12	VDD10_VPLL	N26	XM2DQM_0	AA21	XC2CRXD_2	AJ6	XMIPI2LSDP0
F13	XM1ODT_0	N27	XM2DQS_0	AA22	XC2CTXD_15	AJ7	VDD18_MIPI2L
F14	VSS	P1	XMMC1CDN	AA23	XC2CRXD_0	AJ8	XMIPISDP3
F15	XM1ADDR_1	P2	XMMC0CMD	AA24	VSS	AJ9	XMIPISDP2
F16	VSS	P3	XM0FALE	AA25	XC2CTXD_7	AJ10	XMIPISDPCLK
F17	XM1ADDR_9	P4	XM0ADDR_5	AA26	XC2CTXD_13	AJ11	XMIPISDP1
F18	XM1ADDR_12	P5	XM0ADDR_9	AA27	XC2CRXD_3	AJ12	XMIPISDP0
F19	VSS	P6	XM0ADDR_12	AB1	XGNSS_QSIGN	AJ13	XMIPIMDP3
F20	XM1ODT_1	P7	XM0ADDR_14	AB2	XGNSS_RF_RSTN	AJ14	XMIPIMDP2
F21	XM1ADDR_13	P8	VDDQ_PRE	AB3	XGNSS_GPIO_3	AJ15	XMIPIMDPCLK
F22	VSS	P9	VSS	AB4	XGNSS_GPIO_2	AJ16	XMIPIMDP1
F23	XM2ADDR_2	P10	VSS	AB5	XGNSS_GPIO_1	AJ17	XMIPIMDP0
F24	XM2DATA_25	P11	VDD_INT	AB6	VDDQ_ISP	AJ18	VDD10_HDMI
F25	XM2DATA_14	P12	VSS	AB7	VDDQ_GPS	AJ19	XHDMITXCP
F26	XM2DQM_3	P16	VSS	AB10	VSS_ADC	AJ20	XHDMITX0P
F27	XM2DQS_3	P17	VDD_ARM	AB11	XADCAIN_3	AJ21	XHDMITX1P
G1	XOM_2	P18	VSS	AB12	VDDQ_LCD	AJ22	XHDMITX2P
G2	XOM_5	P19	VSS	AB13	XVVD_7	AJ23	XHDMIXTO
G3	XOM_3	P20	VDDQ_M2	AB14	XVVD_1	AJ24	XSBUSDATA
G4	XCLKOUT	P21	VSS	AB15	XVVD_5	AJ25	XI2S2SDI
G5	XEINT_18	P22	XM2ADDR_13	AB16	VSS_HDMI	AJ26	XI2S1SDI
G6	XEINT_30	P23	VSS	AB17	VSS_HDMI	AJ27	VSS
G7	VSS12_HSIC	P24	XM2ADDR_7	AB18	XURTSN_2	AK1	VSS
G8	VDD18_HSIC	P25	XM2DATA_2	AB21	XC2CTXD_10	AK2	VSS
G9	XEINT_26	P26	VSS	AB22	XC2CTXD_1	AK3	XISPGP5
G10	VDD18_TS	P27	XM2DQSN_0	AB23	XC2CTXD_9	AK4	XMIPI2LSDN1
G11	VSS_APLL	R1	XMMC0DATA_3	AB24	XC2CTXD_12	AK5	XMIPI2LSDNCLK
G12	VDD10_APLL	R2	XMMC1CLK	AB25	XC2CTXD_3	AK6	XMIPI2LSDN0
G13	XM1CASN	R3	XM0DATA_RDN	AB26	XC2CTXD_8	AK7	VSS
G14	XM1BA_0	R4	XM0CSN_1	AB27	XC2CTXD_14	AK8	XMIPISDN3
G15	XM1ADDR_3	R5	XM0ADDR_11	AC1	XGNSS_SYNC	AK9	XMIPISDN2
G16	XM1CKE_0	R6	XM0ADDR_15	AC2	XGNSS_ISIGN	AK10	XMIPISDNCLK
G17	XM1ADDR_6	R7	XM0ADDR_10	AC3	XGNSS_IMAG	AK11	XMIPISDN1
G18	XM1CKE_1	R8	XM0ADDR_13	AC4	XGNSS_SCL	AK12	XMIPISDN0
G19	XM1BA_1	R9	VDD_INT	AC5	XGNSS_QMAG	AK13	XMIPIMDN3

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
G20	XM1ADDR_14	R10	VDD_INT	AC6	VDDQ_AUD	AK14	XMIPIMDN2
G21	XM1ADDR_10	R11	VDD_INT	AC7	VDDQ_ISP	AK15	XMIPIMDNCLK
G22	XM2ODT_0	R12	VDD_INT	AC8	VDDQ_ISP	AK16	XMIPIMDN1
G23	XM2RASN	R16	VDD_ARM	AC9	XISPRGB_0	AK17	XMIPIMDN0
G24	VSS	R17	VDD_ARM	AC10	VDD18_ADC	AK18	VSS
G25	XM2DATA_15	R18	VDD_ARM	AC11	XCIDATA_4	AK19	XHDMITXCN
G26	VSS	R19	VDD_MIF	AC12	XCIDATA_6	AK20	XHDMITX0N
G27	XM2DQSN_3	R20	VDDQ_M2	AC13	XVVD_16	AK21	XHDMITX1N
H1	XOM_6	R21	XM2BA_2	AC14	XVSYS_OE	AK22	XHDMITX2N
H2	XOM_4	R22	XM2ADDR_10	AC15	XVVD_13	AK23	XHDMIXTI
H3	XOM_1	R23	XM2WEN	AC16	XVVSYNC_LDI	AK24	XSBUSCLK
H4	XM0BEN_0	R24	XM2DATA_5	AC17	XPWMTOUT_3	AK25	VDDQ_SBUS
H5	XGNSS_CLKREQ	R25	XM2DATA_1	AC18	XUTXD_1	AK26	VSS
H6	XEINT_10	R26	XM2DATA_0	AC19	XPWMTOUT_2	AK27	VSS
H7	XEINT_12	R27	XM2DQSN_2	AC20	XURXD_2		
H8	VSS12_HSIC	T1	XM0DATA_4	AC21	XC2CTXD_4		

2.4 I/O Control Type Conventions

I/O Control Type	Description
A1	Control at power down mode is possible, power down mode is released by S/W (GPIO pad except Alive, DRAM, Audio, C2C, UART, MMC, and Memory)
A2	Control at power down mode is possible, power down mode is released by S/W (UART pad)
A3	Control at power down mode is possible, power down mode is released by S/W (EBIA pad)
A4	Control at power down mode is possible, power down mode is released by S/W (EBIB pad)
A5	Control at power down mode is possible, power down mode is released by S/W (MMC0/1 pad)
A6	Control at power down mode is possible, power down mode is released by S/W (MMC2/3 pad)
A7	Control at power down mode is possible, power down mode is released by H/W automatically (Audio pad)
A8	Control at power down mode is impossible, power down mode is released by Pin (DRAM CKE)
A9	Control at power down mode is possible, power down mode is released by S/W (C2C pad)
B1	No Retention (Alive I/O)
B2	No Retention (Analog I/O)
B3	No Retention (DRAM I/O)

2.5 Pin Description

2.5.1 Signal Pin

Table 2-2 describes the Exynos 4412 SCP signal pin list.

Table 2-2 Exynos 4412 SCP Signal Pin List

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
XuRXD_0	GPA0[0]/UART_0_RXD	PD	I	A1	VDDQ_EXT
XuTXD_0	GPA0[1]/UART_0_TXD	PD	I	A1	VDDQ_EXT
XuCTSn_0	GPA0[2]/UART_0_CTSn	PD	I	A1	VDDQ_EXT
XuRTSn_0	GPA0[3]/UART_0_RTSn	PD	I	A1	VDDQ_EXT
XuRXD_1	GPA0[4]/UART_1_RXD	PD	I	A1	VDDQ_EXT
XuTXD_1	GPA0[5]/UART_1_TXD	PD	I	A1	VDDQ_EXT
XuCTSn_1	GPA0[6]/UART_1_CTSn/I2C_2_SDA	PD	I	A1	VDDQ_EXT
XuRTSn_1	GPA0[7]/UART_1_RTSn/I2C_2_SCL	PD	I	A1	VDDQ_EXT
XuRXD_2	GPA1[0]/UART_2_RXD/UART_AUDIO_RXD	PD	I	A1	VDDQ_EXT
XuTXD_2	GPA1[1]/UART_2_TXD/UART_AUDIO_TXD	PD	I	A1	VDDQ_EXT
XuCTSn_2	GPA1[2]/UART_2_CTSn/I2C_3_SDA	PD	I	A1	VDDQ_EXT
XuRTSn_2	GPA1[3]/UART_2_RTSn/I2C_3_SCL	PD	I	A1	VDDQ_EXT
XuRXD_3	GPA1[4]/UART_3_RXD/UART_AUDIO_RXD	PD	I	A2	VDDQ_EXT
XuTXD_3	GPA1[5]/UART_3_TXD/UART_AUDIO_TXD	PD	I	A2	VDDQ_EXT
XspiCLK_0	GPB[0]/SPI_0_CLK/I2C_4_SDA	PD	I	A1	VDDQ_EXT
XspiCSn_0	GPB[1]/SPI_0_nSS/I2C_4_SCL	PD	I	A1	VDDQ_EXT
XspiMISO_0	GPB[2]/SPI_0_MISO/I2C_5_SDA	PD	I	A1	VDDQ_EXT
XspiMOSI_0	GPB[3]/SPI_0_MOSI/I2C_5_SCL	PD	I	A1	VDDQ_EXT
XspiCLK_1	GPB[4]/SPI_1_CLK/IEM_SCLK	PD	I	A1	VDDQ_EXT
XspiCSn_1	GPB[5]/SPI_1_nSS/IEM_SPWI	PD	I	A1	VDDQ_EXT
XspiMISO_1	GPB[6]/SPI_1_MISO	PD	I	A1	VDDQ_EXT
XspiMOSI_1	GPB[7]/SPI_1_MOSI	PD	I	A1	VDDQ_EXT
Xi2s1SCLK	GPC0[0]/I2S_1_SCLK/PCM_1_SCLK/AC97BITCLK	PD	I	A1	VDDQ_EXT
Xi2s1CDCLK	GPC0[1]/I2S_1_CDCLK/PCM_1_EXTCLK/AC97RESETn	PD	I	A1	VDDQ_EXT
Xi2s1LRCK	GPC0[2]/I2S_1_LRCK/PCM_1_FSYNC/AC97SYNC	PD	I	A1	VDDQ_EXT
Xi2s1SDI	GPC0[3]/I2S_1_SDI/PCM_1	PD	I	A1	VDDQ_EXT

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
	_SIN/AC97SDI				
Xi2s1SDO	GPC0[4]/I2S_1_SDO/PCM_1 _SOUT/AC97SDO	PD	I	A1	VDDQ_EXT
Xi2s2SCLK	GPC1[0]/I2S_2_SCLK/PCM_2 _SCLK/SPDIF_0_OUT	PD	I	A1	VDDQ_EXT
Xi2s2CDCLK	GPC1[1]/I2S_2_CDCLK/PCM_2 _EXTCLK/SPDIF_EXTCLK/SPI_2_CLK	PD	I	A1	VDDQ_EXT
Xi2s2LRCK	GPC1[2]/I2S_2_LRCK/PCM_2 _FSYNC/SPI_2_nSS	PD	I	A1	VDDQ_EXT
Xi2s2SDI	GPC1[3]/I2S_2_SDI/PCM_2_SIN/I2C _6_SDA/SPI_2_MISO	PD	I	A1	VDDQ_EXT
Xi2s2SDO	GPC1[4]/I2S_2_SDO/PCM_2 _SOUT/I2C_6_SCL/SPI_2_MOSI	PD	I	A1	VDDQ_EXT
XpwmTOUT_0	GPD0[0]/TOUT_0/LCD_FRM	PD	I	A1	VDDQ_EXT
XpwmTOUT_1	GPD0[1]/TOUT_1/LCD_PWM	PD	I	A1	VDDQ_EXT
XpwmTOUT_2	GPD0[2]/TOUT_2/I2C_7_SDA	PD	I	A1	VDDQ_EXT
XpwmTOUT_3	GPD0[3]/TOUT_3/I2C_7_SCL	PD	I	A1	VDDQ_EXT
Xi2c0SDA	GPD1[0]/I2C_0_SDA/MIPI0_BYTE_CLK	PD	I	A1	VDDQ_EXT
Xi2c0SCL	GPD1[1]/I2C_0_SCL/MIPI0_ESC_CLK	PD	I	A1	VDDQ_EXT
Xi2c1SDA	GPD1[2]/I2C_1_SDA	PD	I	A1	VDDQ_EXT
Xi2c1SCL	GPD1[3]/I2C_1_SCL	PD	I	A1	VDDQ_EXT
XvHSYNC	GPF0[0]/LCD_HSYNC	PD	I	A1	VDDQ_LCD
XvVSYNC	GPF0[1]/LCD_VSYNC	PD	I	A1	VDDQ_LCD
XvVDEN	GPF0[2]/LCD_VDEN	PD	I	A1	VDDQ_LCD
XvVCLK	GPF0[3]/LCD_VCLK	PD	I	A1	VDDQ_LCD
XvVD_0	GPF0[4]/LCD_VD[0]	PD	I	A1	VDDQ_LCD
XvVD_1	GPF0[5]/LCD_VD[1]	PD	I	A1	VDDQ_LCD
XvVD_2	GPF0[6]/LCD_VD[2]	PD	I	A1	VDDQ_LCD
XvVD_3	GPF0[7]/LCD_VD[3]	PD	I	A1	VDDQ_LCD
XvVD_4	GPF1[0]/LCD_VD[4]	PD	I	A1	VDDQ_LCD
XvVD_5	GPF1[1]/LCD_VD[5]	PD	I	A1	VDDQ_LCD
XvVD_6	GPF1[2]/LCD_VD[6]	PD	I	A1	VDDQ_LCD
XvVD_7	GPF1[3]/LCD_VD[7]	PD	I	A1	VDDQ_LCD
XvVD_8	GPF1[4]/LCD_VD[8]	PD	I	A1	VDDQ_LCD
XvVD_9	GPF1[5]/LCD_VD[9]	PD	I	A1	VDDQ_LCD
XvVD_10	GPF1[6]/LCD_VD[10]	PD	I	A1	VDDQ_LCD
XvVD_11	GPF1[7]/LCD_VD[11]	PD	I	A1	VDDQ_LCD

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
XvVD_12	GPF2[0]/LCD_VD[12]	PD	I	A1	VDDQ_LCD
XvVD_13	GPF2[1]/LCD_VD[13]	PD	I	A1	VDDQ_LCD
XvVD_14	GPF2[2]/LCD_VD[14]	PD	I	A1	VDDQ_LCD
XvVD_15	GPF2[3]/LCD_VD[15]	PD	I	A1	VDDQ_LCD
XvVD_16	GPF2[4]/LCD_VD[16]	PD	I	A1	VDDQ_LCD
XvVD_17	GPF2[5]/LCD_VD[17]	PD	I	A1	VDDQ_LCD
XvVD_18	GPF2[6]/LCD_VD[18]	PD	I	A1	VDDQ_LCD
XvVD_19	GPF2[7]/LCD_VD[19]	PD	I	A1	VDDQ_LCD
XvVD_20	GPF3[0]/LCD_VD[20]	PD	I	A1	VDDQ_LCD
XvVD_21	GPF3[1]/LCD_VD[21]	PD	I	A1	VDDQ_LCD
XvVD_22	GPF3[2]/LCD_VD[22]	PD	I	A1	VDDQ_LCD
XvVD_23	GPF3[3]/LCD_VD[23]	PD	I	A1	VDDQ_LCD
XvVSYNC_LDI	GPF3[4]/VSYNC_LDI	PD	I	A1	VDDQ_LCD
XvSYS_OE	GPF3[5]/SYS_OE	PD	I	A1	VDDQ_LCD
XsbusDATA	ETC1[0]/SLIMbusData	PD	I	A1	VDDQ_SBUS
XsbusCLK	ETC1[1]/SLIMbusClk	PD	I	A1	VDDQ_SBUS
XciPCLK	GPJ0[0]/CAM_A_PCLK	PD	I	A1	VDDQ_CAM
XciVSYNC	GPJ0[1]/CAM_A_VSYNC	PD	I	A1	VDDQ_CAM
XciHREF	GPJ0[2]/CAM_A_HREF	PD	I	A1	VDDQ_CAM
XciDATA_0	GPJ0[3]/CAM_A_DATA[0]	PD	I	A1	VDDQ_CAM
XciDATA_1	GPJ0[4]/CAM_A_DATA[1]	PD	I	A1	VDDQ_CAM
XciDATA_2	GPJ0[5]/CAM_A_DATA[2]	PD	I	A1	VDDQ_CAM
XciDATA_3	GPJ0[6]/CAM_A_DATA[3]	PD	I	A1	VDDQ_CAM
XciDATA_4	GPJ0[7]/CAM_A_DATA[4]	PD	I	A1	VDDQ_CAM
XciDATA_5	GPJ1[0]/CAM_A_DATA[5]	PD	I	A1	VDDQ_CAM
XciDATA_6	GPJ1[1]/CAM_A_DATA[6]	PD	I	A1	VDDQ_CAM
XciDATA_7	GPJ1[2]/CAM_A_DATA[7]	PD	I	A1	VDDQ_CAM
XciCLKenb	GPJ1[3]/CAM_A_CLKOUT	PD	I	A1	VDDQ_CAM
XciFIELD	GPJ1[4]/CAM_A_FIELD	PD	I	A1	VDDQ_CAM
XabbPBBG_1	XabbPBBG_1	–	IO	B2	VDD18_ABB1
XabbPBBG_2	XabbPBBG_2	–	IO	B2	VDD18_ABB2
XabbPBBG_3	XabbPBBG_3	–	IO	B2	VDD18_ABB3
XhdmiTX0P	HDMI_TX0P	–	O	B2	VDD18_HDMI_OSC
XhdmiTX0N	HDMI_TX0N	–	O	B2	VDD18_HDMI_OSC

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
XhdmiTX1P	HDMI_TX1P	–	O	B2	VDD18_HDMI_OSC
XhdmiTX1N	HDMI_TX1N	–	O	B2	VDD18_HDMI_OSC
XhdmiTX2P	HDMI_TX2P	–	O	B2	VDD18_HDMI_OSC
XhdmiTX2N	HDMI_TX2N	–	O	B2	VDD18_HDMI_OSC
XhdmiTXCP	HDMI_TXCP	–	O	B2	VDD18_HDMI_OSC
XhdmiTXCN	HDMI_TXCN	–	O	B2	VDD18_HDMI_OSC
XhdmiREXT	HDMI_REXT	–	I	B2	VDD18_HDMI_OSC
XhdmiXTI	HDMI_XI	–	I	B2	VDD18_HDMI_OSC
XhdmiXTO	HDMI_XO	–	O	B2	VDD18_HDMI_OSC
XmipiMDP0	MIPI_MDP_0	–	IO	B2	VDD18_MIPI
XmipiMDP1	MIPI_MDP_1	–	IO	B2	VDD18_MIPI
XmipiMDP2	MIPI_MDP_2	–	IO	B2	VDD18_MIPI
XmipiMDP3	MIPI_MDP_3	–	IO	B2	VDD18_MIPI
XmipiMDN0	MIPI_MDN_0	–	IO	B2	VDD18_MIPI
XmipiMDN1	MIPI_MDN_1	–	IO	B2	VDD18_MIPI
XmipiMDN2	MIPI_MDN_2	–	IO	B2	VDD18_MIPI
XmipiMDN3	MIPI_MDN_3	–	IO	B2	VDD18_MIPI
XmipiSDP0	MIPI_SDP_0	–	IO	B2	VDD18_MIPI
XmipiSDP1	MIPI_SDP_1	–	IO	B2	VDD18_MIPI
XmipiSDP2	MIPI_SDP_2	–	IO	B2	VDD18_MIPI
XmipiSDP3	MIPI_SDP_3	–	IO	B2	VDD18_MIPI
XmipiSDN0	MIPI_SDN_0	–	IO	B2	VDD18_MIPI
XmipiSDN1	MIPI_SDN_1	–	IO	B2	VDD18_MIPI
XmipiSDN2	MIPI_SDN_2	–	IO	B2	VDD18_MIPI
XmipiSDN3	MIPI_SDN_3	–	IO	B2	VDD18_MIPI
XmipiMDPCLK	MIPI_CLK_TX_P	–	IO	B2	VDD18_MIPI
XmipiMDNCLK	MIPI_CLK_TX_N	–	IO	B2	VDD18_MIPI
XmipiSDPCLK	MIPI_CLK_RX_P	–	IO	B2	VDD18_MIPI
XmipiSDNCLK	MIPI_CLK_RX_N	–	IO	B2	VDD18_MIPI

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
XmipiVREG_0P4V	MIPI_Reg_cap	–	IO	B2	VDD18_MIPI
Xmipi2LSDP0	MIPI_2L_SDP_0	–	IO	B2	VDD18_MIPI2L
Xmipi2LSDP1	MIPI_2L_SDP_1	–	IO	B2	VDD18_MIPI2L
Xmipi2LSDN0	MIPI_2L_SDN_0	–	IO	B2	VDD18_MIPI2L
Xmipi2LSDN1	MIPI_2L_SDN_1	–	IO	B2	VDD18_MIPI2L
Xmipi2LSDPCLK	MIPI_2L_CLK_RX_P	–	IO	B2	VDD18_MIPI2L
Xmipi2LSDNCLK	MIPI_2L_CLK_RX_N	–	IO	B2	VDD18_MIPI2L
XadcAIN_0	XadcAIN[0]	–	I	B2	VDD18_ADC
XadcAIN_1	XadcAIN[1]	–	I	B2	VDD18_ADC
XadcAIN_2	XadcAIN[2]	–	I	B2	VDD18_ADC
XadcAIN_3	XadcAIN[3]	–	I	B2	VDD18_ADC
Xmmc0CLK	GPK0[0]/SD_0_CLK/SD_4_CLK	PD	I	A5	VDDQ_MMC01
Xmmc0CMD	GPK0[1]/SD_0_CMD/SD_4_CMD	PD	I	A5	VDDQ_MMC01
Xmmc0CDn	GPK0[2]/SD_0_CDn/SD_4_CDn/GNSS _GPIO[8]	PD	I	A5	VDDQ_MMC01
Xmmc0DATA_0	GPK0[3]/SD_0_DATA[0]/SD_4_DATA[0]	PD	I	A5	VDDQ_MMC01
Xmmc0DATA_1	GPK0[4]/SD_0_DATA[1]/SD_4_DATA[1]	PD	I	A5	VDDQ_MMC01
Xmmc0DATA_2	GPK0[5]/SD_0_DATA[2]/SD_4_DATA[2]	PD	I	A5	VDDQ_MMC01
Xmmc0DATA_3	GPK0[6]/SD_0_DATA[3]/SD_4_DATA[3]	PD	I	A5	VDDQ_MMC01
Xmmc1CLK	GPK1[0]/SD_1_CLK	PD	I	A5	VDDQ_MMC01
Xmmc1CMD	GPK1[1]/SD_1_CMD	PD	I	A5	VDDQ_MMC01
Xmmc1CDn	GPK1[2]/SD_1_CDn/GNSS _GPIO[9]/SD_4_nRESET_OUT	PD	I	A5	VDDQ_MMC01
Xmmc1DATA_0	GPK1[3]/SD_1_DATA[0]/SD_0 _DATA[4]/SD_4_DATA[4]	PD	I	A5	VDDQ_MMC01
Xmmc1DATA_1	GPK1[4]/SD_1_DATA[1]/SD_0 _DATA[5]/SD_4_DATA[5]	PD	I	A5	VDDQ_MMC01
Xmmc1DATA_2	GPK1[5]/SD_1_DATA[2]/SD_0 _DATA[6]/SD_4_DATA[6]	PD	I	A5	VDDQ_MMC01
Xmmc1DATA_3	GPK1[6]/SD_1_DATA[3]/SD_0 _DATA[7]/SD_4_DATA[7]	PD	I	A5	VDDQ_MMC01
Xmmc2CLK	GPK2[0]/SD_2_CLK	PD	I	A6	VDDQ_MMC2
Xmmc2CMD	GPK2[1]/SD_2_CMD	PD	I	A6	VDDQ_MMC2
Xmmc2CDn	GPK2[2]/SD_2_CDn/GNSS_GPIO[10]	PD	I	A6	VDDQ_MMC2
Xmmc2DATA_0	GPK2[3]/SD_2_DATA[0]	PD	I	A6	VDDQ_MMC2
Xmmc2DATA_1	GPK2[4]/SD_2_DATA[1]	PD	I	A6	VDDQ_MMC2
Xmmc2DATA_2	GPK2[5]/SD_2_DATA[2]	PD	I	A6	VDDQ_MMC2

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
Xmmc2DATA_3	GPK2[6]/SD_2_DATA[3]	PD	I	A6	VDDQ_MMC2
Xmmc3CLK	GPK3[0]/SD_3_CLK	PD	I	A6	VDDQ_MMC3
Xmmc3CMD	GPK3[1]/SD_3_CMD	PD	I	A6	VDDQ_MMC3
Xmmc3CDn	GPK3[2]/SD_3_CDn/GNSS_GPIO[11]	PD	I	A6	VDDQ_MMC3
Xmmc3DATA_0	GPK3[3]/SD_3_DATA[0]/SD_2_DATA[4]	PD	I	A6	VDDQ_MMC3
Xmmc3DATA_1	GPK3[4]/SD_3_DATA[1]/SD_2_DATA[5]	PD	I	A6	VDDQ_MMC3
Xmmc3DATA_2	GPK3[5]/SD_3_DATA[2]/SD_2_DATA[6]	PD	I	A6	VDDQ_MMC3
Xmmc3DATA_3	GPK3[6]/SD_3_DATA[3]/SD_2_DATA[7]	PD	I	A6	VDDQ_MMC3
XGNSS_SYNC	GPL0[0]/GNSS_SYNC	PD	I	A1	VDDQ_GPS
XGNSS_ISIGN	GPL0[1]/GNSS_ISIGN	PD	I	A1	VDDQ_GPS
XGNSS_IMAG	GPL0[2]/GNSS_IMAG	PD	I	A1	VDDQ_GPS
XGNSS_QSIGN	GPL0[3]/GNSS_QSIGN	PD	I	A1	VDDQ_GPS
XGNSS_QMAG	GPL0[4]/GNSS_QMAG	PD	I	A1	VDDQ_GPS
XGNSS_RF_RSTN	GPL0[6]/GNSS_RF_RSTN	PD	I	A1	VDDQ_GPS
XGNSS_SCL	GPL1[0]/GNSS_SCL	PD	I	A1	VDDQ_GPS
XGNSS_SDA	GPL1[1]/GNSS_SDA	PD	I	A1	VDDQ_GPS
XGNSS_GPIO_0	GPL2[0]/GNSS_GPIO[0]/KP_COL[0]	PD	I	A1	VDDQ_GPS
XGNSS_GPIO_1	GPL2[1]/GNSS_GPIO[1]/KP_COL[1]	PD	I	A1	VDDQ_GPS
XGNSS_GPIO_2	GPL2[2]/GNSS_GPIO[2]/KP_COL[2]	PD	I	A1	VDDQ_GPS
XGNSS_GPIO_3	GPL2[3]/GNSS_GPIO[3]/KP_COL[3]	PD	I	A1	VDDQ_GPS
XGNSS_GPIO_4	GPL2[4]/GNSS_GPIO[4]/KP_COL[4]	PD	I	A1	VDDQ_GPS
XGNSS_GPIO_5	GPL2[5]/GNSS_GPIO[5]/KP_COL[5]	PD	I	A1	VDDQ_GPS
XGNSS_GPIO_6	GPL2[6]/GNSS_GPIO[6]/KP_COL[6]	PD	I	A1	VDDQ_GPS
XGNSS_GPIO_7	GPL2[7]/GNSS_GPIO[7]/KP_COL[7]	PD	I	A1	VDDQ_GPS
XEINT_0	GPX0[0]/WAKEUP_INT0[0]/AUD _TCK/GNSS_TCK/ALV_TCK	PD	I	B1	VDDQ_SYS33
XEINT_1	GPX0[1]/WAKEUP_INT0[1]/AUD _TMS/GNSS_TMS/ALV_TMS	PD	I	B1	VDDQ_SYS33
XEINT_2	GPX0[2]/WAKEUP_INT0[2]/AUD _TDI/GNSS_TDI/ALV_TDI	PD	I	B1	VDDQ_SYS33
XEINT_3	GPX0[3]/WAKEUP_INT0[3]/AUD _TDO/GNSS_TDO/ALV_TDO	PD	I	B1	VDDQ_SYS33
XEINT_4	GPX0[4]/WAKEUP_INT0[4]/AUD _TRSTn/GNSS_TRSTn/ALV_DBG[0]	PD	I	B1	VDDQ_SYS33
XEINT_5	GPX0[5]/WAKEUP_INT0[5]/ALV_DBG[1]	PD	I	B1	VDDQ_SYS33
XEINT_6	GPX0[6]/WAKEUP_INT0[6]/ALV_DBG[2]	PD	I	B1	VDDQ_SYS33
XEINT_7	GPX0[7]/WAKEUP_INT0[7]/ALV_DBG[3]	PD	I	B1	VDDQ_SYS33

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
XEINT_8	GPX1[0]/WAKEUP_INT1[0]/KP _COL[0]/ALV_DBG[4]	PD	I	B1	VDDQ_SYS00
XEINT_9	GPX1[1]/WAKEUP_INT1[1]/KP _COL[1]/ALV_DBG[5]	PD	I	B1	VDDQ_SYS00
XEINT_10	GPX1[2]/WAKEUP_INT1[2]/KP _COL[2]/ALV_DBG[6]	PD	I	B1	VDDQ_SYS00
XEINT_11	GPX1[3]/WAKEUP_INT1[3]/KP _COL[3]/ALV_DBG[7]	PD	I	B1	VDDQ_SYS00
XEINT_12	GPX1[4]/WAKEUP_INT1[4]/KP _COL[4]/ALV_DBG[8]	PD	I	B1	VDDQ_SYS00
XEINT_13	GPX1[5]/WAKEUP_INT1[5]/KP _COL[5]/ALV_DBG[9]	PD	I	B1	VDDQ_SYS00
XEINT_14	GPX1[6]/WAKEUP_INT1[6]/KP _COL[6]/ALV_DBG[10]	PD	I	B1	VDDQ_SYS00
XEINT_15	GPX1[7]/WAKEUP_INT1[7]/KP _COL[7]/ALV_DBG[11]	PD	I	B1	VDDQ_SYS00
XEINT_16	GPX2[0]/WAKEUP_INT2[0]/KP _ROW[0]/ALV_DBG[12]	PD	I	B1	VDDQ_SYS00
XEINT_17	GPX2[1]/WAKEUP_INT2[1]/KP _ROW[1]/ALV_DBG[13]	PD	I	B1	VDDQ_SYS00
XEINT_18	GPX2[2]/WAKEUP_INT2[2]/KP _ROW[2]/ALV_DBG[14]	PD	I	B1	VDDQ_SYS00
XEINT_19	GPX2[3]/WAKEUP_INT2[3]/KP _ROW[3]/ALV_DBG[15]	PD	I	B1	VDDQ_SYS00
XEINT_20	GPX2[4]/WAKEUP_INT2[4]/KP _ROW[4]/ALV_DBG[16]	PD	I	B1	VDDQ_SYS00
XEINT_21	GPX2[5]/WAKEUP_INT2[5]/KP _ROW[5]/ALV_DBG[17]	PD	I	B1	VDDQ_SYS00
XEINT_22	GPX2[6]/WAKEUP_INT2[6]/KP _ROW[6]/ALV_DBG[18]	PD	I	B1	VDDQ_SYS00
XEINT_23	GPX2[7]/WAKEUP_INT2[7]/KP _ROW[7]/ALV_DBG[19]	PD	I	B1	VDDQ_SYS00
XEINT_24	GPX3[0]/WAKEUP_INT3[0]/KP _ROW[8]/ALV_DBG[20]	PD	I	B1	VDDQ_SYS00
XEINT_25	GPX3[1]/WAKEUP_INT3[1]/KP _ROW[9]/ALV_DBG[21]	PD	I	B1	VDDQ_SYS00
XEINT_26	GPX3[2]/WAKEUP_INT3[2]/KP _ROW[10]/ALV_DBG[22]	PD	I	B1	VDDQ_SYS00
XEINT_27	GPX3[3]/WAKEUP_INT3[3]/KP _ROW[11]/ALV_DBG[23]	PD	I	B1	VDDQ_SYS00
XEINT_28	GPX3[4]/WAKEUP_INT3[4]/KP _ROW[12]/ALV_DBG[24]	PD	I	B1	VDDQ_SYS00

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
XEINT_29	GPX3[5]/WAKEUP_INT3[5]/KP _ROW[13]/ALV_DBG[25]	PD	I	B1	VDDQ_SYS00
XEINT_30	GPX3[6]/WAKEUP_INT3[6]/HDMI _CEC/ALV_DBG[26]	PD	I	B1	VDDQ_SYS00
XEINT_31	GPX3[7]/WAKEUP_INT3[7]/HDMI _HPD/ALV_DBG[27]	PD	I	B1	VDDQ_SYS00
Xm0CSn_0	GPY0[0]/SROM_CSn[0]/NF_CSn[2]	PU	I	A3	VDDQ_M0
Xm0CSn_1	GPY0[1]/SROM_CSn[1]/NF_CSn[3]	PU	I	A3	VDDQ_M0
Xm0CSn_2	GPY0[2]/SROM_CSn[2]/NF _CSn[0]/OND_CSn[0]	PU	I	A3	VDDQ_M0
Xm0CSn_3	GPY0[3]/SROM_CSn[3]/NF _CSn[1]/OND_CSn[1]	PU	I	A3	VDDQ_M0
Xm0OEn	GPY0[4]/EBI_OEn	PU	I	A3	VDDQ_M0
Xm0WEn	GPY0[5]/EBI_WEn	PU	I	A3	VDDQ_M0
Xm0BEn_0	GPY1[0]/EBI_BEn[0]	PU	I	A3	VDDQ_M0
Xm0BEn_1	GPY1[1]/EBI_BEn[1]	PU	I	A3	VDDQ_M0
Xm0WAITn	GPY1[2]/SROM_WAITn	PU	I	A3	VDDQ_M0
Xm0DATA_RDn	GPY1[3]/EBI_DATA_RDn	PU	I	A3	VDDQ_M0
Xm0FCLE	GPY2[0]/NF_CLE/OND_ADDRVALID	PU	I	A3	VDDQ_M0
Xm0FALE	GPY2[1]/NF_ALE/OND_SMCLK	PU	I	A3	VDDQ_M0
Xm0FRnB_0	GPY2[2]/NF_RnB[0]/OND_INT[0]	PU	I	A3	VDDQ_M0
Xm0FRnB_1	GPY2[3]/NF_RnB[1]/OND_INT[1]	PU	I	A3	VDDQ_M0
Xm0FRnB_2	GPY2[4]/NF_RnB[2]/OND_RPn	PU	I	A3	VDDQ_M0
Xm0FRnB_3	GPY2[5]/NF_RnB[3]	PU	I	A3	VDDQ_M0
Xm0ADDR_0	GPY3[0]/EBI_ADDR[0]	PD	I	A4	VDDQ_M0
Xm0ADDR_1	GPY3[1]/EBI_ADDR[1]	PD	I	A4	VDDQ_M0
Xm0ADDR_2	GPY3[2]/EBI_ADDR[2]	PD	I	A4	VDDQ_M0
Xm0ADDR_3	GPY3[3]/EBI_ADDR[3]	PD	I	A4	VDDQ_M0
Xm0ADDR_4	GPY3[4]/EBI_ADDR[4]	PD	I	A4	VDDQ_M0
Xm0ADDR_5	GPY3[5]	PD	I	A4	VDDQ_M0
Xm0ADDR_6	GPY3[6]/EBI_ADDR[6]	PD	I	A4	VDDQ_M0
Xm0ADDR_7	GPY3[7]/EBI_ADDR[7]	PD	I	A4	VDDQ_M0
Xm0ADDR_8	GPY4[0]/EBI_ADDR[8]/XhsiCAWAKE	PD	I	A4	VDDQ_MIPIHSI
Xm0ADDR_9	GPY4[1]/EBI_ADDR[9]/XhsiCADATA	PD	I	A4	VDDQ_MIPIHSI
Xm0ADDR_10	GPY4[2]/EBI_ADDR[10]/XhsiCAFLAG	PD	I	A4	VDDQ_MIPIHSI
Xm0ADDR_11	GPY4[3]/EBI_ADDR[11]/XhsiACREADY	PD	I	A4	VDDQ_MIPIHSI
Xm0ADDR_12	GPY4[4]/EBI_ADDR[12]/XhsiACWAKE	PD	I	A4	VDDQ_MIPIHSI

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
Xm0ADDR_13	GPY4[5]/EBI_ADDR[13]/XhsiACDATA	PD	I	A4	VDDQ_MIPIHSI
Xm0ADDR_14	GPY4[6]/EBI_ADDR[14]/XhsiACFLAG	PD	I	A4	VDDQ_MIPIHSI
Xm0ADDR_15	GPY4[7]/EBI_ADDR[15]/XhsiCAREADY	PD	I	A4	VDDQ_MIPIHSI
Xm0DATA_0	GPY5[0]/EBI_DATA[0]	PD	I	A3	VDDQ_M0
Xm0DATA_1	GPY5[1]/EBI_DATA[1]	PD	I	A3	VDDQ_M0
Xm0DATA_2	GPY5[2]/EBI_DATA[2]	PD	I	A3	VDDQ_M0
Xm0DATA_3	GPY5[3]/EBI_DATA[3]	PD	I	A3	VDDQ_M0
Xm0DATA_4	GPY5[4]/EBI_DATA[4]	PD	I	A3	VDDQ_M0
Xm0DATA_5	GPY5[5]/EBI_DATA[5]	PD	I	A3	VDDQ_M0
Xm0DATA_6	GPY5[6]/EBI_DATA[6]	PD	I	A3	VDDQ_M0
Xm0DATA_7	GPY5[7]/EBI_DATA[7]	PD	I	A3	VDDQ_M0
Xm0DATA_8	GPY6[0]/EBI_DATA[8]	PD	I	A3	VDDQ_M0
Xm0DATA_9	GPY6[1]/EBI_DATA[9]	PD	I	A3	VDDQ_M0
Xm0DATA_10	GPY6[2]/EBI_DATA[10]	PD	I	A3	VDDQ_M0
Xm0DATA_11	GPY6[3]/EBI_DATA[11]	PD	I	A3	VDDQ_M0
Xm0DATA_12	GPY6[4]/EBI_DATA[12]	PD	I	A3	VDDQ_M0
Xm0DATA_13	GPY6[5]/EBI_DATA[13]	PD	I	A3	VDDQ_M0
Xm0DATA_14	GPY6[6]/EBI_DATA[14]	PD	I	A3	VDDQ_M0
Xm0DATA_15	GPY6[7]/EBI_DATA[15]	PD	I	A3	VDDQ_M0
Xm1ADDR_0	MP1_0[0]/Xm1ADDR[0]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_1	MP1_0[1]/Xm1ADDR[1]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_2	MP1_0[2]/Xm1ADDR[2]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_3	MP1_0[3]/Xm1ADDR[3]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_4	MP1_0[4]/Xm1ADDR[4]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_5	MP1_0[5]/Xm1ADDR[5]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_6	MP1_0[6]/Xm1ADDR[6]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_7	MP1_0[7]/Xm1ADDR[7]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_8	MP1_1[0]/Xm1ADDR[8]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_9	MP1_1[1]/Xm1ADDR[9]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_10	MP1_1[2]/Xm1ADDR[10]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_11	MP1_1[3]/Xm1ADDR[11]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_12	MP1_1[4]/Xm1ADDR[12]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_13	MP1_1[5]/Xm1ADDR[13]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_14	MP1_1[6]/Xm1ADDR[14]	-	O(L)	B3	VDDQ_M1
Xm1ADDR_15	MP1_1[7]/Xm1ADDR[15]	-	O(L)	B3	VDDQ_M1
Xm1DATA_0	MP1_2[0]/Xm1DATA[0]	-	I	B3	VDDQ_M1

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
Xm1DATA_1	MP1_2[1]/Xm1DATA[1]	-	I	B3	VDDQ_M1
Xm1DATA_2	MP1_2[2]/Xm1DATA[2]	-	I	B3	VDDQ_M1
Xm1DATA_3	MP1_2[3]/Xm1DATA[3]	-	I	B3	VDDQ_M1
Xm1DATA_4	MP1_2[4]/Xm1DATA[4]	-	I	B3	VDDQ_M1
Xm1DATA_5	MP1_2[5]/Xm1DATA[5]	-	I	B3	VDDQ_M1
Xm1DATA_6	MP1_2[6]/Xm1DATA[6]	-	I	B3	VDDQ_M1
Xm1DATA_7	MP1_2[7]/Xm1DATA[7]	-	I	B3	VDDQ_M1
Xm1DATA_8	MP1_3[0]/Xm1DATA[8]	-	I	B3	VDDQ_M1
Xm1DATA_9	MP1_3[1]/Xm1DATA[9]	-	I	B3	VDDQ_M1
Xm1DATA_10	MP1_3[2]/Xm1DATA[10]	-	I	B3	VDDQ_M1
Xm1DATA_11	MP1_3[3]/Xm1DATA[11]	-	I	B3	VDDQ_M1
Xm1DATA_12	MP1_3[4]/Xm1DATA[12]	-	I	B3	VDDQ_M1
Xm1DATA_13	MP1_3[5]/Xm1DATA[13]	-	I	B3	VDDQ_M1
Xm1DATA_14	MP1_3[6]/Xm1DATA[14]	-	I	B3	VDDQ_M1
Xm1DATA_15	MP1_3[7]/Xm1DATA[15]	-	I	B3	VDDQ_M1
Xm1DATA_16	MP1_4[0]/Xm1DATA[16]	-	I	B3	VDDQ_M1
Xm1DATA_17	MP1_4[1]/Xm1DATA[17]	-	I	B3	VDDQ_M1
Xm1DATA_18	MP1_4[2]/Xm1DATA[18]	-	I	B3	VDDQ_M1
Xm1DATA_19	MP1_4[3]/Xm1DATA[19]	-	I	B3	VDDQ_M1
Xm1DATA_20	MP1_4[4]/Xm1DATA[20]	-	I	B3	VDDQ_M1
Xm1DATA_21	MP1_4[5]/Xm1DATA[21]	-	I	B3	VDDQ_M1
Xm1DATA_22	MP1_4[6]/Xm1DATA[22]	-	I	B3	VDDQ_M1
Xm1DATA_23	MP1_4[7]/Xm1DATA[23]	-	I	B3	VDDQ_M1
Xm1DATA_24	MP1_5[0]/Xm1DATA[24]	-	I	B3	VDDQ_M1
Xm1DATA_25	MP1_5[1]/Xm1DATA[25]	-	I	B3	VDDQ_M1
Xm1DATA_26	MP1_5[2]/Xm1DATA[26]	-	I	B3	VDDQ_M1
Xm1DATA_27	MP1_5[3]/Xm1DATA[27]	-	I	B3	VDDQ_M1
Xm1DATA_28	MP1_5[4]/Xm1DATA[28]	-	I	B3	VDDQ_M1
Xm1DATA_29	MP1_5[5]/Xm1DATA[29]	-	I	B3	VDDQ_M1
Xm1DATA_30	MP1_5[6]/Xm1DATA[30]	-	I	B3	VDDQ_M1
Xm1DATA_31	MP1_5[7]/Xm1DATA[31]	-	I	B3	VDDQ_M1
Xm1DQS_0	MP1_6[0]/Xm1DQS[0]	-	I	B3	VDDQ_M1
Xm1DQS_1	MP1_6[1]/Xm1DQS[1]	-	I	B3	VDDQ_M1
Xm1DQS_2	MP1_6[2]/Xm1DQS[2]	-	I	B3	VDDQ_M1
Xm1DQS_3	MP1_6[3]/Xm1DQS[3]	-	I	B3	VDDQ_M1
Xm1DQSn_0	MP1_6[4]/Xm1DQSn[0]	-	I	B3	VDDQ_M1

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
Xm1DQSn_1	MP1_6[5]/Xm1DQSn[1]	-	I	B3	VDDQ_M1
Xm1DQSn_2	MP1_6[6]/Xm1DQSn[2]	-	I	B3	VDDQ_M1
Xm1DQSn_3	MP1_6[7]/Xm1DQSn[3]	-	I	B3	VDDQ_M1
Xm1DQM_0	MP1_7[0]/Xm1DQM[0]	-	O(L)	B3	VDDQ_M1
Xm1DQM_1	MP1_7[1]/Xm1DQM[1]	-	O(L)	B3	VDDQ_M1
Xm1DQM_2	MP1_7[2]/Xm1DQM[2]	-	O(L)	B3	VDDQ_M1
Xm1DQM_3	MP1_7[3]/Xm1DQM[3]	-	O(L)	B3	VDDQ_M1
Xm1CKE_0	MP1_7[4]/Xm1CKE[0]	-	O(L)	A8	VDDQ_CKEM1
Xm1CKE_1	MP1_7[5]/Xm1CKE[1]	-	O(L)	A8	VDDQ_CKEM1
Xm1CLK	MP1_7[6]/Xm1CLK	-	O(L)	B3	VDDQ_M1
Xm1CLKn	MP1_7[7]/Xm1CLKn	-	O(L)	B3	VDDQ_M1
Xm1CSn_0	MP1_8[0]/Xm1CSn[0]	-	O(H)	B3	VDDQ_M1
Xm1CSn_1	MP1_8[1]/Xm1CSn[1]	-	O(H)	B3	VDDQ_M1
Xm1RASn	MP1_8[2]/Xm1RASn	-	O(H)	B3	VDDQ_M1
Xm1CASn	MP1_8[3]/Xm1CASn	-	O(H)	B3	VDDQ_M1
Xm1WEn	MP1_8[4]/Xm1WEn	-	O(H)	B3	VDDQ_M1
Xm1GATEI	MP1_8[5]/Xm1GATEI	-	I	B3	VDDQ_M1
Xm1GATEO	MP1_8[6]/Xm1GATEO	-	O(L)	B3	VDDQ_M1
Xm1ZQ	MP1_9[0]/Xm1ZQ	-	IO	B3	VDDQ_M1
Xm1VREF0	MP1_9[1:2]/Xm1VREF0/ Xm1VREF1	-	I	B3	VDDQ_M1
Xm1VREF2	MP1_9[3:4]/Xm1VREF2/ Xm1VREF3	-	I	B3	VDDQ_M1
Xm1BA_0	MP1_9[5]/Xm1BA[0]	-	O(L)	B3	VDDQ_M1
Xm1BA_1	MP1_9[6]/Xm1BA[1]	-	O(L)	B3	VDDQ_M1
Xm1BA_2	MP1_9[7]/Xm1BA[2]	-	O(L)	B3	VDDQ_M1
Xm1ODT_0	MP1_10[0]/Xm1ODT[0]	-	O(L)	B3	VDDQ_M1
Xm1ODT_1	MP1_10[1]/Xm1ODT[1]		O(L)	B3	VDDQ_M1
Xm2ADDR_0	MP2_0[0]/Xm2ADDR[0]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_1	MP2_0[1]/Xm2ADDR[1]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_2	MP2_0[2]/Xm2ADDR[2]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_3	MP2_0[3]/Xm2ADDR[3]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_4	MP2_0[4]/Xm2ADDR[4]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_5	MP2_0[5]/Xm2ADDR[5]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_6	MP2_0[6]/Xm2ADDR[6]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_7	MP2_0[7]/Xm2ADDR[7]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_8	MP2_1[0]/Xm2ADDR[8]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_9	MP2_1[1]/Xm2ADDR[9]	-	O(L)	B3	VDDQ_M2

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
Xm2ADDR_10	MP2_1[2]/Xm2ADDR[10]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_11	MP2_1[3]/Xm2ADDR[11]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_12	MP2_1[4]/Xm2ADDR[12]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_13	MP2_1[5]/Xm2ADDR[13]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_14	MP2_1[6]/Xm2ADDR[14]	-	O(L)	B3	VDDQ_M2
Xm2ADDR_15	MP2_1[7]/Xm2ADDR[15]	-	O(L)	B3	VDDQ_M2
Xm2DATA_0	MP2_2[0]/Xm2DATA[0]	-	I	B3	VDDQ_M2
Xm2DATA_1	MP2_2[1]/Xm2DATA[1]	-	I	B3	VDDQ_M2
Xm2DATA_2	MP2_2[2]/Xm2DATA[2]	-	I	B3	VDDQ_M2
Xm2DATA_3	MP2_2[3]/Xm2DATA[3]	-	I	B3	VDDQ_M2
Xm2DATA_4	MP2_2[4]/Xm2DATA[4]	-	I	B3	VDDQ_M2
Xm2DATA_5	MP2_2[5]/Xm2DATA[5]	-	I	B3	VDDQ_M2
Xm2DATA_6	MP2_2[6]/Xm2DATA[6]	-	I	B3	VDDQ_M2
Xm2DATA_7	MP2_2[7]/Xm2DATA[7]	-	I	B3	VDDQ_M2
Xm2DATA_8	MP2_3[0]/Xm2DATA[8]	-	I	B3	VDDQ_M2
Xm2DATA_9	MP2_3[1]/Xm2DATA[9]	-	I	B3	VDDQ_M2
Xm2DATA_10	MP2_3[2]/Xm2DATA[10]	-	I	B3	VDDQ_M2
Xm2DATA_11	MP2_3[3]/Xm2DATA[11]	-	I	B3	VDDQ_M2
Xm2DATA_12	MP2_3[4]/Xm2DATA[12]	-	I	B3	VDDQ_M2
Xm2DATA_13	MP2_3[5]/Xm2DATA[13]	-	I	B3	VDDQ_M2
Xm2DATA_14	MP2_3[6]/Xm2DATA[14]	-	I	B3	VDDQ_M2
Xm2DATA_15	MP2_3[7]/Xm2DATA[15]	-	I	B3	VDDQ_M2
Xm2DATA_16	MP2_4[0]/Xm2DATA[16]	-	I	B3	VDDQ_M2
Xm2DATA_17	MP2_4[1]/Xm2DATA[17]	-	I	B3	VDDQ_M2
Xm2DATA_18	MP2_4[2]/Xm2DATA[18]	-	I	B3	VDDQ_M2
Xm2DATA_19	MP2_4[3]/Xm2DATA[19]	-	I	B3	VDDQ_M2
Xm2DATA_20	MP2_4[4]/Xm2DATA[20]	-	I	B3	VDDQ_M2
Xm2DATA_21	MP2_4[5]/Xm2DATA[21]	-	I	B3	VDDQ_M2
Xm2DATA_22	MP2_4[6]/Xm2DATA[22]	-	I	B3	VDDQ_M2
Xm2DATA_23	MP2_4[7]/Xm2DATA[23]	-	I	B3	VDDQ_M2
Xm2DATA_24	MP2_5[0]/Xm2DATA[24]	-	I	B3	VDDQ_M2
Xm2DATA_25	MP2_5[1]/Xm2DATA[25]	-	I	B3	VDDQ_M2
Xm2DATA_26	MP2_5[2]/Xm2DATA[26]	-	I	B3	VDDQ_M2
Xm2DATA_27	MP2_5[3]/Xm2DATA[27]	-	I	B3	VDDQ_M2
Xm2DATA_28	MP2_5[4]/Xm2DATA[28]	-	I	B3	VDDQ_M2
Xm2DATA_29	MP2_5[5]/Xm2DATA[29]	-	I	B3	VDDQ_M2

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
Xm2DATA_30	MP2_5[6]/Xm2DATA[30]	-	I	B3	VDDQ_M2
Xm2DATA_31	MP2_5[7]/Xm2DATA[31]	-	I	B3	VDDQ_M2
Xm2DQS_0	MP2_6[0]/Xm2DQS[0]	-	I	B3	VDDQ_M2
Xm2DQS_1	MP2_6[1]/Xm2DQS[1]	-	I	B3	VDDQ_M2
Xm2DQS_2	MP2_6[2]/Xm2DQS[2]	-	I	B3	VDDQ_M2
Xm2DQS_3	MP2_6[3]/Xm2DQS[3]	-	I	B3	VDDQ_M2
Xm2DQSn_0	MP2_6[4]/Xm2DQSn[0]	-	I	B3	VDDQ_M2
Xm2DQSn_1	MP2_6[5]/Xm2DQSn[1]	-	I	B3	VDDQ_M2
Xm2DQSn_2	MP2_6[6]/Xm2DQSn[2]	-	I	B3	VDDQ_M2
Xm2DQSn_3	MP2_6[7]/Xm2DQSn[3]	-	I	B3	VDDQ_M2
Xm2DQM_0	MP2_7[0]/Xm2DQM[0]	-	O(L)	B3	VDDQ_M2
Xm2DQM_1	MP2_7[1]/Xm2DQM[1]	-	O(L)	B3	VDDQ_M2
Xm2DQM_2	MP2_7[2]/Xm2DQM[2]	-	O(L)	B3	VDDQ_M2
Xm2DQM_3	MP2_7[3]/Xm2DQM[3]	-	O(L)	B3	VDDQ_M2
Xm2CKE_0	MP2_7[4]/Xm2CKE[0]	-	O(L)	A8	VDDQ_CKEM2
Xm2CKE_1	MP2_7[5]/Xm2CKE[1]	-	O(L)	A8	VDDQ_CKEM2
Xm2CLK	MP2_7[6]/Xm2CLK	-	O(L)	B3	VDDQ_M2
Xm2CLKn	MP2_7[7]/Xm2CLKn	-	O(L)	B3	VDDQ_M2
Xm2CSn_0	MP2_8[0]/Xm2CSn[0]	-	O(H)	B3	VDDQ_M2
Xm2CSn_1	MP2_8[1]/Xm2CSn[1]	-	O(H)	B3	VDDQ_M2
Xm2RASn	MP2_8[2]/Xm2RASn	-	O(H)	B3	VDDQ_M2
Xm2CASn	MP2_8[3]/Xm2CASn	-	O(H)	B3	VDDQ_M2
Xm2WEn	MP2_8[4]/Xm2WEn	-	O(H)	B3	VDDQ_M2
Xm2GATEI	MP2_8[5]/Xm2GATEI	-	I	B3	VDDQ_M2
Xm2GATEO	MP2_8[6]/Xm2GATEO	-	O(L)	B3	VDDQ_M2
Xm2ZQ	MP2_9[0]/Xm2ZQ	-	IO	B3	VDDQ_M2
Xm2VREF0	MP2_9[1:2]/Xm2VREF0/ Xm2VREF1	-	I	B3	VDDQ_M2
Xm2VREF2	MP2_9[3:4]/Xm2VREF2/ Xm2VREF3	-	I	B3	VDDQ_M2
Xm2BA_0	MP2_9[5]/Xm2BA[0]	-	O(L)	B3	VDDQ_M2
Xm2BA_1	MP2_9[6]/Xm2BA[1]	-	O(L)	B3	VDDQ_M2
Xm2BA_2	MP2_9[7]/Xm2BA[2]	-	O(L)	B3	VDDQ_M2
Xm2ODT_0	MP2_10[0]/Xm2ODT[0]	-	O(L)	B3	VDDQ_M2
Xm2ODT_1	MP2_10[1]/Xm2ODT[1]		O(L)	B3	VDDQ_M2
XjTRSTn	ETC0[0]/XjTRSTn	–	I	A1	VDDQ_ISP
XjTMS	ETC0[1]/XjTMS	–	I(H)	A1	VDDQ_ISP
XjTCK	ETC0[2]/XjTCK	–	I(H)	A1	VDDQ_ISP

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
XjTDI	ETC0[3]/XjTDI	–	I(H)	A1	VDDQ_ISP
XjTDO	ETC0[4]/XjTDO	–	O(Z)	A1	VDDQ_ISP
XjDBGSEL	ETC0[5]/XjDBGSEL	PD	I	A1	VDDQ_ISP
XnRESET	ETC6[0]/XnRESET	–	I	B2	VDDQ_ISP
XCLKOUT	ETC6[1]/CLKOUT	–	O(L)	B2	VDDQ_ISP
XnRSTOUT	ETC6[2]/XnRSTOUT	–	O(H)	B2	VDDQ_ISP
XnWRESET	ETC6[3]/XnWRESET	–	I(H)	B2	VDDQ_ISP
XRTCCLKO	ETC6[4]/RTC_CLKOUT	–	O(L)	B2	VDDQ_CKO
XuotgDRVVBUS	ETC6[5]/XuotgDRVVBUS	–	O(L)	A1	VDDQ_ISP
XuhostPWREN	ETC6[6]/XuhostPWREN	–	O(L)	A1	VDDQ_ISP
XuhostOVERCUR	ETC6[7]/XuhostOVERCUR	PU	I	A1	VDDQ_ISP
XispPCLK	GPM0[0]/CAM_BAY_PCLK/CAM_B _PCLK/TS_CLK/TraceClk	PD	I	A1	VDDQ_ISP
XispRGB_0	GPM0[1]/CAM_BAY_RGB[0]/CAM_B _DATA[0]/TS_SYNC/TraceData[0]	PD	I	A1	VDDQ_ISP
XispRGB_1	GPM0[2]/CAM_BAY_RGB[1]/CAM_B _DATA[1]/TS_VAL/TraceData[1]	PD	I	A1	VDDQ_ISP
XispRGB_2	GPM0[3]/CAM_BAY_RGB[2]/CAM_B _DATA[2]/TS_DATA/TraceData[2]	PD	I	A1	VDDQ_ISP
XispRGB_3	GPM0[4]/CAM_BAY_RGB[3]/CAM_B _DATA[3]/TS_ERROR/TraceData[3]	PD	I	A1	VDDQ_ISP
XispRGB_4	GPM0[5]/CAM_BAY_RGB[4]/CAM_B _DATA[4]/XhsiCAWAKE/TraceData[4]	PD	I	A1	VDDQ_ISP
XispRGB_5	GPM0[6]/CAM_BAY_RGB[5]/CAM_B _DATA[5]/XhsiCADATA/TraceData[5]	PD	I	A1	VDDQ_ISP
XispRGB_6	GPM0[7]/CAM_BAY_RGB[6]/CAM_B _DATA[6]/XhsiCAFLAG/TraceData[6]	PD	I	A1	VDDQ_ISP
XispRGB_7	GPM1[0]/CAM_BAY_RGB[7]/CAM_B _DATA[7]/XhsiACREADY/TraceData[7]	PD	I	A1	VDDQ_ISP
XispRGB_8	GPM1[1]/CAM_BAY_RGB[8]/CAM_B _FIELD/XhsiACWAKE/TraceCtl	PD	I	A1	VDDQ_ISP
XispRGB_9	GPM1[2]/CAM_BAY _RGB[9]/XhsiACDATA/TraceData[8]	PD	I	A1	VDDQ_ISP
XispRGB_10	GPM1[3]/CAM_BAY _RGB[10]/XhsiACFLAG/TraceData[9]	PD	I	A1	VDDQ_ISP
XispRGB_11	GPM1[4]/CAM_BAY _RGB[11]/XhsiCAREADY/TraceData[10]	PD	I	A1	VDDQ_ISP
XispRGB_12	GPM1[5]/CAM_BAY _RGB[12]/TraceData[11]	PD	I	A1	VDDQ_ISP
XispRGB_13	GPM1[6]/CAM_BAY	PD	I	A1	VDDQ_ISP

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
	_RGB[13]/TraceData[12]				
XispVSYNC	GPM2[0]/CAM_BAY_Vsync/CAM_B _VSYNC/TraceData[13]	PD	I	A1	VDDQ_ISP
XispHSYNC	GPM2[1]/CAM_BAY_Hsync/CAM_B _HREF/TraceData[14]	PD	I	A1	VDDQ_ISP
XispMCLK	GPM2[2]/CAM_BAY_MCLK/CAM_B _CLKOUT/TraceData[15]	PD	I	A1	VDDQ_ISP
XispGP0	GPM2[3]/CAM_GPIO[0]/MPWM1_OUT _ISP	PD	I	A1	VDDQ_ISP
XispGP1	GPM2[4]/CAM_GPIO[1]/MPWM2_OUT _ISP	PD	I	A1	VDDQ_ISP
XispGP2	GPM3[0]/CAM_GPIO[2]/MPWM3_OUT _ISP	PD	I	A1	VDDQ_ISP
XispGP3	GPM3[1]/CAM_GPIO[3]/MPWM4_OUT _ISP	PD	I	A1	VDDQ_ISP
XispGP4	GPM3[2]/CAM_GPIO[4]/MPWM5_OUT _ISP/CAM_SPI1_MISO	PD	I	A1	VDDQ_ISP
XispGP5	GPM3[3]/CAM_GPIO[5]/MPWM6_OUT _ISP/CAM_SPI1_MOSI	PD	I	A1	VDDQ_ISP
XispGP6	GPM3[4]/CAM_GPIO[6]/nRTS_UART _ISP	PD	I	A1	VDDQ_ISP
XispGP7	GPM3[5]/CAM_GPIO[7]/TXD_UART _ISP	PD	I	A1	VDDQ_ISP
XispGP8	GPM3[6]/CAM_GPIO[8]/nCTS_UART _ISP	PD	I	A1	VDDQ_ISP
XispGP9	GPM3[7]/CAM_GPIO[9]/RXD_UART _ISP	PD	I	A1	VDDQ_ISP
XispI2C0SCL	GPM4[0]/CAM_I2C0_SCL/CAM _GPIO[10]	PD	I	A1	VDDQ_ISP
XispI2C0SDA	GPM4[1]/CAM_I2C0_SDA/CAM _GPIO[11]	PD	I	A1	VDDQ_ISP
XispI2C1SCL	GPM4[2]/CAM_I2C1_SCL/CAM _GPIO[12]/CAM_SPI1_CLKVDDQ_ISPI	PD	I	A1	VDDQ_ISP
XispI2C1SDA	GPM4[3]/CAM_I2C1_SDA/CAM _GPIO[13]/CAM_SPI1_nSS	PD	I	A1	VDDQ_ISP
XispSPICLK	GPM4[4]/CAM_SPI_CLK/CAM _GPIO[14]	PD	I	A1	VDDQ_ISP
XispSPICSn	GPM4[5]/CAM_SPI_nSS/CAM _GPIO[15]	PD	I	A1	VDDQ_ISP
XispSPIMISO	GPM4[6]/CAM_SPI_MISO/CAM _GPIO[16]	PD	I	A1	VDDQ_ISP

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
XispSPIMOSI	GPM4[7]/CAM_SPI_MOSI/CAM _GPIO[17]	PD	I	A1	VDDQ_ISP
XGNSS_RTC_ OUT	XGNSS_RTC_OUT	–	O	B2	VDDQ_SYS00
XGNSS_MCLK	GNSS_MCLK	–	I	B2	VDDQ_SYS00
XGNSS_CLKREQ	GNSS_CLKREQ	–	I	B2	VDDQ_SYS00
XtsEXT_RES	XtsEXT_RES	–	IO	B2	VDD18_TS
XXTI	XXTI	–	I	B2	VDDQ_SYS00
XOM_0	XOM[0]	–	I	B2	VDDQ_SYS00
XOM_1	XOM[1]	–	I	B2	VDDQ_SYS00
XOM_2	XOM[2]	–	I	B2	VDDQ_SYS00
XOM_3	XOM[3]	–	I	B2	VDDQ_SYS00
XOM_4	XOM[4]	–	I	B2	VDDQ_SYS00
XOM_5	XOM[5]	–	I	B2	VDDQ_SYS00
XOM_6	XOM[6]	–	I	B2	VDDQ_SYS00
XPWRRGTON	PWRRGTON	–	IO	B2	VDDQ_SYS00
XPSHOLD	XPSHOLD	–	O	B2	VDDQ_SYS00
XrtcXTI	XrtcXTI	–	I	B2	VDDQ_RTC
XrtcXTO	XrtcXTO	–	O	B2	VDDQ_RTC
XhsicSTROBE_0	XhsicSTROBE[0]	–	IO	B2	USBHSIC0_1.2 V
XhsicDATA_0	XhsicDATA[0]	–	IO	B2	USBHSIC0_1.2 V
XhsicSTROBE_1	XhsicSTROBE[1]	–	IO	B2	USBHSIC1_1.2 V
XhsicDATA_1	XhsicDATA[1]	–	IO	B2	USBHSIC1_1.2 V
XuotgDP	XuotgDP	–	IO	B2	VDD33_UOTG
XuotgREXT	XuotgREXT	–	IO	B2	VDD33_UOTG
XuotgDM	XuotgDM	–	IO	B2	VDD33_UOTG
XuotgANTEST	XuotgANALOGTEST	–	IO	B2	VDD33_UOTG
XuotgID	XuotgID	–	I	B2	VDD33_UOTG
XuotgVBUS	XuotgVBUS	–	IO	B2	VDD33_UOTG
XusbXTI	XusbXTI	–	I	B2	VDDQ_SYS02
XusbXTO	XusbXTO	–	O	B2	VDDQ_SYS02
XefFSOURCE	efrom_fsource	–	I	B2	VDDQ_SYS02
Xepllfilter	XepllEFILTER	–	IO	B2	VDD10_EPLL
Xvpllfilter	XvpllEFILTER	–	IO	B2	VDD10_VPLL
XabbPBBG_0	XabbPBBG_0	–	IO	B2	VDD18_ABB0
Xi2s0SCLK	GPZ[0]/I2S_0_SCLK	PD	I	A7	VDDQ_AUD

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
Xi2s0CDCLK	GPZ[1]/I2S_0_CDCLK	PD	I	A7	VDDQ_AUD
Xi2s0LRCK	GPZ[2]/I2S_0_LRCK	PD	I	A7	VDDQ_AUD
Xi2s0SDI	GPZ[3]/I2S_0_SDI	PD	I	A7	VDDQ_AUD
Xi2s0SDO_0	GPZ[4]/I2S_0_SDO[0]	PD	I	A7	VDDQ_AUD
Xi2s0SDO_1	GPZ[5]/I2S_0_SDO[1]	PD	I	A7	VDDQ_AUD
Xi2s0SDO_2	GPZ[6]/I2S_0_SDO[2]	PD	I	A7	VDDQ_AUD
Xc2cRXD_0	GPV0[0]/C2C_RXD[0]	PD	I	A9	VDDQ_C2C
Xc2cRXD_1	GPV0[1]/C2C_RXD[1]	PD	I	A9	VDDQ_C2C
Xc2cRXD_2	GPV0[2]/C2C_RXD[2]	PD	I	A9	VDDQ_C2C
Xc2cRXD_3	GPV0[3]/C2C_RXD[3]	PD	I	A9	VDDQ_C2C
Xc2cRXD_4	GPV0[4]/C2C_RXD[4]	PD	I	A9	VDDQ_C2C
Xc2cRXD_5	GPV0[5]/C2C_RXD[5]	PD	I	A9	VDDQ_C2C
Xc2cRXD_6	GPV0[6]/C2C_RXD[6]	PD	I	A9	VDDQ_C2C
Xc2cRXD_7	GPV0[7]/C2C_RXD[7]	PD	I	A9	VDDQ_C2C
Xc2cRXD_8	GPV1[0]/C2C_RXD[8]	PD	I	A9	VDDQ_C2C
Xc2cRXD_9	GPV1[1]/C2C_RXD[9]	PD	I	A9	VDDQ_C2C
Xc2cRXD_10	GPV1[2]/C2C_RXD[10]	PD	I	A9	VDDQ_C2C
Xc2cRXD_11	GPV1[3]/C2C_RXD[11]	PD	I	A9	VDDQ_C2C
Xc2cRXD_12	GPV1[4]/C2C_RXD[12]	PD	I	A9	VDDQ_C2C
Xc2cRXD_13	GPV1[5]/C2C_RXD[13]	PD	I	A9	VDDQ_C2C
Xc2cRXD_14	GPV1[6]/C2C_RXD[14]	PD	I	A9	VDDQ_C2C
Xc2cRXD_15	GPV1[7]/C2C_RXD[15]	PD	I	A9	VDDQ_C2C
Xc2cRXCLK_0	ETC7[0]/C2C_RXCLK[0]	PD	I	A9	VDDQ_C2C
Xc2cRXCLK_1	ETC7[1]/C2C_RXCLK[1]	PD	I	A9	VDDQ_C2C
Xc2cTXD_0	GPV2[0]/C2C_TXD[0]	PD	I	A9	VDDQ_C2C
Xc2cTXD_1	GPV2[1]/C2C_TXD[1]	PD	I	A9	VDDQ_C2C
Xc2cTXD_2	GPV2[2]/C2C_TXD[2]	PD	I	A9	VDDQ_C2C
Xc2cTXD_3	GPV2[3]/C2C_TXD[3]	PD	I	A9	VDDQ_C2C
Xc2cTXD_4	GPV2[4]/C2C_TXD[4]	PD	I	A9	VDDQ_C2C
Xc2cTXD_5	GPV2[5]/C2C_TXD[5]	PD	I	A9	VDDQ_C2C
Xc2cTXD_6	GPV2[6]/C2C_TXD[6]	PD	I	A9	VDDQ_C2C
Xc2cTXD_7	GPV2[7]/C2C_TXD[7]	PD	I	A9	VDDQ_C2C
Xc2cTXD_8	GPV3[0]/C2C_TXD[8]	PD	I	A9	VDDQ_C2C
Xc2cTXD_9	GPV3[1]/C2C_TXD[9]	PD	I	A9	VDDQ_C2C
Xc2cTXD_10	GPV3[2]/C2C_TXD[10]	PD	I	A9	VDDQ_C2C
Xc2cTXD_11	GPV3[3]/C2C_TXD[11]	PD	I	A9	VDDQ_C2C

Pin Name	Function	@Reset		Sleep State	Power Domain
		PUD	I/O		
Xc2cTXD_12	GPV3[4]/C2C_TXD[12]	PD	I	A9	VDDQ_C2C
Xc2cTXD_13	GPV3[5]/C2C_TXD[13]	PD	I	A9	VDDQ_C2C
Xc2cTXD_14	GPV3[6]/C2C_TXD[14]	PD	I	A9	VDDQ_C2C
Xc2cTXD_15	GPV3[7]/C2C_TXD[15]	PD	I	A9	VDDQ_C2C
Xc2cTXCLK_0	ETC8[0]/C2C_TXCLK[0]	PD	I	A9	VDDQ_C2C
Xc2cTXCLK_1	ETC8[1]/C2C_TXCLK[1]	PD	I	A9	VDDQ_C2C
Xc2cWKREQIN	GPV4[0]/C2C_WKREQIN	PD	I	A9	VDDQ_C2C_W
Xc2cWKREQOUT	GPV4[1]/C2C_WKREQOUT	PD	I	A9	VDDQ_C2C_W

2.5.2 Digital IO Power Pin

Table 2-3 describes the Exynos 4412 SCP digital I/O power pin list.

Table 2-3 Exynos 4412 SCP Digital I/O Power Pin List

Power Ball Name	GND Ball Name	Description	@SLEEP
VDDQ_M2	VSS	DDR Port2 IO VDD	OFF
VDDQ_CKEM2	VSS	DDR Port2 Clock Enable VDD	ON
VDDQ_M1	VSS	DDR Port1 IO VDD	OFF
VDDQ_CKEM1	VSS	DDR Port1 Clock Enable VDD	ON
VDDQ_C2C	VSS	C2C IO VDD	ON
VDDQ_C2C_W	VSS	C2C Wake up IO VDD	ON
VDDQ_SBUS	VSS	SLIM BUS IO VDD	ON
VDDQ_MIPIHSI	VSS	MIPI HSI IO VDD	ON
VDDQ_M0	VSS	EBI IO VDD	ON
VDDQ_SYS00	VSS	EINT[31:8], Reset IO VDD	ON
VDDQ_SYS02	VSS	USBXTI IO VDD	ON
VDDQ_ISP	VSS	ISP Interface IO VDD	ON
VDDQ_EXT	VSS	Peripheral IO VDD	ON
VDDQ_GPS	VSS	GPS IO VDD	ON
VDDQ_AUD	VSS	AUDIO IO VDD	ON
VDDQ_CAM	VSS	CAM IO VDD	ON
VDDQ_LCD	VSS	LCD IO VDD	ON
VDDQ_PRE 1)	VSS	IO Predriver VDD	ON/OFF
VDDQ_SYS33	VSS	EINT[7:0] IO VDD	ON
VDDQ_CKO	VSS	RTC Clock Out VDD	ON
VDDQ_MMC01 1)	VSS	MMC0/1 IO VDD	ON/OFF
VDDQ_MMC2 1)	VSS	MMC2 IO VDD	ON/OFF
VDDQ_MMC3	VSS	MMC3 IO VDD	ON
VDD_RTC	VSS	RTC Block and IO VDD	ON/OFF
VDD_INT	VSS	Internal block VDD (except ARM/DMC/G3D)	OFF
VDD_ALIVE	VSS	Alive block VDD	ON
VDD_ARM	VSS	ARM Core VDD	OFF
VDD_G3D	VSS	G3D VDD	OFF
VDD_MIF 2)	VSS	DMC block VDD	ON/OFF

NOTE:

1) VDDQ_PRE, VDDQ_MMC01, VDDQ_MMC2 could be OFF in sleep state to reduce SLEEP IO current. You should power off these signals together, if you want to reduce IO power in SLEEP mode.

2) VDD_MIF should be ON if C2C interface is used.

2.5.3 Analog/HSI Power Pin

Table 2-4 describes the Exynos 4412 SCP analog/HSI power pin list.

Table 2-4 Exynos 4412 SCP Analog/HSI Power Pin List

Power Ball Name	GND Ball Name	Description	@SLEEP
VDD10_HDMI	VSS_HDMI	HDMI TX VDD	OFF
VDD10_HDMI_PLL	VSS_HDMI_OSC	HDMI PLL VDD	OFF
VDD18_HDMI_OSC	VSS_HDMI	HDMI OSC VDD	OFF
VDD18_MIPI	VSS_MIPI	MIPI 4L 1.8 V VDD	OFF
VDD10_MIPI	VSS_MIPI	MIPI 4L 1.0 V VDD	OFF
VDD10_MIPI_PLL	VSS_MIPI	MIPI 4L PLL VDD	OFF
VDD18_MIPI2L	VSS_MIPI2L	MIPI 2L 1.8 V VDD	OFF
VDD10_MIPI2L	VSS_MIPI2L	MIPI 2L 1.0 V VDD	OFF
VDD10_APLL	VSS_APLL	APLL VDD	OFF
VDD10_MPLL 3)	VSS_MPLL	MPLL VDD	ON/OFF
VDD10_EPLL	VSS_EPLL	EPLL VDD	OFF
VDD10_VPLL	VSS_VPLL	VPLL VDD	OFF
VDD33_UOTG	VSSA_UOTG	USB OTG 3.3 V VDD	OFF
VDD10_UOTG	VSSA_UOTG	USB OTG 1.0 V VDD	OFF
VDD18_HSIC	VSS12_HSIC	USB HSIC 1.8 V VDD	OFF
VDD12_HSIC0 1)	VSS12_HSIC	USB HSIC 1.2 V VDD HSIC 0	N/A
VDD12_HSIC1 1)	VSS12_HSIC	USB HSIC 1.2 V VDD HSIC 1	N/A
VDD10_HSIC	VSS	USB HSIC 1.0 V VDD	OFF
VDD18_ADC 2)	VSS_ADC	ADC 1.8 V AVDD/ ADC 1.8 V DVDD	ON/OFF
VDD18_TS	VSS	TS 1.8 V AVDD	OFF
VDD18_ABB0	VSS	ABB AVDD 0 (INT Block)	OFF
VDD18_ABB1 3)	VSS	ABB AVDD 1 (MIF Block)	ON/OFF
VDD18_ABB2	VSS	ABB AVDD 2, 3 (G3D, ARM Block)	OFF

NOTE:

1) There is no power connection. It only monitors and caps pin..

2) VDD18_ADC should be ON in SLEEP mode if XADCAIN_x is not GND level.

3) VDD10_MPLL and VDD18_ABB1 should be ON if C2C interface is used.

2.5.4 Memory Power Pin

Table 2-5 describes the Exynos 4412 SCP memory power pin list.

Table 2-5 Exynos 4412 SCP Memory Power Pin List

Power Ball Name	GND Ball Name	Description	@SLEEP
VDDQ_E1	VSS	LPDDR2 IO Power	ON
VDDCA_E1	VSS	LPDDR2 Input Receiver Power	ON
VDDQ_E2	VSS	LPDDR2 IO Power	ON
VDDCA_E2	VSS	LPDDR2 Input Receiver Power	ON
VDD1_E	VSS	LPDDR2 Core1 Power	ON
VDD2_E	VSS	LPDDR2 Core2 Power	ON

3 Memory Map

3.1 Overview

This section describes the base address of region.

Base Address	Limit Address	Size	Description
0x0000_0000	0x0001_0000	64 KB	iROM
0x0200_0000	0x0201_0000	64 KB	iROM (mirror of 0x0 to 0x10000)
0x0202_0000	0x0206_0000	256 KB	iRAM
0x0300_0000	0x0302_0000	128 KB	Data memory or general purpose of Samsung Reconfigurable Processor SRP.
0x0302_0000	0x0303_0000	64 KB	I-cache or general purpose of SRP.
0x0303_0000	0x0303_9000	36 KB	Configuration memory (write only) of SRP
0x0381_0000	0x0383_0000	–	AudioSS's SFR region
0x0400_0000	0x0500_0000	16 MB	Bank0 of Static Read Only Memory Controller (SMC) (16-bit only)
0x0500_0000	0x0600_0000	16 MB	Bank1 of SMC
0x0600_0000	0x0700_0000	16 MB	Bank2 of SMC
0x0700_0000	0x0800_0000	16 MB	Bank3 of SMC
0x0800_0000	0x0C00_0000	64 MB	Reserved
0x0C00_0000	0x0CD0_0000	–	Reserved
0x0CE0_0000	0x0D00_0000	–	SFR region of Nand Flash Controller (NFCON)
0x1000_0000	0x1400_0000	–	SFR region
0x4000_0000	0xA000_0000	1.5 GB	Memory of Dynamic Memory Controller (DMC)-0
0xA000_0000	0x0000_0000	1.5 GB	Memory of DMC-1

3.2 SFR Base Address

This section describes the base address of SFR.

Base Address	IP
0x1000_0000	CHIPID
0x1001_0000	SYSREG
0x1002_0000	Power Management Unit (PMU)
0x1003_0000	CMU_TOP_PART
0x1004_0000	CMU_CORE_ISP_PART
0x1005_0000	Multi Core Timer (MCT)
0x1006_0000	Watch Dog Timer (WDT)
0x1007_0000	Real Time Clock (RTC)
0x100A_0000	KEYIF
0x100B_0000	HDMI_CEC
0x100C_0000	Thermal Management Unit (TMU)
0x1010_0000	SECKEY
0x1011_0000	TZPC0
0x1012_0000	TZPC1
0x1013_0000	TZPC2
0x1014_0000	TZPC3
0x1015_0000	TZPC4
0x1016_0000	TZPC5
0x1044_0000	Int_combiner
0x1048_0000	GIC_controller
0x1049_0000	GIC_distributor
0x1054_0000	AP_C2C
0x1058_0000	CP_C2C (Modem side)
0x1060_0000	DMC0
0x1061_0000	DMC1
0x106A_0000	PPMU_DMC_L
0x106B_0000	PPMU_DMC_R
0x106C_0000	PPMU_CPU
0x106E_0000	GPIO_C2C
0x1070_0000	TZASC_LR
0x1071_0000	TZASC_LW
0x1072_0000	TZASC_RR
0x1073_0000	TZASC_RW
0x1080_0000	G2D_ACP

Base Address	IP
0x1083_0000	Security Sub System (SSS)
0x1088_0000	Coresight
0x1089_0000	Coresight
0x108B_0000	Coresight
0x10A4_0000	SMMUG2D_ACP
0x10A5_0000	SMMUSSS
0x1100_0000	GPIO_right
0x1140_0000	GPIO_left
0x1180_0000	FIMC0
0x1181_0000	FIMC1
0x1182_0000	FIMC2
0x1183_0000	FIMC3
0x1184_0000	JPEG
0x1188_0000	MIPI_CSI0
0x1189_0000	MIPI_CSI1
0x11A2_0000	SMMUFIMC0
0x11A3_0000	SMMUFIMC1
0x11A4_0000	SMMUFIMC2
0x11A5_0000	SMMUFIMC3
0x11A6_0000	SMMUJPEG
0x11C0_0000	FIMD0
0x11C8_0000	MIPI_DSI0
0x11E2_0000	SMMUFIMD0
0x1200_0000	FIMC_ISP
0x1201_0000	FIMC_DRC_TOP
0x1204_0000	FIMC_FD_TOP
0x1211_0000	MPWM_ISP
0x1213_0000	I2C0_ISP
0x1214_0000	I2C1_ISP
0x1215_0000	MTCADC_ISP
0x1216_0000	PWM_ISP
0x1217_0000	WDT_ISP
0x1218_0000	MCUCTL_ISP
0x1219_0000	UART_ISP
0x121A_0000	SPI0_ISP
0x121B_0000	SPI1_ISP
0x121E_0000	GIC_C_ISP

Base Address	IP
0x121F_0000	GIC_D_ISP
0x1226_0000	sysMMU_FIMC-ISP
0x1227_0000	sysMMU_FIMC-DRC
0x122A_0000	sysMMU_FIMC-FD
0x122B_0000	sysMMU_ISPCPU
0x1239_0000	FIMC_LITE0
0x123A_0000	FIMC_LITE1
0x123B_0000	sysMMU_FIMC-LITE0
0x123C_0000	sysMMU_FIMC-LITE1
0x1248_0000	USBDEV0
0x1249_0000	USBDEV0
0x124A_0000	USBDEV0
0x124B_0000	USBDEV0
0x1250_0000	Transport Stream Interface (TSI)
0x1251_0000	SDMMC0
0x1252_0000	SDMMC1
0x1253_0000	SDMMC2
0x1254_0000	SDMMC3
0x1255_0000	SDMMC4
0x1256_0000	MIPI_HSI
0x1257_0000	SROMC
0x1258_0000	USBHOST0
0x1259_0000	USBHOST1
0x125B_0000	USBOTG1
0x1268_0000	PDMA0
0x1269_0000	PDMA1
0x126C_0000	General ADC
0x1281_0000	Rotator
0x1284_0000	sMDMA
0x1285_0000	nsMDMA
0x12A3_0000	SMMURotator
0x12A4_0000	SMMUMDMA
0x12C0_0000	Video Processor (VP)
0x12C1_0000	Mixer
0x12D0_0000	HDMI0
0x12D1_0000	HDMI1
0x12D2_0000	HDMI2

Base Address	IP
0x12D3_0000	HDMI3
0x12D4_0000	HDMI4
0x12D5_0000	HDMI5
0x12D6_0000	HDMI6
0x12E2_0000	SMMUTV
0x1300_0000	3D Graphic Accelerator (G3D)
0x1322_0000	PPMU_3D
0x1340_0000	Multi Format Codec (MFC)
0x1362_0000	SMMUMFC_L
0x1363_0000	SMMUMFC_R
0x1366_0000	PPMU_MFC_L
0x1367_0000	PPMU_MFC_R
0x1380_0000	Universal Asynchronous Receiver And Transmitter0 (UART)
0x1381_0000	UART1
0x1382_0000	UART2
0x1383_0000	UART3
0x1384_0000	UART4
0x1386_0000	Inter-Integrated Circuit0 (I2C)
0x1387_0000	I2C1
0x1388_0000	I2C2
0x1389_0000	I2C3
0x138A_0000	I2C4
0x138B_0000	I2C5
0x138C_0000	I2C6
0x138D_0000	I2C7
0x138E_0000	I2CHDMI
0x1392_0000	Serial Peripheral Interface0 (SPI)
0x1393_0000	SPI1
0x1394_0000	SPI2
0x1396_0000	I2S1
0x1397_0000	I2S2
0x1398_0000	PCM1
0x1399_0000	PCM2
0x139A_0000	AC97
0x139B_0000	SPDIF
0x139D_0000	PWMTimer

4 Chip ID

4.1 Overview

The Exynos 4412 SCP includes a Chip ID block for the Software (SW) that sends and receives Advanced Peripheral Bus (APB) interface signals to the bus system.

4.2 Register Description

4.2.1 Register Map Summary

- Base Address: 0x1000_0000

Register	Offset	Description	Reset Value
PRO_ID	0x0000	Product information (ID, package, and revision)	0xE441_2XXX
PACKAGE_ID	0x0004	Package information (POP type and package)	0xXXXX_XXXX

4.2.1.1 PRO_ID

- Base Address: 0x1000_0000
- Address = Base Address + 0x0000, Reset Value = 0xE441_2XXX

Name	Bit	Type	Description	Reset Value
Product ID	[31:12]	R	Product ID	0x4412
RSVD	[11:10]	R	Reserved	0x0
Package	[9:8]	R	Package Information	Exynos 4412 SCP: SCP : 0x0
MainRev	[7:4]	R	Main Revision Number	0x0
SubRev	[3:0]	R	Sub Revision Number	0x0

NOTE: PRO_ID register[31:0] depends on the e-fuse ROM value. As power on sequence is progressing, it loads the e-fuse ROM values to the registers. It can read the loaded current e-fuse ROM values. An e-fuse ROM has main and sub revision numbers.

4.2.1.2 PACKAGE_ID

- Base Address: 0x1000_0000
- Address = Base Address + 0x0004, Reset Value = 0xXXXX_XXXX

Name	Bit	Type	Description	Reset Value
Package ID	[31:0]	R	Package information (POP type and package)	0xXXXX_XXXX

NOTE: PACKAGE_ID register[31:0] depends on the e-fuse ROM value.

5 Booting Sequence

5.1 Overview

Exynos 4412 SCP has 64 KB ROM (iROM) and 256 KB SRAM (iRAM) as internal memory.

You can select the booting device from the following list:

- General NAND flash memory
- SD/MMC memory card
- eMMC memory
- USB device

At the system reset, the program execution starts at iROM. The system reset may be asserted not only on booting time, but also on wakeup from low power modes. Therefore, the boot loader code executes appropriate processes according to the reset status. Refer to *Figure 5-1* for more information.

The boot loader is comprises the first and the second boot loaders. The characteristics of these boot loaders are:

- iROM: It is a small and simple code to initiate SOC. It is implemented on internal ROM of SOC.
- First boot loader (BL1): It is chip-specific and stored in external memory device.
- Second boot loader (BL2): It is platform-specific and stored in external memory device. User should build and store this in an external memory device. It is not provided by Samsung.

5.2 Functional Description

This section includes:

- Block Diagram
- Scenario Description

5.2.1 Block Diagram

Figure 5-1 illustrates the block diagram of booting time operation.

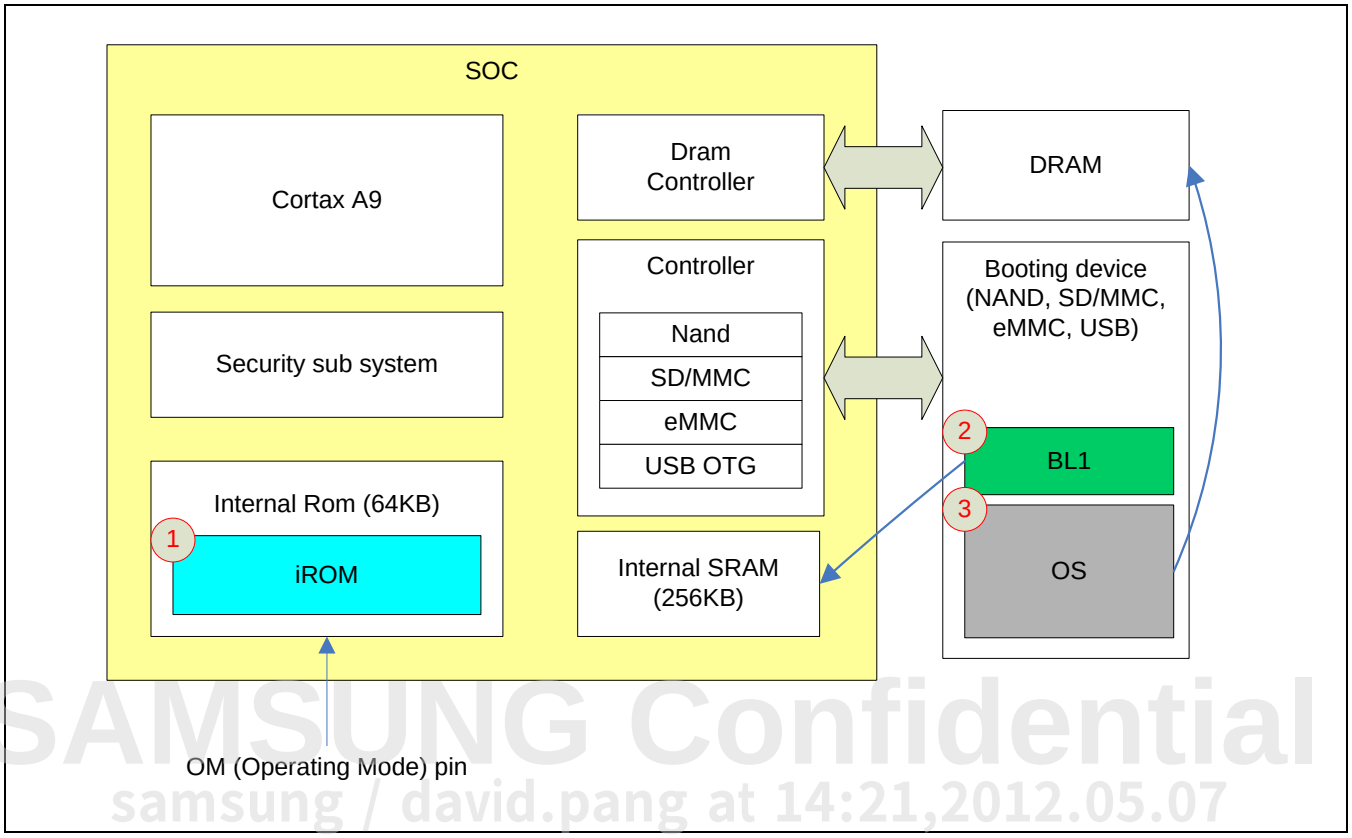


Figure 5-1 Block Diagram of Booting Time Operation

- iROM is placed in internal 64 KB ROM. It initializes basic system functions such as clock and stack.
- iROM loads BL1 image from a specific booting device to internal 256 KB SRAM. The booting device is selected by Operating Mode (OM) pins. According to the secure boot key values, iROM may do an integrity check on BL1 image.
- BL1 initializes system clock, and DRAM controller. After initializing DRAM controller, it loads OS image from the booting device to DRAM. According to the secure boot key values, BL1 can do an integrity check on the OS image.
- After the booting completes, BL1 jumps to the operating system.

iROM reads the OM pins to find the booting device. The OM register provides the OM pin and other information required for booting. Refer to Section 4, "Chip ID" for more information on OM register.
The OM pin decides the booting devices such as NAND, MoviNAND, and iNAND.
USB booting is provided for system debugging and flash reprogramming, not for normal booting.

5.2.2 Scenario Description

This section includes:

- Reset Status
- PLL and Clock Setting at Booting Time
- OM Pin Configuration

5.2.2.1 Reset Status

There are several scenarios for system reset such as hardware reset, watchdog reset, software reset, and wake up from power down modes.
Table 5-1 lists the mandatory functions required for various reset status.

Table 5-1 Functions Needed for Various Reset Status

	Initialization in iROM	PLL Setting in iROM	BL1 Loading	PLL and DRAM Setting in BL1	OS Loading	Restore Previous State
XnRESET	O	O	O	O	O	X
Watchdog reset	O	O	O	O	O	X
Wake up from SLEEP	O	O	O	O	X	O
SW reset	O	O	O	O	O	X
Wake up from DEEP-STOP	O	X (1)	X (2)	O	X	O
Wake up from LPA	O	X	X (2)	O	X	O

NOTE:

1. PLL registers are in retention state in DEEP_STOP. Even though there is no PLL setting in iROM, PLL register values are not in reset state.
2. When the contents of SRAM are preserved by retention option.

At the time of hardware reset and watchdog reset, the system should boot fully with BL1 and loading of OS image. The new reset status is classified as reset group 0.

Because the contents of DRAM memory are preserved in the SLEEP mode, it does not require loading the OS image to DRAM. However, it does not supply SoC internal power to internal logic during SLEEP mode and it does not preserve all contents in internal SRAM. Therefore, BL1 should be loaded again. This reset status is classified as reset group 1.

At the time of software reset, the contents of both internal SRAM and external DRAM are preserved. Therefore, it does not require loading of boot loader. Although power of top block power is gated in DEEP_STOP and LPA modes, the internal SRAM can be reserved, so that it does not require re-loading of boot loader. In case of non-retention of SRAM in DEEP_STOP and LPA modes, BL1 should be loaded again. These software reset that wake up from DEEP_STOP and LPA statuses are classified as reset group 2.

When system enters into all power down modes, you should save the current system status to safe memory region such as DRAM so that the system continues processing seamlessly after waking up from power down modes. Finally, it requires restoring previous state function on wake up from SLEEP, DEEP_STOP, and LPA modes.

5.2.2.2 PLL and Clock Setting at Booting Time

iROM initializes the PLL with fixed value. The PLL setting is as follows:

- APLL: M = 100, P = 3, S = 1 FOUT = (MDIV × FIN)/(PDIV × 2SDIV) = 400 MHz
- MPLL: M = 100, P = 3, S = 1 FOUT = (MDIV × FIN)/(PDIV × 2SDIV) = 400 MHz

Table 5-2 lists the system clock frequencies for various external crystals after initialization of the PLL by iROM.

Table 5-2 iROM's Clock Speed at 24 MHz ext. Crystal

ARMCLK	ACLK200	ACLK160	ACLK133	ACLK100	ACLK _ACP	ACLK _COREM0	ACLK _COREM1
400	24	80	66	50	100	100	50

ACLK _DMCD	ACLK _DMCP	ACLK _DMC	ACLK _GDL	ACLK _GDR	ACLK _GPL	ACLK _GPR	PCLK _ACP
100	50	200	100	100	50	50	50

5.2.2.3 OM Pin Configuration

Table 5-3 lists the booting options set by the OM pins.

Table 5-3 OM Pin Setting

OM[5:1]	1st Device	2nd Device
5b'00000	Reserved	
5b'00001	Reserved	
5b'00010	SDMMC_CH2	USB
5b'00011	eMMC43_CH0	USB
5b'00100	eMMC44_CH4	USB
5b'00101 to 5b'00111	Reserved	
5b'01000	NAND_512_8ECC	USB
5b'01001	NAND_2KB_OVER	USB
5b'01001 to 5b'01111	Reserved	
5b'10000	Reserved	
5b'10001	Reserved	
5b'10010	Reserved	
5b'10011	eMMC43_CH0	SDMMC_CH2
5b'10100	eMMC44_CH4	SDMMC_CH2
5b'10101 to 5b'10111	Reserved	
5b'11000	NAND_512_8ECC	SDMMC_CH2
5b'11001	NAND_2KB_OVER	SDMMC_CH2
5b'11001 to 5b'11111	Reserved	

NOTE:

1. You should tie OM[6] to the low level.
2. If it fails to download BL1 from the first booting device, the iROM code tries to download BL1 from the second booting device (USB or SDMMC_CH2).

6 General Purpose Input/Output (GPIO) Control

This chapter describes the General Purpose Input/Output (GPIO).

6.1 Overview

Exynos 4412 SCP contains 304 multi-functional input/output port pins and 164 memory port pins. There are 37 general port groups and two memory port groups. They are:

- GPA0, GPA1: 14 in/out ports-3xUART with flow control, UART without flow control, and/ or 2xI2C
- GPB: 8 in/out ports-2xSPI and/ or 2xI2C and/ or IEM
- GPC0, GPC1: 10 in/out ports-2xI2S, and/ or 2xPCM, and/ or AC97, SPDIF, I2C, and/ or SPI
- GPD0, GPD1: 8 in/out ports-PWM, 2xI2C, and/ or LCD I/F, MIPI
- GPM0, GPM1, GPM2, GPM3, GPM4: 35 in/out ports-CAM I/F, and/ or TS I/F, HSI, and/ or Trace I/F
- GPF0, GPF1, GPF2, GPF3: 30 in/out ports-LCD I/F
- GPJ0, GPJ1: 13 in/out ports-CAM I/F
- GPK0, GPK1, GPK2, GPK3: 28 in/out ports-4xMMC (4-bit MMC), and/ or 2xMMC (8-bit MMC)), and/ or GPS debugging I/F
- GPL0, GPL1: 11 in/out ports-GPS I/F
- GPL2: 8 in/out ports-GPS debugging I/F or Key pad I/F
- GPX0, GPX1, GPX2, GPX3: 32 in/out ports-External wake-up, and/ or Key pad I/F

NOTE: These are in ALIVE region.

- GPZ: 7 in/out ports-low Power I2S and/ or PCM
- GPY0, GPY1, GPY2: 16 in/out ports-Control signals of EBI (SROM, NF, One NAND)
- GPY3, GPY4, GPY5, GPY6: 32 in/out memory ports-EBI (For more information about EBI configuration, refer to Chapter 5, and 6)
- MP1_0-MP1_9: 78 DRAM1 ports

NOTE: GPIO registers does not control these ports.

- MP2_0-MP2_9: 78 DRAM2 ports

NOTE: GPIO registers does not control these ports.

- ETC0, ETC1, ETC6: 18 in/out ETC ports-JTAG, SLIMBUS, RESET, CLOCK
- ETC7, ETC8 : 4 clock port for C2C

Warning: When you do not use or connect port to an input pin without Pull-up/Pull-down then do not leave a port in Input Pull-up/Pull-down disable state. It may cause unexpected state and leakage current. Disable Pull-up/Pull-down when you use port as output function.

6.1.1 Features of GPIO

The features of GPIO include:

- Controls 172 External Interrupts
- Controls 32 External Wake-up Interrupts
- 252 multi-functional input/output ports
- Controls pin states in Sleep Mode except GPX0, GPX1, GPX2, and GPX3 (GPX* pins are alive-pads)

6.1.2 Input/Output Description

This section includes:

- General Purpose Input/Output Block Diagram
- Register Description

6.1.2.1 General Purpose Input/Output Block Diagram

GPIO consists of two parts,

- alive-part
- off-part

In Alive-part, you should supply power on sleep mode, but in off-part, it is not same. Therefore, registers in alive-part keep their values during sleep mode.

Figure 6-1 illustrates the block diagram of GPIO.

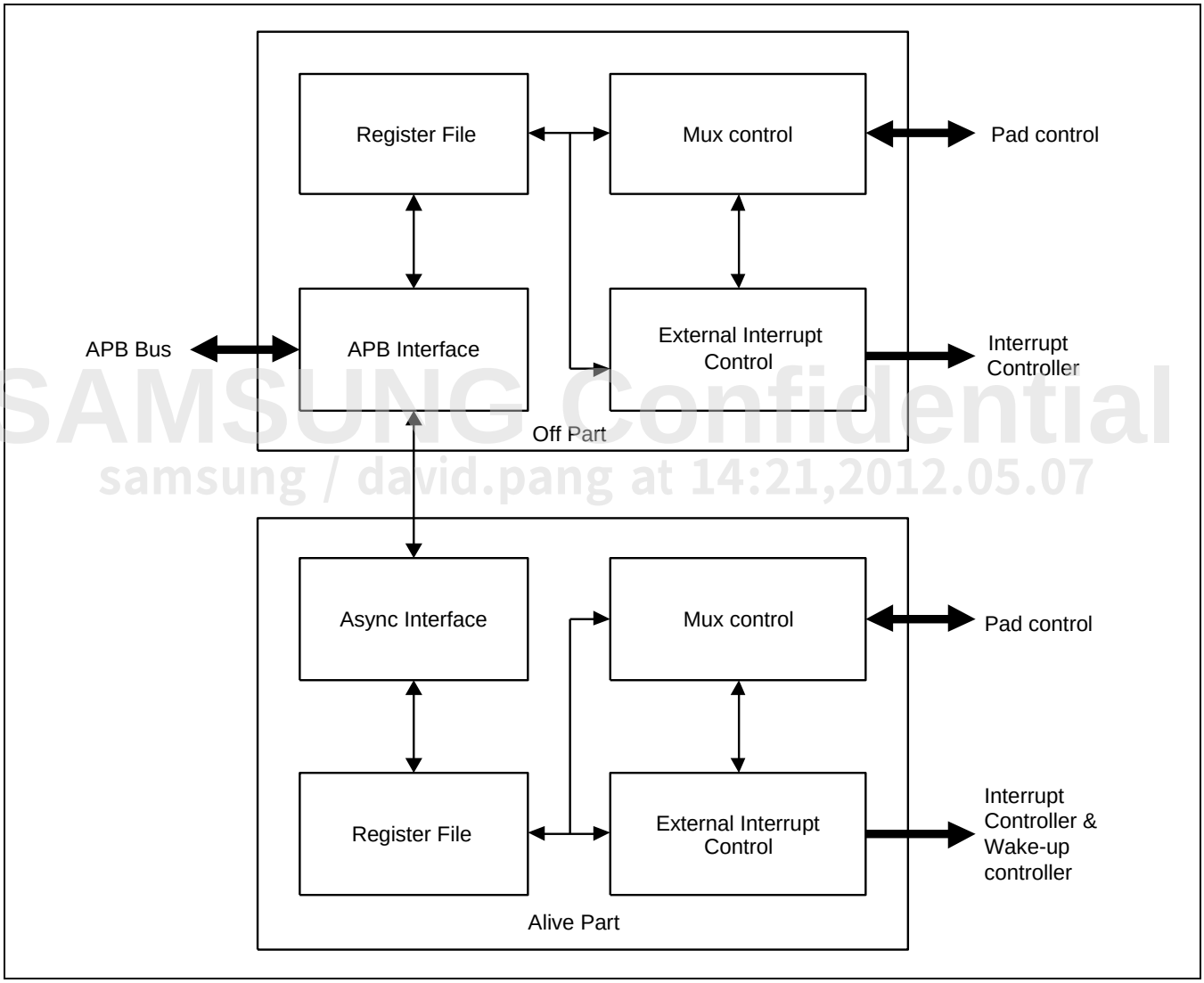


Figure 6-1 GPIO Block Diagram

6.2 Register Description

6.2.1 Registers Summary

- Base Address: 0x1140_0000

Register	Offset	Description	Reset Value
GPA0CON	0x0000	Port group GPA0 configuration register	0x0000_0000
GPA0DAT	0x0004	Port group GPA0 data register	0x00
GPA0PUD	0x0008	Port group GPA0 pull-up/ pull-down register	0x5555
GPA0DRV	0x000C	Port group GPA0 drive strength control register	0x00_0000
GPA0CONPDN	0x0010	Port group GPA0 power down mode configuration register	0x0000
GPA0PUDPDN	0x0014	Port group GPA0 power down mode pull-up/ pull-down register	0x0000
GPA1CON	0x0020	Port group GPA1 configuration register	0x0000_0000
GPA1DAT	0x0024	Port group GPA1 data register	0x00
GPA1PUD	0x0028	Port group GPA1 pull-up/ pull-down register	0x0555
GPA1DRV	0x002C	Port group GPA1 drive strength control register	0x00_0000
GPA1CONPDN	0x0030	Port group GPA1 power down mode configuration register	0x0000
GPA1PUDPDN	0x0034	Port group GPA1 power down mode pull-up/ pull-down register	0x0000
GPBCON	0x0040	Port group GPB configuration register	0x0000_0000
GPBDAT	0x0044	Port group GPB data register	0x00
GPBPUD	0x0048	Port group GPB pull-up/ pull-down register	0x5555
GPBDRV	0x004C	Port group GPB drive strength control register	0x00_0000
GPBCONPDN	0x0050	Port group GPB power down mode configuration register	0x0000
GPBPUDPDN	0x0054	Port group GPB power down mode pull-up/ pull-down register	0x0000
GPC0CON	0x0060	Port group GPC0 configuration register	0x0000_0000
GPC0DAT	0x0064	Port group GPC0 data register	0x00
GPC0PUD	0x0068	Port group GPC0 Pull-up/ pull-down register	0x0155
GPC0DRV	0x006C	Port group GPC0 drive strength control register	0x00_0000
GPC0CONPDN	0x0070	Port group GPC0 power down mode configuration register	0x0000
GPC0PUDPDN	0x0074	Port group GPC0 power down mode pull-up/ pull-down register	0x0000
GPC1CON	0x0080	Port group GPC1 configuration register	0x0000_0000
GPC1DAT	0x0084	Port group GPC1 data register	0x00
GPC1PUD	0x0088	Port group GPC1 pull-up/ pull-down register	0x0155
GPC1DRV	0x008C	Port group GPC1 drive strength control register	0x00_0000
GPC1CONPDN	0x0090	Port group GPC1 power down mode configuration register	0x0000
GPC1PUDPDN	0x0094	Port group GPC1 power down mode pull-up/ pull-down	0x0000

Register	Offset	Description	Reset Value
		register	
GPD0CON	0x00A0	Port group GPD0 configuration register	0x0000_0000
GPD0DAT	0x00A4	Port group GPD0 data register	0x00
GPD0PUD	0x00A8	Port group GPD0 pull-up/ pull-down register	0x0055
GPD0DRV	0x00AC	Port group GPD0 drive strength control register	0x00_0000
GPD0CONPDN	0x00B0	Port group GPD0 power down mode configuration register	0x0000
GPD0PUDPDN	0x00B4	Port group GPD0 power down mode pull-up/ pull-down register	0x0000
GPD1CON	0x00C0	Port group GPD1 configuration register	0x0000_0000
GPD1DAT	0x00C4	Port group GPD1 data register	0x00
GPD1PUD	0x00C8	Port group GPD1 Pull-up/ pull-down register	0x0055
GPD1DRV	0x00CC	Port group GPD1 drive strength control register	0x00_0000
GPD1CONPDN	0x00D0	Port group GPD1 power down mode configuration register	0x0000
GPD1PUDPDN	0x00D4	Port group GPD1 power down mode pull-up/ pull-down register	0x0000
GPF0CON	0x0180	Port group GPF0 configuration register	0x0000_0000
GPF0DAT	0x0184	Port group GPF0 data register	0x00
GPF0PUD	0x0188	Port group GPF0 pull-up/ pull-down register	0x5555
GPF0DRV	0x018C	Port group GPF0 drive strength control register	0x00_0000
GPF0CONPDN	0x0190	Port group GPF0 power down mode configuration register	0x0000
GPF0PUDPDN	0x0194	Port group GPF0 power down mode pull-up/ pull-down register	0x0000
GPF1CON	0x01A0	Port group GPF1 configuration register	0x0000_0000
GPF1DAT	0x01A4	Port group GPF1 data register	0x00
GPF1PUD	0x01A8	Port group GPF1 pull-up/ pull-down register	0x5555
GPF1DRV	0x01AC	Port group GPF1 drive strength control register	0x00_0000
GPF1CONPDN	0x01B0	Port group GPF1 power down mode configuration register	0x0000
GPF1PUDPDN	0x01B4	Port group GPF1 power down mode pull-up/ pull-down register	0x0000
GPF2CON	0x01C0	Port group GPF2 configuration register	0x0000_0000
GPF2DAT	0x01C4	Port group GPF2 data register	0x00
GPF2PUD	0x01C8	Port group GPF2 pull-up/ pull-down register	0x5555
GPF2DRV	0x01CC	Port group GPF2 drive strength control register	0x00_0000
GPF2CONPDN	0x01D0	Port group GPF2 power down mode configuration register	0x0000
GPF2PUDPDN	0x01D4	Port group GPF2 power down mode pull-up/ pull-down register	0x0000
GPF3CON	0x01E0	Port group GPF3 configuration register	0x0000_0000
GPF3DAT	0x01E4	Port group GPF3 data register	0x00

Register	Offset	Description	Reset Value
GPF3PUD	0x01E8	Port group GPF3 pull-up/ pull-down register	0x0555
GPF3DRV	0x01EC	Port group GPF3 drive strength control register	0x00_0000
GPF3CONPDN	0x01F0	Port group GPF3 power down mode configuration register	0x0000
GPF3PUDPDN	0x01F4	Port group GPF3 power down mode pull-up/ pull-down register	0x0000
ETC1PUD	0x0228	Port group ETC1 pull-up/ pull-down register	0x0005
ETC1DRV	0x022C	Port group ETC1 drive strength control register	0x00_0000
GPJ0CON	0x0240	Port group GPJ0 configuration register	0x0000_0000
GPJ0DAT	0x0244	Port group GPJ0 data register	0x00
GPJ0PUD	0x0248	Port group GPJ0 pull-up/ pull-down register	0x5555
GPJ0DRV	0x024C	Port group GPJ0 drive strength control register	0x00_0000
GPJ0CONPDN	0x0250	Port group GPJ0 power down mode configuration register	0x0000
GPJ0PUDPDN	0x0254	Port group GPJ0 power down mode pull-up/ pull-down register	0x0000
GPJ1CON	0x0260	Port group GPJ1 configuration register	0x0000_0000
GPJ1DAT	0x0264	Port group GPJ1 data register	0x00
GPJ1PUD	0x0268	Port group GPJ1 pull-up/ pull-down register	0x0155
GPJ1DRV	0x026C	Port group GPJ1 drive strength control register	0x00_0000
GPJ1CONPDN	0x0270	Port group GPJ1 power down mode configuration register	0x0000
GPJ1PUDPDN	0x0274	Port group GPJ1 power down mode pull-up/ pull-down register	0x0000
EXT_INT1_CON	0x0700	External interrupt EXT_INT1 configuration register	0x0000_0000
EXT_INT2_CON	0x0704	External interrupt EXT_INT2 configuration register	0x0000_0000
EXT_INT3_CON	0x0708	External interrupt EXT_INT3 configuration register	0x0000_0000
EXT_INT4_CON	0x070C	External interrupt EXT_INT4 configuration register	0x0000_0000
EXT_INT5_CON	0x0710	External interrupt EXT_INT5 configuration register	0x0000_0000
EXT_INT6_CON	0x0714	External interrupt EXT_INT6 configuration register	0x0000_0000
EXT_INT7_CON	0x0718	External interrupt EXT_INT7 configuration register	0x0000_0000
EXT_INT13_CON	0x0730	External interrupt EXT_INT13 configuration register	0x0000_0000
EXT_INT14_CON	0x0734	External interrupt EXT_INT14 configuration register	0x0000_0000
EXT_INT15_CON	0x0738	External interrupt EXT_INT15 configuration register	0x0000_0000
EXT_INT16_CON	0x073C	External interrupt EXT_INT16 configuration register	0x0000_0000
EXT_INT21_CON	0x0740	External interrupt EXT_INT21 configuration register	0x0000_0000
EXT_INT22_CON	0x0744	External interrupt EXT_INT22 configuration register	0x0000_0000
EXT_INT1_FLTCON0	0x0800	External interrupt EXT_INT1 filter configuration register 0	0x0000_0000
EXT_INT1_FLTCON1	0x0804	External interrupt EXT_INT1 filter configuration register 1	0x0000_0000
EXT_INT2_FLTCON0	0x0808	External interrupt EXT_INT2 filter configuration register 0	0x0000_0000

Register	Offset	Description	Reset Value
EXT_INT2_FLTCON1	0x080C	External interrupt EXT_INT2 filter configuration register 1	0x0000_0000
EXT_INT3_FLTCON0	0x0810	External interrupt EXT_INT3 filter configuration register 0	0x0000_0000
EXT_INT3_FLTCON1	0x0814	External interrupt EXT_INT3 filter configuration register 1	0x0000_0000
EXT_INT4_FLTCON0	0x0818	External interrupt EXT_INT4 filter configuration register 0	0x0000_0000
EXT_INT4_FLTCON1	0x081C	External interrupt EXT_INT4 filter configuration register 1	0x0000_0000
EXT_INT5_FLTCON0	0x0820	External interrupt EXT_INT5 filter configuration register 0	0x0000_0000
EXT_INT5_FLTCON1	0x0824	External interrupt EXT_INT5 filter configuration register 1	0x0000_0000
EXT_INT6_FLTCON0	0x0828	External interrupt EXT_INT6 filter configuration register 0	0x0000_0000
EXT_INT6_FLTCON1	0x082C	External interrupt EXT_INT6 filter configuration register 1	0x0000_0000
EXT_INT7_FLTCON0	0x0830	External interrupt EXT_INT7 filter configuration register 0	0x0000_0000
EXT_INT7_FLTCON1	0x0834	External interrupt EXT_INT7 filter configuration register 1	0x0000_0000
EXT_INT13_FLTCON0	0x0860	External interrupt EXT_INT13 filter configuration register 0	0x0000_0000
EXT_INT13_FLTCON1	0x0864	External interrupt EXT_INT13 filter configuration register 1	0x0000_0000
EXT_INT14_FLTCON0	0x0868	External interrupt EXT_INT14 filter configuration register 0	0x0000_0000
EXT_INT14_FLTCON1	0x086C	External interrupt EXT_INT14 filter configuration register 1	0x0000_0000
EXT_INT15_FLTCON0	0x0870	External interrupt EXT_INT15 filter configuration register 0	0x0000_0000
EXT_INT15_FLTCON1	0x0874	External interrupt EXT_INT15 filter configuration register 1	0x0000_0000
EXT_INT16_FLTCON0	0x0878	External interrupt EXT_INT16 filter configuration register 0	0x0000_0000
EXT_INT16_FLTCON1	0x087C	External interrupt EXT_INT16 filter configuration register 1	0x0000_0000
EXT_INT21_FLTCON0	0x0880	External interrupt EXT_INT21 filter configuration register 0	0x0000_0000
EXT_INT21_FLTCON1	0x0884	External interrupt EXT_INT21 filter configuration register 1	0x0000_0000
EXT_INT22_FLTCON0	0x0888	External interrupt EXT_INT22 filter configuration register 0	0x0000_0000
EXT_INT22_FLTCON1	0x088C	External interrupt EXT_INT22 filter configuration register 1	0x0000_0000
EXT_INT1_MASK	0x0900	External interrupt EXT_INT1 mask register	0x0000_00FF
EXT_INT2_MASK	0x0904	External interrupt EXT_INT2 mask register	0x0000_003F
EXT_INT3_MASK	0x0908	External interrupt EXT_INT3 mask register	0x0000_00FF
EXT_INT4_MASK	0x090C	External interrupt EXT_INT4 mask register	0x0000_001F
EXT_INT5_MASK	0x0910	External interrupt EXT_INT5 mask register	0x0000_001F
EXT_INT6_MASK	0x0914	External interrupt EXT_INT6 mask register	0x0000_000F
EXT_INT7_MASK	0x0918	External interrupt EXT_INT7 mask register	0x0000_000F
EXT_INT13_MASK	0x0930	External interrupt EXT_INT13 mask register	0x0000_00FF
EXT_INT14_MASK	0x0934	External interrupt EXT_INT14 mask register	0x0000_00FF
EXT_INT15_MASK	0x0938	External interrupt EXT_INT15 mask register	0x0000_00FF
EXT_INT16_MASK	0x093C	External interrupt EXT_INT16 mask register	0x0000_003F
EXT_INT21_MASK	0x0940	External interrupt EXT_INT21 mask register	0x0000_00FF
EXT_INT22_MASK	0x0944	External interrupt EXT_INT22 mask register	0x0000_001F
EXT_INT1_PEND	0x0A00	External interrupt EXT_INT1 pending register	0x0000_0000

Register	Offset	Description	Reset Value
EXT_INT2_PEND	0x0A04	External interrupt EXT_INT2 pending register	0x0000_0000
EXT_INT3_PEND	0x0A08	External interrupt EXT_INT3 pending register	0x0000_0000
EXT_INT4_PEND	0x0A0C	External interrupt EXT_INT4 pending register	0x0000_0000
EXT_INT5_PEND	0x0A10	External interrupt EXT_INT5 pending register	0x0000_0000
EXT_INT6_PEND	0x0A14	External interrupt EXT_INT6 pending register	0x0000_0000
EXT_INT7_PEND	0x0A18	External interrupt EXT_INT7 pending register	0x0000_0000
EXT_INT13_PEND	0x0A30	External interrupt EXT_INT13 pending register	0x0000_0000
EXT_INT14_PEND	0x0A34	External interrupt EXT_INT14 pending register	0x0000_0000
EXT_INT15_PEND	0x0A38	External interrupt EXT_INT15 pending register	0x0000_0000
EXT_INT16_PEND	0x0A3C	External interrupt EXT_INT16 pending register	0x0000_0000
EXT_INT21_PEND	0x0A40	External interrupt EXT_INT21 pending register	0x0000_0000
EXT_INT22_PEND	0x0A44	External interrupt EXT_INT22 pending register	0x0000_0000
EXT_INT_SERVICE_XB	0x0B08	Current service register	0x0000_0000
EXT_INT_SERVICE_PEND_XB	0x0B0C	Current service pending register	0x0000_0000
EXT_INT_GRPFIXPRI_XB	0x0B10	External interrupt group fixed priority control register	0x0000_0000
EXT_INT1_FIXPRI	0x0B14	External interrupt 1 fixed priority control register	0x0000_0000
EXT_INT2_FIXPRI	0x0B18	External interrupt 2 fixed priority control register	0x0000_0000
EXT_INT3_FIXPRI	0x0B1C	External interrupt 3 fixed priority control register	0x0000_0000
EXT_INT4_FIXPRI	0x0B20	External interrupt 4 fixed priority control register	0x0000_0000
EXT_INT5_FIXPRI	0x0B24	External interrupt 5 fixed priority control register	0x0000_0000
EXT_INT6_FIXPRI	0x0B28	External interrupt 6 fixed priority control register	0x0000_0000
EXT_INT7_FIXPRI	0x0B2C	External interrupt 7 fixed priority control register	0x0000_0000
EXT_INT13_FIXPRI	0x0B44	External interrupt 13 fixed priority control register	0x0000_0000
EXT_INT14_FIXPRI	0x0B48	External interrupt 14 fixed priority control register	0x0000_0000
EXT_INT15_FIXPRI	0x0B4C	External interrupt 15 fixed priority control register	0x0000_0000
EXT_INT16_FIXPRI	0x0B50	External interrupt 16 fixed priority control register	0x0000_0000
EXT_INT21_FIXPRI	0x0B54	External interrupt 21 fixed priority control register	0x0000_0000
EXT_INT22_FIXPRI	0x0B58	External interrupt 22 fixed priority control register	0x0000_0000
PDNEN	0x0F80	Power down mode pad configure register	0x00

- Base Address: 0x1100_0000

Register	Offset	Description	Reset Value
GPK0CON	0x0040	Port group GPK0 configuration register	0x0000_0000
GPK0DAT	0x0044	Port group GPK0 data register	0x00
GPK0PUD	0x0048	Port group GPK0 pull-up/ pull-down register	0x1555
GPK0DRV	0x004C	Port group GPK0 drive strength control register	0x00_2AAA
GPK0CONPDN	0x0050	Port group GPK0 power down mode configuration register	0x0000
GPK0PUDPDN	0x0054	Port group GPK0 power down mode pull-up/ pull-down register	0x0000
GPK1CON	0x0060	Port group GPK1 configuration register	0x0000_0000
GPK1DAT	0x0064	Port group GPK1 data register	0x00
GPK1PUD	0x0068	Port group GPK1 pull-up/ pull-down register	0x1555
GPK1DRV	0x006C	Port group GPK1 drive strength control register	0x00_0000
GPK1CONPDN	0x0070	Port group GPK1 power down mode configuration register	0x0000
GPK1PUDPDN	0x0074	Port group GPK1 power down mode pull-up/ pull-down register	0x0000
GPK2CON	0x0080	Port group GPK2 configuration register	0x0000_0000
GPK2DAT	0x0084	Port group GPK2 data register	0x00
GPK2PUD	0x0088	Port group GPK2 pull-up/ pull-down register	0x1555
GPK2DRV	0x008C	Port group GPK2 drive strength control register	0x00_0000
GPK2CONPDN	0x0090	Port group GPK2 power down mode configuration register	0x0000
GPK2PUDPDN	0x0094	Port group GPK2 power down mode pull-up/ pull-down register	0x0000
GPK3CON	0x00A0	Port group GPK3 configuration register	0x0000_0000
GPK3DAT	0x00A4	Port group GPK3 data register	0x00
GPK3PUD	0x00A8	Port group GPK3 pull-up/ pull-down register	0x1555
GPK3DRV	0x00AC	Port group GPK3 drive strength control register	0x00_0000
GPK3CONPDN	0x00B0	Port group GPK3 power down mode configuration register	0x0000
GPK3PUDPDN	0x00B4	Port group GPK3 power down mode pull-up/ pull-down register	0x0000
GPL0CON	0x00C0	Port group GPL0 configuration register	0x0000_0000
GPL0DAT	0x00C4	Port group GPL0 data register	0x00
GPL0PUD	0x00C8	Port group GPL0 pull-up/ pull-down register	0x1555
GPL0DRV	0x00CC	Port group GPL0 drive strength control register	0x00_0000
GPL0CONPDN	0x00D0	Port group GPL0 power down mode configuration register	0x0000
GPL0PUDPDN	0x00D4	Port group GPL0 power down mode pull-up/ pull-down register	0x0000
GPL1CON	0x00E0	Port group GPL1 configuration register	0x0000_0000
GPL1DAT	0x00E4	Port group GPL1 data register	0x00

Register	Offset	Description	Reset Value
GPL1PUD	0x00E8	Port group GPL1 pull-up/ pull-down register	0x0005
GPL1DRV	0x00EC	Port group GPL1 drive strength control register	0x00_0000
GPL1CONPDN	0x00F0	Port group GPL1 power down mode configuration register	0x0000
GPL1PUDPDN	0x00F4	Port group GPL1 power down mode pull-up/ pull-down register	0x0000
GPL2CON	0x0100	Port group GPL2 configuration register	0x0000_0000
GPL2DAT	0x0104	Port group GPL2 data register	0x00
GPL2PUD	0x0108	Port group GPL2 pull-up/ pull-down register	0x5555
GPL2DRV	0x010C	Port group GPL2 drive strength control register	0x00_0000
GPL2CONPDN	0x0110	Port group GPL2 power down mode configuration register	0x0000
GPL2PUDPDN	0x0114	Port group GPL2 power down mode pull-up/ pull-down register	0x0000
GPY0CON	0x0120	Port group GPY0 configuration register	0x0000_0000
GPY0DAT	0x0124	Port group GPY0 data register	0x00
GPY0PUD	0x0128	Port group GPY0 pull-up/ pull-down register	0x0FFF
GPY0DRV	0x012C	Port group GPY0 drive strength control register	0x00_0AAA
GPY0CONPDN	0x0130	Port group GPY0 power down mode configuration register	0x0000
GPY0PUDPDN	0x0134	Port group GPY0 power down mode pull-up/ pull-down register	0x0000
GPY1CON	0x0140	Port group GPY1 configuration register	0x0000_0000
GPY1DAT	0x0144	Port group GPY1 data register	0x00
GPY1PUD	0x0148	Port group GPY1 pull-up/ pull-down register	0x00FF
GPY1DRV	0x014C	Port group GPY1 drive strength control register	0x00_00AA
GPY1CONPDN	0x0150	Port group GPY1 power down mode configuration register	0x0000
GPY1PUDPDN	0x0154	Port group GPY1 power down mode pull-up/ pull-down register	0x0000
GPY2CON	0x0160	Port group GPY2 configuration register	0x0000_0000
GPY2DAT	0x0164	Port group GPY2 data register	0x00
GPY2PUD	0x0168	Port group GPY2 pull-up/ pull-down register	0x0FFF
GPY2DRV	0x016C	Port group GPY2 drive strength control register	0x00_0AAA
GPY2CONPDN	0x0170	Port group GPY2 power down mode configuration register	0x0000
GPY2PUDPDN	0x0174	Port group GPY2 power down mode pull-up/ pull-down register	0x0000
GPY3CON	0x0180	Port group GPY3 configuration register	0x0000_0000
GPY3DAT	0x0184	Port group GPY3 data register	0x00
GPY3PUD	0x0188	Port group GPY3 pull-up/ pull-down register	0x5555
GPY3DRV	0x018C	Port group GPY3 drive strength control register	0x00_AAAA
GPY3CONPDN	0x0190	Port group GPY3 power down mode configuration register	0x0000

Register	Offset	Description	Reset Value
GPY3PUDPDN	0x0194	Port group GPY3 power down mode pull-up/ pull-down register	0x0000
GPY4CON	0x01A0	Port group GPY4 configuration register	0x0000_0000
GPY4DAT	0x01A4	Port group GPY4 data register	0x00
GPY4PUD	0x01A8	Port group GPY4 pull-up/ pull-down register	0x5555
GPY4DRV	0x01AC	Port group GPY4 drive strength control register	0x00_AAAA
GPY4CONPDN	0x01B0	Port group GPY4 power down mode configuration register	0x0000
GPY4PUDPDN	0x01B4	Port group GPY4 power down mode pull-up/ pull-down register	0x0000
GPY5CON	0x01C0	Port group GPY5 configuration register	0x0000_0000
GPY5DAT	0x01C4	Port group GPY5 data register	0x00
GPY5PUD	0x01C8	Port group GPY5 pull-up/ pull-down register	0x5555
GPY5DRV	0x01CC	Port group GPY5 drive strength control register	0x00_AAAA
GPY5CONPDN	0x01D0	Port group GPY5 power down mode configuration register	0x0000
GPY5PUDPDN	0x01D4	Port group GPY5 power down mode pull-up/ pull-down register	0x0000
GPY6CON	0x01E0	Port group GPY6 configuration register	0x0000_0000
GPY6DAT	0x01E4	Port group GPY6 data register	0x00
GPY6PUD	0x01E8	Port group GPY6 pull-up/ pull-down register	0x5555
GPY6DRV	0x01EC	Port group GPY6 drive strength control register	0x00_AAAA
GPY6CONPDN	0x01F0	Port group GPY6 power down mode configuration register	0x0000
GPY6PUDPDN	0x01F4	Port group GPY6 power down mode pull-up/ pull-down register	0x0000
ETC0PUD	0x0208	Port group ETC0 pull-up/ pull-down register	0x0400
ETC0DRV	0x020C	Port group ETC0 drive strength control register	0x00_0000
ETC6PUD	0x0228	Port group ETC6 pull-up/ pull-down register	0xC000
ETC6DRV	0x022C	Port group ETC6 drive strength control register	0x00_0000
GPM0CON	0x0260	Port group GPM0 configuration register	0x0000_0000
GPM0DAT	0x0264	Port group GPM0 data register	0x00
GPM0PUD	0x0268	Port group GPM0 pull-up/ pull-down register	0x5555
GPM0DRV	0x026C	Port group GPM0 drive strength control register	0x00_0000
GPM0CONPDN	0x0270	Port group GPM0 power down mode configuration register	0x0000
GPM0PUDPDN	0x0274	Port group GPM0 power down mode pull-up/ pull-down register	0x0000
GPM1CON	0x0280	Port group GPM1 configuration register	0x0000_0000
GPM1DAT	0x0284	Port group GPM1 data register	0x00
GPM1PUD	0x0288	Port group GPM1 pull-up/ pull-down register	0x1555
GPM1DRV	0x028C	Port group GPM1 drive strength control register	0x00_0000

Register	Offset	Description	Reset Value
GPM1CONPDN	0x0290	Port group GPM1 power down mode configuration register	0x0000
GPM1PUDPDN	0x0294	Port group GPM1 power down mode pull-up/ pull-down register	0x0000
GPM2CON	0x02A0	Port group GPM2 configuration register	0x0000_0000
GPM2DAT	0x02A4	Port group GPM2 data register	0x00
GPM2PUD	0x02A8	Port group GPM2 pull-up/ pull-down register	0x0155
GPM2DRV	0x02AC	Port group GPM2 drive strength control register	0x00_0000
GPM2CONPDN	0x02B0	Port group GPM2 power down mode configuration register	0x0000
GPM2PUDPDN	0x02B4	Port group GPM2 power down mode pull-up/ pull-down register	0x0000
GPM3CON	0x02C0	Port group GPM3 configuration register	0x0000_0000
GPM3DAT	0x02C4	Port group GPM3 data register	0x00
GPM3PUD	0x02C8	Port group GPM3 pull-up/ pull-down register	0x5555
GPM3DRV	0x02CC	Port group GPM3 drive strength control register	0x00_0000
GPM3CONPDN	0x02D0	Port group GPM3 power down mode configuration register	0x0000
GPM3PUDPDN	0x02D4	Port group GPM3 power down mode pull-up/ pull-down register	0x0000
GPM4CON	0x02E0	Port group GPM4 configuration register	0x0000_0000
GPM4DAT	0x02E4	Port group GPM4 data register	0x00
GPM4PUD	0x02E8	Port group GPM4 pull-up/ pull-down register	0x5555
GPM4DRV	0x02EC	Port group GPM4 drive strength control register	0x00_0000
GPM4CONPDN	0x02F0	Port group GPM4 power down mode configuration register	0x0000
GPM4PUDPDN	0x02F4	Port group GPM4 power down mode pull-up/ pull-down register	0x0000
EXT_INT23_CON	0x0708	External interrupt EXT_INT23 configuration register	0x0000_0000
EXT_INT24_CON	0x070C	External interrupt EXT_INT24 configuration register	0x0000_0000
EXT_INT25_CON	0x0710	External interrupt EXT_INT25 configuration register	0x0000_0000
EXT_INT26_CON	0x0714	External interrupt EXT_INT26 configuration register	0x0000_0000
EXT_INT27_CON	0x0718	External interrupt EXT_INT27 configuration register	0x0000_0000
EXT_INT28_CON	0x071C	External interrupt EXT_INT28 configuration register	0x0000_0000
EXT_INT29_CON	0x0720	External interrupt EXT_INT29 configuration register	0x0000_0000
EXT_INT8_CON	0x0724	External interrupt EXT_INT8 configuration register	0x0000_0000
EXT_INT9_CON	0x0728	External interrupt EXT_INT9 configuration register	0x0000_0000
EXT_INT10_CON	0x072C	External interrupt EXT_INT10 configuration register	0x0000_0000
EXT_INT11_CON	0x0730	External interrupt EXT_INT11 configuration register	0x0000_0000
EXT_INT12_CON	0x0734	External interrupt EXT_INT12 configuration register	0x0000_0000
EXT_INT23_FLTCON0	0x0810	External interrupt EXT_INT23 filter configuration register 0	0x0000_0000
EXT_INT23_FLTCON1	0x0814	External interrupt EXT_INT23 filter configuration register 1	0x0000_0000

Register	Offset	Description	Reset Value
EXT_INT24_FLTCON0	0x0818	External interrupt EXT_INT24 filter configuration register 0	0x0000_0000
EXT_INT24_FLTCON1	0x081C	External interrupt EXT_INT24 filter configuration register 1	0x0000_0000
EXT_INT25_FLTCON0	0x0820	External interrupt EXT_INT25 filter configuration register 0	0x0000_0000
EXT_INT25_FLTCON1	0x0824	External interrupt EXT_INT25 filter configuration register 1	0x0000_0000
EXT_INT26_FLTCON0	0x0828	External interrupt EXT_INT26 filter configuration register 0	0x0000_0000
EXT_INT26_FLTCON1	0x082C	External interrupt EXT_INT26 filter configuration register 1	0x0000_0000
EXT_INT27_FLTCON0	0x0830	External interrupt EXT_INT27 filter configuration register 0	0x0000_0000
EXT_INT27_FLTCON1	0x0834	External interrupt EXT_INT27 filter configuration register 1	0x0000_0000
EXT_INT28_FLTCON0	0x0838	External interrupt EXT_INT28 filter configuration register 0	0x0000_0000
EXT_INT28_FLTCON1	0x083C	External interrupt EXT_INT28 filter configuration register 1	0x0000_0000
EXT_INT29_FLTCON0	0x0840	External interrupt EXT_INT29 filter configuration register 0	0x0000_0000
EXT_INT29_FLTCON1	0x0844	External interrupt EXT_int29 filter configuration register 1	0x0000_0000
EXT_INT8_FLTCON0	0x0848	External interrupt EXT_INT8 filter configuration register 0	0x0000_0000
EXT_INT8_FLTCON1	0x084C	External interrupt EXT_INT8 filter configuration register 1	0x0000_0000
EXT_INT9_FLTCON0	0x0850	External interrupt EXT_INT9 filter configuration register 0	0x0000_0000
EXT_INT9_FLTCON1	0x0854	External interrupt EXT_INT9 filter configuration register 1	0x0000_0000
EXT_INT10_FLTCON0	0x0858	External interrupt EXT_INT10 filter configuration register 0	0x0000_0000
EXT_INT10_FLTCON1	0x085C	External interrupt EXT_INT10 filter configuration register 1	0x0000_0000
EXT_INT11_FLTCON0	0x0860	External interrupt EXT_INT11 filter configuration register 0	0x0000_0000
EXT_INT11_FLTCON1	0x0864	External interrupt EXT_INT11 filter configuration register 1	0x0000_0000
EXT_INT12_FLTCON0	0x0868	External interrupt EXT_INT12 filter configuration register 0	0x0000_0000
EXT_INT12_FLTCON1	0x086C	External interrupt EXT_INT12 filter configuration register 1	0x0000_0000
EXT_INT23_MASK	0x0908	External interrupt EXT_INT23 mask register	0x0000_007F
EXT_INT24_MASK	0x090C	External interrupt EXT_INT24 mask register	0x0000_007F
EXT_INT25_MASK	0x0910	External interrupt EXT_INT25 mask register	0x0000_007F
EXT_INT26_MASK	0x0914	External interrupt EXT_INT26 mask register	0x0000_007F
EXT_INT27_MASK	0x0918	External interrupt EXT_INT27 mask register	0x0000_007F
EXT_INT28_MASK	0x091C	External interrupt EXT_INT28 mask register	0x0000_0003
EXT_INT29_MASK	0x0920	External interrupt EXT_INT29 mask register	0x0000_00FF
EXT_INT8_MASK	0x0924	External interrupt EXT_INT8 mask register	0x0000_00FF
EXT_INT9_MASK	0x0928	External interrupt EXT_INT9 mask register	0x0000_007F
EXT_INT10_MASK	0x092C	External interrupt EXT_INT10 mask register	0x0000_001F
EXT_INT11_MASK	0x0930	External interrupt EXT_INT11 mask register	0x0000_00FF
EXT_INT12_MASK	0x0934	External interrupt EXT_INT12 mask register	0x0000_00FF
EXT_INT23_PEND	0x0A08	External interrupt EXT_INT23 pending register	0x0000_0000
EXT_INT24_PEND	0x0A0C	External interrupt EXT_INT24 pending register	0x0000_0000
EXT_INT25_PEND	0x0A10	External interrupt EXT_INT25 pending register	0x0000_0000

Register	Offset	Description	Reset Value
EXT_INT26_PEND	0x0A14	External interrupt EXT_INT26 pending register	0x0000_0000
EXT_INT27_PEND	0x0A18	External interrupt EXT_INT27 pending register	0x0000_0000
EXT_INT28_PEND	0x0A1C	External interrupt EXT_INT28 pending register	0x0000_0000
EXT_INT29_PEND	0x0A20	External interrupt EXT_INT29 pending register	0x0000_0000
EXT_INT8_PEND	0x0A24	External interrupt EXT_INT8 pending register	0x0000_0000
EXT_INT9_PEND	0x0A28	External interrupt EXT_INT9 pending register	0x0000_0000
EXT_INT10_PEND	0x0A2C	External interrupt EXT_INT10 pending register	0x0000_0000
EXT_INT11_PEND	0x0A30	External interrupt EXT_INT11 pending register	0x0000_0000
EXT_INT12_PEND	0x0A34	External interrupt EXT_INT12 pending register	0x0000_0000
EXT_INT_SERVICE_XA	0x0B08	Current service register	0x0000_0000
EXT_INT_SERVICE_PEND_XA	0x0B0C	Current service pending register	0x0000_0000
EXT_INT_GRPFIXPRI_XA	0x0B10	External interrupt group fixed priority control register	0x0000_0000
EXT_INT23_FIXPRI	0x0B1C	External interrupt 23 fixed priority control register	0x0000_0000
EXT_INT24_FIXPRI	0x0B20	External interrupt 24 fixed priority control register	0x0000_0000
EXT_INT25_FIXPRI	0x0B24	External interrupt 25 fixed priority control register	0x0000_0000
EXT_INT26_FIXPRI	0x0B28	External interrupt 26 fixed priority control register	0x0000_0000
EXT_INT27_FIXPRI	0x0B2C	External interrupt 27 fixed priority control register	0x0000_0000
EXT_INT28_FIXPRI	0x0B30	External interrupt 28 fixed priority control register	0x0000_0000
EXT_INT29_FIXPRI	0x0B34	External interrupt 29 fixed priority control register	0x0000_0000
EXT_INT8_FIXPRI	0x0B38	External interrupt 8 fixed priority control register	0x0000_0000
EXT_INT9_FIXPRI	0x0B3C	External interrupt 9 fixed priority control register	0x0000_0000
EXT_INT10_FIXPRI	0x0B40	External interrupt 10 fixed priority control register	0x0000_0000
EXT_INT11_FIXPRI	0x0B44	External interrupt 11 fixed priority control register	0x0000_0000
EXT_INT12_FIXPRI	0x0B48	External interrupt 12 fixed priority control register	0x0000_0000
GPX0CON	0x0C00	Port group GPX0 configuration register	0x0000_0000
GPX0DAT	0x0C04	Port group GPX0 data register	0x00
GPX0PUD	0x0C08	Port group GPX0 pull-up/ pull-down register	0x5555
GPX0DRV	0x0C0C	Port group GPX0 drive strength control register	0x00_0000
GPX1CON	0x0C20	Port group GPX1 configuration register	0x0000_0000
GPX1DAT	0x0C24	Port group GPX1 data register	0x00
GPX1PUD	0x0C28	Port group GPX1 pull-up/ pull-down register	0x5555
GPX1DRV	0x0C2C	Port group GPX1 drive strength control register	0x00_0000
GPX2CON	0x0C40	Port group GPX2 configuration register	0x0000_0000
GPX2DAT	0x0C44	Port group GPX2 data register	0x00

Register	Offset	Description	Reset Value
GPX2PUD	0x0C48	Port group GPX2 pull-up/ pull-down register	0x5555
GPX2DRV	0x0C4C	Port group GPX2 drive strength control register	0x00_0000
GPX3CON	0x0C60	Port group GPX3 configuration register	0x0000_0000
GPX3DAT	0x0C64	Port group GPX3 data register	0x00
GPX3PUD	0x0C68	Port group GPX3 pull-up/ pull-down register	0x5555
GPX3DRV	0x0C6C	Port group GPX3 drive strength control register	0x00_0000
EXT_INT40_CON	0x0E00	External interrupt EXT_INT40 configuration register	0x0000_0000
EXT_INT41_CON	0x0E04	External interrupt EXT_INT41 configuration register	0x0000_0000
EXT_INT42_CON	0x0E08	External interrupt EXT_INT42 configuration register	0x0000_0000
EXT_INT43_CON	0x0E0C	External interrupt EXT_INT43 configuration register	0x0000_0000
EXT_INT40_FLTCON0	0x0E80	External Interrupt EXT_INT40 filter configuration register 0	0x8080_8080
EXT_INT40_FLTCON1	0x0E84	External interrupt EXT_INT40 filter configuration register 1	0x8080_8080
EXT_INT41_FLTCON0	0x0E88	External interrupt EXT_INT41 filter configuration register 0	0x8080_8080
EXT_INT41_FLTCON1	0x0E8C	External interrupt EXT_INT41 filter configuration register 1	0x8080_8080
EXT_INT42_FLTCON0	0x0E90	External interrupt EXT_INT42 filter configuration register 0	0x8080_8080
EXT_INT42_FLTCON1	0x0E94	External interrupt EXT_INT42 filter configuration register 1	0x8080_8080
EXT_INT43_FLTCON0	0x0E98	External interrupt EXT_INT43 filter configuration register 0	0x8080_8080
EXT_INT43_FLTCON1	0x0E9C	External interrupt EXT_INT43 filter configuration register 1	0x8080_8080
EXT_INT40_MASK	0x0F00	External interrupt EXT_INT40 mask register	0x0000_00FF
EXT_INT41_MASK	0x0F04	External interrupt EXT_INT41 mask register	0x0000_00FF
EXT_INT42_MASK	0x0F08	External interrupt EXT_INT42 mask register	0x0000_00FF
EXT_INT43_MASK	0x0F0C	External interrupt EXT_INT43 mask register	0x0000_00FF
EXT_INT40_PEND	0x0F40	External interrupt EXT_INT40 pending register	0x0000_0000
EXT_INT41_PEND	0x0F44	External interrupt EXT_INT41 pending register	0x0000_0000
EXT_INT42_PEND	0x0F48	External interrupt EXT_INT42 pending register	0x0000_0000
EXT_INT43_PEND	0x0F4C	External interrupt EXT_INT43 pending register	0x0000_0000
PDNEN	0x0F80	Power down mode pad configure register	0x00

- Base Address: 0x0386_0000

Register	Offset	Description	Reset Value
GPZCON	0x0000	Port group GPIO group Z (GPZ) configuration register	0x0000_0000
GPZDAT	0x0004	Port group GPZ data register	0x00
GPZPUD	0x0008	Port group GPZ pull-up/ pull-down register	0x1555
GPZDRV	0x000C	Port group GPZ drive strength control register	0x00_0000
GPZCONPDN	0x0010	Port group GPZ power down mode configuration register	0x0000
GPZPUDPDN	0x0014	Port group GPZ power down mode pull-up/ pull-down register	0x0000
EXT_INT50_CON	0x0700	External interrupt EXT_INT50 configuration register	0x0000_0000
EXT_INT50_FLTCON0	0x0800	External interrupt EXT_INT50 filter configuration register 0	0x0000_0000
EXT_INT50_FLTCON1	0x0804	External interrupt EXT_INT50 filter configuration register 1	0x0000_0000
EXT_INT50_MASK	0x0900	External interrupt EXT_INT50 mask register	0x0000_007F
EXT_INT50_PEND	0x0A00	External interrupt EXT_INT50 pending register	0x0000_0000
EXT_INT_SERVICE_XD	0x0B08	Current service register	0x0000_0000
EXT_INT_SERVICE_PEND_XD	0x0B0C	Current service pending register	0x0000_0000
EXT_INT_GRPFIXPRI_XD	0x0B10	External interrupt group fixed priority control register	0x0000_0000
EXT_INT50_FIXPRI	0x0B14	External interrupt 50 fixed priority control register	0x0000_0000
PDNEN	0x0F80	Power down mode pad configure register	0x00

- Base Address: 0x106E_0000

Register	Offset	Description	Reset Value
GPV0CON	0x0000	Port group GPV0 configuration register	0x0000_0000
GPV0DAT	0x0004	Port group GPV0 data register	0x00
GPV0PUD	0x0008	Port group GPV0 pull-up/ pull-down register	0x5555
GPV0DRV	0x000C	Port group GPV0 drive strength control register	0x00_0000
GPV0CONPDN	0x0010	Port group GPV0 power down mode configuration register	0x0000
GPV0PUDPDN	0x0014	Port group GPV0 power down mode pull-up/ pull-down register	0x0000
GPV1CON	0x0020	Port group GPV1 configuration register	0x0000_0000
GPV1DAT	0x0024	Port group GPV1 data register	0x00
GPV1PUD	0x0028	Port group GPV1 pull-up/ pull-down register	0x5555
GPV1DRV	0x002C	Port group GPV1 drive strength control register	0x00_0000
GPV1CONPDN	0x0030	Port group GPV1 power down mode configuration register	0x0000
GPV1PUDPDN	0x0034	Port group GPV1 power down mode pull-up/ pull-down register	0x0000
ETC7PUD	0x0048	Port group ETC7 pull-up/ pull-down register	0x0005
ETC7DRV	0x004C	Port group ETC7 drive strength control register	0x00_0000
GPV2CON	0x0060	Port group GPV2 configuration register	0x0000_0000
GPV2DAT	0x0064	Port group GPV2 data register	0x00
GPV2PUD	0x0068	Port group GPV2 pull-up/ pull-down register	0x5555
GPV2DRV	0x006C	Port group GPV2 drive strength control register	0x00_0000
GPV2CONPDN	0x0070	Port group GPV2 power down mode configuration register	0x0000
GPV2PUDPDN	0x0074	Port group GPV2 power down mode pull-up/ pull-down register	0x0000
GPV3CON	0x0080	Port group GPV3 configuration register	0x0000_0000
GPV3DAT	0x0084	Port group GPV3 data register	0x00
GPV3PUD	0x0088	Port group GPV3 pull-up/ pull-down register	0x5555
GPV3DRV	0x008C	Port group GPV3 drive strength control register	0x00_0000
GPV3CONPDN	0x0090	Port group GPV3 power down mode configuration register	0x0000
GPV3PUDPDN	0x0094	Port group GPV3 power down mode pull-up/ pull-down register	0x0000
ETC8PUD	0x00A8	Port group ETC8 pull-up/ pull-down register	0x0005
ETC8DRV	0x00AC	Port group ETC8 drive strength control register	0x00_0000
GPV4CON	0x00C0	Port group GPV4 configuration register	0x0000_0000
GPV4DAT	0x00C4	Port group GPV4 data register	0x00
GPV4PUD	0x00C8	Port group GPV4 pull-up/ pull-down register	0x0005
GPV4DRV	0x00CC	Port group GPV4 drive strength control register	0x00_0000
GPV4CONPDN	0x00D0	Port group GPV4 power down mode configuration register	0x0000

Register	Offset	Description	Reset Value
GPV4PUDPDN	0x00D4	Port group GPV4 power down mode pull-up/ pull-down register	0x0000
EXT_INT30_CON	0x0700	External interrupt EXT_INT30 configuration register	0x0000_0000
EXT_INT31_CON	0x0704	External interrupt EXT_INT31 configuration register	0x0000_0000
EXT_INT32_CON	0x0708	External interrupt EXT_INT32 configuration register	0x0000_0000
EXT_INT33_CON	0x070C	External interrupt EXT_INT33 configuration register	0x0000_0000
EXT_INT34_CON	0x0710	External interrupt EXT_INT34 configuration register	0x0000_0000
EXT_INT30_FLTCON0	0x0800	External interrupt EXT_INT30 filter configuration register 0	0x0000_0000
EXT_INT30_FLTCON1	0x0804	External interrupt EXT_INT30 filter configuration register 1	0x0000_0000
EXT_INT31_FLTCON0	0x0808	External interrupt EXT_INT31 filter configuration register 0	0x0000_0000
EXT_INT31_FLTCON1	0x080C	External interrupt EXT_INT31 filter configuration register 1	0x0000_0000
EXT_INT32_FLTCON0	0x0810	External interrupt EXT_INT32 filter configuration register 0	0x0000_0000
EXT_INT32_FLTCON1	0x0814	External interrupt EXT_INT32 filter configuration register 1	0x0000_0000
EXT_INT33_FLTCON0	0x0818	External interrupt EXT_INT33 filter configuration register 0	0x0000_0000
EXT_INT33_FLTCON1	0x081C	External interrupt EXT_INT33 filter configuration register 1	0x0000_0000
EXT_INT34_FLTCON0	0x0820	External interrupt EXT_INT34 filter configuration register 0	0x0000_0000
EXT_INT34_FLTCON1	0x0824	External interrupt EXT_INT34 filter configuration register 1	0x0000_0000
EXT_INT30_MASK	0x0900	External interrupt EXT_INT30 mask register	0x0000_00FF
EXT_INT31_MASK	0x0904	External interrupt EXT_INT31 mask register	0x0000_00FF
EXT_INT32_MASK	0x0908	External interrupt EXT_INT32 mask register	0x0000_00FF
EXT_INT33_MASK	0x090C	External interrupt EXT_INT33 mask register	0x0000_00FF
EXT_INT34_MASK	0x0910	External interrupt EXT_INT34 mask register	0x0000_0003
EXT_INT30_PEND	0x0A00	External interrupt EXT_INT30 pending register	0x0000_0000
EXT_INT31_PEND	0x0A04	External interrupt EXT_INT31 pending register	0x0000_0000
EXT_INT32_PEND	0x0A08	External interrupt EXT_INT32 pending register	0x0000_0000
EXT_INT33_PEND	0x0A0C	External interrupt EXT_INT33 pending register	0x0000_0000
EXT_INT34_PEND	0x0A10	External interrupt EXT_INT34 pending register	0x0000_0000
EXT_INT_SERVICE_XC	0x0B08	Current service register	0x0000_0000
EXT_INT_SERVICE_PEND_XC	0x0B0C	Current service pending register	0x0000_0000
EXT_INT_GRPFIXPRI_XC	0x0B10	External interrupt group fixed priority control register	0x0000_0000
EXT_INT30_FIXPRI	0x0B14	External interrupt 30 fixed priority control register	0x0000_0000
EXT_INT31_FIXPRI	0x0B18	External interrupt 31 fixed priority control register	0x0000_0000
EXT_INT32_FIXPRI	0x0B1C	External interrupt 32 fixed priority control register	0x0000_0000
EXT_INT33_FIXPRI	0x0B20	External interrupt 33 fixed priority control register	0x0000_0000
EXT_INT34_FIXPRI	0x0B24	External interrupt 34 fixed priority control register	0x0000_0000

Register	Offset	Description	Reset Value
PDNEN	0x0F80	Power down mode pad configure register	0x00

6.2.2 Part 1

For the following SFRs, Sets the value does not take effect immediately. It takes at least 800 APB clocks for the value to take effect after the SFR is actually changed: The SFRs are:

GPA0PUD, GPA0DRV, GPA1PUD, GPA1DRV, GPBPUD, GPBDRV, GPC0PUD, GPC0DRV, GPC1PUD, GPC1DRV, GPD0PUD, GPD0DRV, GPD1PUD, GPD1DRV, GPF0PUD, GPF0DRV, GPF1PUD, GPF1DRV, GPF2PUD, GPF2DRV, GPF3PUD, GPF3DRV, GPJ0PUD, GPJ0DRV, GPJ1PUD, GPJ1DRV.

6.2.2.1 GPA0CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPA0CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = UART_1_RTSn 0x3 = I2C_2_SCL 0x4 to 0xE = Reserved 0xF = EXT_INT1[7]	0x00
GPA0CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = UART_1_CTSn 0x3 = I2C_2_SDA 0x4 to 0xE = Reserved 0xF = EXT_INT1[6]	0x00
GPA0CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = UART_1_TXD 0x3 to 0xE = Reserved 0xF = EXT_INT1[5]	0x00
GPA0CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = UART_1_RXD 0x3 to 0xE = Reserved 0xF = EXT_INT1[4]	0x00
GPA0CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = UART_0_RTSn 0x3 to 0xE = Reserved 0xF = EXT_INT1[3]	0x00
GPA0CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = UART_0_CTSn 0x3 to 0xE = Reserved 0xF = EXT_INT1[2]	0x00
GPA0CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = UART_0_TXD 0x3 to 0xE = Reserved 0xF = EXT_INT1[1]	0x00
GPA0CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = UART_0_RXD 0x3 to 0xE = Reserved 0xF = EXT_INT1[0]	0x00

6.2.2.2 GPA0DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPA0DAT[7:0]	[7:0]	RWX	When you configure port as input port, then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.3 GPA0PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0008, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPA0PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.2.4 GPA0DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x000C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPA0DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.2.5 GPA0CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPA0[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.2.6 GPA0PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPA0[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.2.7 GPA1CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPA1CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = UART_3_TXD 0x3 = Reserved 0x4 = UART_AUDIO_TXD 0x5 to 0xE = Reserved 0xF = EXT_INT2[5]	0x00
GPA1CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = UART_3_RXD 0x3 = Reserved 0x4 = UART_AUDIO_RXD 0x5 to 0xE = Reserved 0xF = EXT_INT2[4]	0x00
GPA1CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = UART_2_RTSn 0x3 = I2C_3_SCL 0x4 to 0xE = Reserved 0xF = EXT_INT2[3]	0x00
GPA1CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = UART_2_CTSn 0x3 = I2C_3_SDA 0x4 to 0xE = Reserved 0xF = EXT_INT2[2]	0x00
GPA1CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = UART_2_TXD 0x3 = Reserved 0x4 = UART_AUDIO_TXD 0x5 to 0xE = Reserved 0xF = EXT_INT2[1]	0x00
GPA1CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = UART_2_RXD 0x3 = Reserved 0x4 = UART_AUDIO_RXD 0x5 to 0xE = Reserved 0xF = EXT_INT2[0]	0x00

6.2.2.8 GPA1DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0024, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPA1DAT[5:0]	[5:0]	RWX	When you configure port as input port, then corresponding bit is the pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.9 GPA1PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0028, Reset Value = 0x0555

Name	Bit	Type	Description	Reset Value
GPA1PUD[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0555

6.2.2.10 GPA1DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x002C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPA1DRV [n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 5	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.2.11 GPA1CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPA1[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.2.12 GPA1PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPA1[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.2.13 GPBCON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPBCON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = SPI_1_MOSI 0x3 to 0xE = Reserved 0xF = EXT_INT3[7]	0x00
GPBCON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = SPI_1_MISO 0x3 to 0xE = Reserved 0xF = EXT_INT3[6]	0x00
GPBCON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = SPI_1_nSS 0x3 = Reserved 0x4 = IEM_SPWI 0x5 to 0xE = Reserved 0xF = EXT_INT3[5]	0x00
GPBCON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = SPI_1_CLK 0x3 = Reserved 0x4 = IEM_SCLK 0x5 to 0xE = Reserved 0xF = EXT_INT3[4]	0x00
GPBCON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = SPI_0_MOSI 0x3 = I2C_5_SCL 0x4 to 0xE = Reserved 0xF = EXT_INT3[3]	0x00
GPBCON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = SPI_0_MISO 0x3 = I2C_5_SDA 0x4 to 0xE = Reserved 0xF = EXT_INT3[2]	0x00
GPBCON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = SPI_0_nSS 0x3 = I2C_4_SCL 0x4 to 0xE = Reserved 0xF = EXT_INT3[1]	0x00
GPBCON[0]	[3:0]	RW	0x0 = Input 0x1 = Output	0x00

Name	Bit	Type	Description	Reset Value
			0x2 = SPI_0_CLK 0x3 = I2C_4_SDA 0x4 to 0xE = Reserved 0xF = EXT_INT3[0]	

6.2.2.14 GPBDAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0044, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPBDAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.15 GPBPUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0048, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPBPUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.2.16 GPBDRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x004C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPBDRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4	0x0000

6.2.2.17 GPBCONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPB[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.2.18 GPBPUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPB[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved, 0x3 = Enables Pull-up	0x00

6.2.2.19 GPC0CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPC0CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_1_SDO 0x3 = PCM_1_SOUT 0x4 = AC97SDO 0x5 to 0xE = Reserved 0xF = EXT_INT4[4]	0x00
GPC0CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_1_SDI 0x3 = PCM_1_SIN 0x4 = AC97SDI 0x5 to 0xE = Reserved 0xF = EXT_INT4[3]	0x00
GPC0CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_1_LRCK 0x3 = PCM_1_FSYNC 0x4 = AC97SYNC 0x5 to 0xE = Reserved 0xF = EXT_INT4[2]	0x00
GPC0CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_1_CDCLK 0x3 = PCM_1_EXTCLK 0x4 = AC97RESETn 0x5 to 0xE = Reserved 0xF = EXT_INT4[1]	0x00
GPC0CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_1_SCLK 0x3 = PCM_1_SCLK 0x4 = AC97BITCLK 0x5 to 0xE = Reserved 0xF = EXT_INT4[0]	0x00

6.2.2.20 GPC0DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0064, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPC0DAT[4:0]	[4:0]	RWX	When you configure as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.21 GPC0PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0068, Reset Value = 0x0155

Name	Bit	Type	Description	Reset Value
GPC0PUD[n]	[2n + 1:2n] n = 0 to 4	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0155

6.2.2.22 GPC0DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x006C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPC0DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 4	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.2.23 GPC0CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0070, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPC0[n]	[2n + 1:2n] n = 0 to 4	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.2.24 GPC0PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0074, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPC0[n]	[2n + 1:2n] n = 0 to 4	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.2.25 GPC1CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPC1CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_2_SDO 0x3 = PCM_2_SOUT 0x4 = I2C_6_SCL 0x5 = SPI_2_MOSI 0x6 to 0xE = Reserved 0xF = EXT_INT5[4]	0x00
GPC1CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_2_SDI 0x3 = PCM_2_SIN 0x4 = I2C_6_SDA 0x5 = SPI_2_MISO 0x6 to 0xE = Reserved 0xF = EXT_INT5[3]	0x00
GPC1CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_2_LRCK 0x3 = PCM_2_FSYNC 0x4 = Reserved 0x5 = SPI_2_nSS 0x6 to 0xE = Reserved 0xF = EXT_INT5[2]	0x00
GPC1CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_2_CDCLK 0x3 = PCM_2_EXTCLK 0x4 = SPDIF_EXTCLK 0x5 = SPI_2_CLK 0x6 to 0xE = Reserved 0xF = EXT_INT5[1]	0x00
GPC1CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_2_SCLK 0x3 = PCM_2_SCLK 0x4 = SPDIF_0_OUT 0x5 to 0xE = Reserved 0xF = EXT_INT5[0]	0x00

6.2.2.26 GPC1DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0084, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPC1DAT[4:0]	[4:0]	RWX	When you configure port as input port, corresponding bit is pin state. When configuring as output port, pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.27 GPC1PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0088, Reset Value = 0x0155

Name	Bit	Type	Description	Reset Value
GPC1PUD[n]	[2n + 1:2n] n = 0 to 4	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0155

6.2.2.28 GPC1DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x008C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPC1DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 4	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.2.29 GPC1CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPC1[n]	[2n + 1:2n] n = 0 to 4	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.2.30 GPC1PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0094, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPC1[n]	[2n + 1:2n] n = 0 to 4	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.2.31 GPD0CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPD0CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = TOUT_3 0x3 = I2C_7_SCL 0x4 to 0xE = Reserved 0xF = EXT_INT6[3]	0x00
GPD0CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = TOUT_2 0x3 = I2C_7_SDA 0x4 to 0xE = Reserved 0xF = EXT_INT6[2]	0x00
GPD0CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = TOUT_1 0x3 = LCD_PWM 0x4 to 0xE = Reserved 0xF = EXT_INT6[1]	0x00
GPD0CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = TOUT_0 0x3 = LCD_FRM 0x4 to 0xE = Reserved 0xF = EXT_INT6[0]	0x00

6.2.2.32 GPD0DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x00A4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPD0DAT[3:0]	[3:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.33 GPD0PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x00A8, Reset Value = 0x0055

Name	Bit	Type	Description	Reset Value
GPD0PUD[n]	[2n + 1:2n] n = 0 to 3	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0055

6.2.2.34 GPD0DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x00AC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPD0DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 3	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.2.35 GPD0CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x00B0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPD0[n]	[2n + 1:2n] n = 0 to 3	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.2.36 GPD0PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x00B4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPD0[n]	[2n + 1:2n] n = 0 to 3	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.2.37 GPD1CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPD1CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = I2C_1_SCL 0x3 = MIPI1_ESC_CLK 0x4 to 0xE = Reserved 0xF = EXT_INT7[3]	0x00
GPD1CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = I2C_1_SDA 0x3 = MIPI1_BYTE_CLK 0x4 to 0xE = Reserved 0xF = EXT_INT7[2]	0x00
GPD1CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = I2C_0_SCL 0x3 = MIPI0_ESC_CLK 0x4 to 0xE = Reserved 0xF = EXT_INT7[1]	0x00
GPD1CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = I2C_0_SDA 0x3 = MIPI0_BYTE_CLK 0x4 to 0xE = Reserved 0xF = EXT_INT7[0]	0x00

6.2.2.38 GPD1DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x00C4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPD1DAT[3:0]	[3:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.39 GPD1PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x00C8, Reset Value = 0x0055

Name	Bit	Type	Description	Reset Value
GPD1PUD[n]	[2n + 1:2n] n = 0 to 3	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved, 0x3 = Enables Pull-up	0x0055

6.2.2.40 GPD1DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x00CC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPD1DRV[n]	[23:16]	RW	Reserved (should be zero)	0x00
	[2n + 1:2n] n = 0 to 3	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.2.41 GPD1CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x00D0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPD1[n]	[2n + 1:2n] n = 0 to 3	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.2.42 GPD1PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x00D4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPD1[n]	[2n + 1:2n] n = 0 to 3	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.2.43 GPF0CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0180, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPF0CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[3] 0x3 to 0xE = Reserved 0xF = EXT_INT13[7]	0x00
GPF0CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[2] 0x3 to 0xE = Reserved 0xF = EXT_INT13[6]	0x00
GPF0CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[1] 0x3 to 0xE = Reserved 0xF = EXT_INT13[5]	0x00
GPF0CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[0] 0x3 to 0xE = Reserved 0xF = EXT_INT13[4]	0x00
GPF0CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VCLK 0x3 to 0xE = Reserved 0xF = EXT_INT13[3]	0x00
GPF0CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output, 0x2 = LCD_VDEN, 0x3 to 0xE = Reserved, 0xF = EXT_INT13[2]	0x00
GPF0CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VSYNC 0x3 to 0xE = Reserved 0xF = EXT_INT13[1]	0x00
GPF0CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_HSYNC 0x3 to 0xE = Reserved 0xF = EXT_INT13[0]	0x00

6.2.2.44 GPF0DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0184, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPF0DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.45 GPF0PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0188, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPF0PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.2.46 GPF0DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x018C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPF0DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.2.47 GPF0CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0190, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPF0[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.2.48 GPF0PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0194, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPF0[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.2.49 GPF1CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x01A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPF1CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[11] 0x3 to 0xE = Reserved 0xF = EXT_INT14[7]	0x00
GPF1CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[10] 0x3 to 0xE = Reserved 0xF = EXT_INT14[6]	0x00
GPF1CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[9] 0x3 to 0xE = Reserved 0xF = EXT_INT14[5]	0x00
GPF1CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[8] 0x3 to 0xE = Reserved 0xF = EXT_INT14[4]	0x00
GPF1CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[7] 0x3 to 0xE = Reserved 0xF = EXT_INT14[3]	0x00
GPF1CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[6] 0x3 to 0xE = Reserved 0xF = EXT_INT14[2]	0x00
GPF1CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[5] 0x3 to 0xE = Reserved 0xF = EXT_INT14[1]	0x00
GPF1CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[4] 0x3 to 0xE = Reserved 0xF = EXT_INT14[0]	0x00

6.2.2.50 GPF1DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x01A4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPF1DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.51 GPF1PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x01A8, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPF1PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.2.52 GPF1DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x01AC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPF1DRV[n]	[23:16]	RW	Reserved (should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.2.53 GPF1CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x01B0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPF1[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0, 0x1 = Outputs 1, 0x2 = Input, 0x3 = Previous state	0x00

6.2.2.54 GPF1PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x01B4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPF1[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved, 0x3 = Enables Pull-up	0x00

6.2.2.55 GPF2CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x01C0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPF2CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[19] 0x3 to 0xE = Reserved 0xF = EXT_INT15[7]	0x00
GPF2CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[18] 0x3 to 0xE = Reserved 0xF = EXT_INT15[6]	0x00
GPF2CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[17] 0x3 to 0xE = Reserved 0xF = EXT_INT15[5]	0x00
GPF2CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[16] 0x3 to 0xE = Reserved 0xF = EXT_INT15[4]	0x00
GPF2CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[15] 0x3 to 0xE = Reserved 0xF = EXT_INT15[3]	0x00
GPF2CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[14] 0x3 to 0xE = Reserved 0xF = EXT_INT15[2]	0x00
GPF2CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output, 0x2 = LCD_VD[13], 0x3 to 0xE = Reserved 0xF = EXT_INT15[1]	0x00
GPF2CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[12] 0x3 to 0xE = Reserved, 0xF = EXT_INT15[0]	0x00

6.2.2.56 GPF2DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x01C4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPF2DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.57 GPF2PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x01C8, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPF2PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.2.58 GPF2DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x01CC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPF2DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.2.59 GPF2CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x01D0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPF2[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.2.60 GPF2PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x01D4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPF2[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.2.61 GPF3CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x01E0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPF3CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = SYS_OE 0x3 to 0xE = Reserved 0xF = EXT_INT16[5]	0x00
GPF3CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = VSYNC_LDI 0x3 to 0xE = Reserved 0xF = EXT_INT16[4]	0x00
GPF3CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[23] 0x3 to 0xE = Reserved 0xF = EXT_INT16[3]	0x00
GPF3CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[22] 0x3 to 0xE = Reserved 0xF = EXT_INT16[2]	0x00
GPF3CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[21] 0x3 to 0xE = Reserved 0xF = EXT_INT16[1]	0x00
GPF3CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = LCD_VD[20] 0x3 to 0xE = Reserved 0xF = EXT_INT16[0]	0x00

6.2.2.62 GPF3DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x01E4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPF3DAT[5:0]	[5:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.63 GPF3PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x01E8, Reset Value = 0x0555

Name	Bit	Type	Description	Reset Value
GPF3PUD[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0555

6.2.2.64 GPF3DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x01EC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPF3DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 5	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.2.65 GPF3CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x01F0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPF3[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.2.66 GPF3PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x01F4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPF3[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.2.67 ETC1PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0228, Reset Value = 0x0005

Name	Bit	Type	Description	Reset Value
ETC1PUD[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0005

ETC1PUD[1:0] controls XsbusDATA.

ETC1PUD[3:2] controls XsbusCLK.

6.2.2.68 ETC1DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x022C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
ETC1DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 5	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

ETC1PUD[1:0] controls XsbusDATA.

ETC1PUD[3:2] controls XsbusCLK.

6.2.2.69 GPJ0CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0240, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPJ0CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[4] 0x3 to 0xE = Reserved 0xF = EXT_INT21[7]	0x00
GPJ0CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[3] 0x3 to 0xE = Reserved 0xF = EXT_INT21[6]	0x00
GPJ0CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[2] 0x3 to 0xE = Reserved 0xF = EXT_INT21[5]	0x00
GPJ0CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[1] 0x3 to 0xE = Reserved 0xF = EXT_INT21[4]	0x00
GPJ0CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[0] 0x3 to 0xE = Reserved 0xF = EXT_INT21[3]	0x00
GPJ0CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_HREF 0x3 to 0xE = Reserved 0xF = EXT_INT21[2]	0x00
GPJ0CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_VSYNC 0x3 to 0xE = Reserved 0xF = EXT_INT21[1]	0x00
GPJ0CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_PCLK 0x3 to 0xE = Reserved 0xF = EXT_INT21[0]	0x00

6.2.2.70 GPJ0DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0244, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPJ0DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.71 GPJ0PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0248, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPJ0PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.2.72 GPJ0DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x024C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPJ0DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4	0x0000

6.2.2.73 GPJ0CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0250, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPJ0[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.2.74 GPJ0PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0254, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPJ0[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.2.75 GPJ1CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0260, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPJ1CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_FIELD 0x3 to 0xE = Reserved 0xF = EXT_INT22[4]	0x00
GPJ1CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_CLKOUT 0x3 to 0xE = Reserved 0xF = EXT_INT22[3]	0x00
GPJ1CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[7] 0x3 to 0xE = Reserved 0xF = EXT_INT22[2]	0x00
GPJ1CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[6] 0x3 to 0xE = Reserved 0xF = EXT_INT22[1]	0x00
GPJ1CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[5] 0x3 to 0xE = Reserved 0xF = EXT_INT22[0]	0x00

6.2.2.76 GPJ1DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0264, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPJ1DAT[4:0]	[4:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.2.77 GPJ1PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0268, Reset Value = 0x0155

Name	Bit	Type	Description	Reset Value
GPJ1PUD[n]	[2n + 1:2n] n = 0 to 4	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0155

6.2.2.78 GPJ1DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x026C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPJ1DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 4	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.2.79 GPJ1CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0270, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPJ1[n]	[2n + 1:2n] n = 0 to 4	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.2.80 GPJ1PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0274, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPJ1[n]	[2n + 1:2n] n = 0 to 4	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.2.81 EXT_INT1CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0700, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT1_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT1[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT1_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT1[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT1_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT1[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT1_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT1[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT1_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT1[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT1_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT1[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT1_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT1[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT1_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT1[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.82 EXT_INT2CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0704, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
RSVD	[23]	–	Reserved	0x0
EXT_INT2_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT2[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT2_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT2[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT2_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT2[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT2_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT2[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT2_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT2[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

Name	Bit	Type	Description	Reset Value
RSVD	[3]	–	Reserved	0x0
EXT_INT2_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT2[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.83 EXT_INT3CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0708, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT3_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT3[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT3_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT3[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT3_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT3[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT3_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT3[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT3_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT3[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT3_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT3[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT3_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT3[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT3_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT3[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.84 EXT_INT4CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x070C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	0x000
RSVD	[19]	–	Reserved	0x0
EXT_INT4_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT4[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT4_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT4[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT4_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT4[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT4_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT4[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT4_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT4[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.85 EXT_INT5CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0710, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	0x000
RSVD	[19]	–	Reserved	0x0
EXT_INT5_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT5[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT5_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT5[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT5_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT5[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT5_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT5[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT5_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT5[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.86 EXT_INT6CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0714, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0000
RSVD	[15]	–	Reserved	0x0
EXT_INT6_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT6[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT6_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT6[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT6_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT6[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT6_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT6[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.87 EXT_INT7CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0718, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0000
RSVD	[15]	–	Reserved	0x0
EXT_INT7_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT7[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT7_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT7[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT7_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT7[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT7_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT7[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.88 EXT_INT13CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0730, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT13_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT13[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT13_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT13[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT13_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT13[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT13_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT13[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT13_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT13[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT13_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT13[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT13_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT13[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT13_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT13[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.89 EXT_INT14CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0734, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT14_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT14[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT14_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT14[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT14_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT14[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT14_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT14[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT14_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT14[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT14_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT14[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT14_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT14[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT14_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT14[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.90 EXT_INT15CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0738, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT15_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT15[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT15_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT15[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT15_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT15[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT15_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT15[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT15_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT15[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT15_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT15[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT15_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT15[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT15_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT15[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.91 EXT_INT16CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x073C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
RSVD	[23]	–	Reserved	0x0
EXT_INT16_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT16[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT16_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT16[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT16_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT16[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT16_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT16[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT16_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT16[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

Name	Bit	Type	Description	Reset Value
RSVD	[3]	–	Reserved	0x0
EXT_INT16_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT16[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.92 EXT_INT21CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0740, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT21_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT21[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT21_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT21[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT21_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT21[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT21_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT21[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT21_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT21[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT21_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT21[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT21_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT21[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT21_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT21[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.93 EXT_INT22CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0744, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	0x000
RSVD	[19]	–	Reserved	0x0
EXT_INT22_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT22[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT22_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT22[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT22_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT22[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT22_CON[1]	[6:4]	W	Sets signaling method of EXT_INT22[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT22_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT22[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.2.94 EXT_INT1_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0800, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN1[3]	[31]	RW	Filter Enable for EXT_INT1[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH1[3]	[30:24]	RW	Filtering width of EXT_INT1[3]	0x00
FLTEN1[2]	[23]	RW	Filter Enable for EXT_INT1[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH1[2]	[22:16]	RW	Filtering width of EXT_INT1[2]	0x00
FLTEN1[1]	[15]	RW	Filter Enable for EXT_INT1[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH1[1]	[14:8]	RW	Filtering width of EXT_INT1[1]	0x00
FLTEN1[0]	[7]	RW	Filter Enable for EXT_INT1[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH1[0]	[6:0]	RW	Filtering width of EXT_INT1[0]	0x00

6.2.2.95 EXT_INT1_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0804, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN1[7]	[31]	RW	Filter Enable for EXT_INT1[7] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH1[7]	[30:24]	RW	Filtering width of EXT_INT1[7]	0x00
FLTEN1[6]	[23]	RW	Filter Enable for EXT_INT1[6] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH1[6]	[22:16]	RW	Filtering width of EXT_INT1[6]	0x00
FLTEN1[5]	[15]	RW	Filter Enable for EXT_INT1[5] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH1[5]	[14:8]	RW	Filtering width of EXT_INT1[5]	0x00
FLTEN1[4]	[7]	RW	Filter Enable for EXT_INT1[4] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH1[4]	[6:0]	RW	Filtering width of EXT_INT1[4]	0x00

6.2.2.96 EXT_INT2_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0808, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN2[3]	[31]	RW	Filter Enable for EXT_INT2[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH2[3]	[30:24]	RW	Filtering width of EXT_INT2[3]	0x00
FLTEN2[2]	[23]	RW	Filter Enable for EXT_INT2[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH2[2]	[22:16]	RW	Filtering width of EXT_INT2[2]	0x00
FLTEN2[1]	[15]	RW	Filter Enable for EXT_INT2[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH2[1]	[14:8]	RW	Filtering width of EXT_INT2[1]	0x00
FLTEN2[0]	[7]	RW	Filter Enable for EXT_INT2[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH2[0]	[6:0]	RW	Filtering width of EXT_INT2[0]	0x00

6.2.2.97 EXT_INT2_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x080C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0000
FLTEN2[5]	[15]	RW	Filter Enable for EXT_INT2[5] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH2[5]	[14:8]	RW	Filtering width of EXT_INT2[5]	0x00
FLTEN2[4]	[7]	RW	Filter Enable for EXT_INT2[4] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH2[4]	[6:0]	RW	Filtering width of EXT_INT2[4]	0x00

6.2.2.98 EXT_INT3_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0810, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN3[3]	[31]	RW	Filter Enable for EXT_INT3[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH3[3]	[30:24]	RW	Filtering width of EXT_INT3[3]	0x00
FLTEN3[2]	[23]	RW	Filter Enable for EXT_INT3[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH3[2]	[22:16]	RW	Filtering width of EXT_INT3[2]	0x00
FLTEN3[1]	[15]	RW	Filter Enable for EXT_INT3[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH3[1]	[14:8]	RW	Filtering width of EXT_INT3[1]	0x00
FLTEN3[0]	[7]	RW	Filter Enable for EXT_INT3[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH3[0]	[6:0]	RW	Filtering width of EXT_INT3[0]	0x00

6.2.2.99 EXT_INT3_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0814, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN3[7]	[31]	RW	Filter Enable for EXT_INT3[7] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH3[7]	[30:24]	RW	Filtering width of EXT_INT3[7]	0x00
FLTEN3[6]	[23]	RW	Filter Enable for EXT_INT3[6] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH3[6]	[22:16]	RW	Filtering width of EXT_INT3[6]	0x00
FLTEN3[5]	[15]	RW	Filter Enable for EXT_INT3[5] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH3[5]	[14:8]	RW	Filtering width of EXT_INT3[5]	0x00
FLTEN3[4]	[7]	RW	Filter Enable for EXT_INT3[4] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH3[4]	[6:0]	RW	Filtering width of EXT_INT3[4]	0x00

6.2.2.100 EXT_INT4_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0818, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN4[3]	[31]	RW	Filter Enable for EXT_INT4[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH4[3]	[30:24]	RW	Filtering width of EXT_INT4[3]	0x00
FLTEN4[2]	[23]	RW	Filter Enable for EXT_INT4[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH4[2]	[22:16]	RW	Filtering width of EXT_INT4[2]	0x00
FLTEN4[1]	[15]	RW	Filter Enable for EXT_INT4[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH4[1]	[14:8]	RW	Filtering width of EXT_INT4[1]	0x00
FLTEN4[0]	[7]	RW	Filter Enable for EXT_INT4[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH4[0]	[6:0]	RW	Filtering width of EXT_INT4[0]	0x00

6.2.2.101 EXT_INT4_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x081C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
FLTEN4[4]	[7]	RW	Filter Enable for EXT_INT4[4] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH4[4]	[6:0]	RW	Filtering width of EXT_INT4[4]	0x00

6.2.2.102 EXT_INT5_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0820, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN5[3]	[31]	RW	Filter Enable for EXT_INT5[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH5[3]	[30:24]	RW	Filtering width of EXT_INT5[3]	0x00
FLTEN5[2]	[23]	RW	Filter Enable for EXT_INT5[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH5[2]	[22:16]	RW	Filtering width of EXT_INT5[2]	0x00
FLTEN5[1]	[15]	RW	Filter Enable for EXT_INT5[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH5[1]	[14:8]	RW	Filtering width of EXT_INT5[1]	0x00
FLTEN5[0]	[7]	RW	Filter Enable for EXT_INT5[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH5[0]	[6:0]	RW	Filtering width of EXT_INT5[0]	0x00

6.2.2.103 EXT_INT5_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0824, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
FLTEN5[4]	[7]	RW	Filter Enable for EXT_INT5[4] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH5[4]	[6:0]	RW	Filtering width of EXT_INT5[4]	0x00

6.2.2.104 EXT_INT6_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0828, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN6[3]	[31]	RW	Filter Enable for EXT_INT6[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH6[3]	[30:24]	RW	Filtering width of EXT_INT6[3]	0x00
FLTEN6[2]	[23]	RW	Filter Enable for EXT_INT6[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH6[2]	[22:16]	RW	Filtering width of EXT_INT6[2]	0x00
FLTEN6[1]	[15]	RW	Filter Enable for EXT_INT6[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH6[1]	[14:8]	RW	Filtering width of EXT_INT6[1]	0x00
FLTEN6[0]	[7]	RW	Filter Enable for EXT_INT6[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH6[0]	[6:0]	RW	Filtering width of EXT_INT6[0]	0x00

6.2.2.105 EXT_INT6_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x082C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:0]	–	Reserved	0x00000000

6.2.2.106 EXT_INT7_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0830, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN7[3]	[31]	RW	Filter Enable for EXT_INT7[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH7[3]	[30:24]	RW	Filtering width of EXT_INT7[3]	0x00
FLTEN7[2]	[23]	RW	Filter Enable for EXT_INT7[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH7[2]	[22:16]	RW	Filtering width of EXT_INT7[2]	0x00
FLTEN7[1]	[15]	RW	Filter Enable for EXT_INT7[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH7[1]	[14:8]	RW	Filtering width of EXT_INT7[1]	0x00
FLTEN7[0]	[7]	RW	Filter Enable for EXT_INT7[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH7[0]	[6:0]	RW	Filtering width of EXT_INT7[0]	0x00

6.2.2.107 EXT_INT7_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0834, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:0]	–	Reserved	0x00000000

6.2.2.108 EXT_INT13_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0860, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN13[3]	[31]	RW	Filter Enable for EXT_INT13[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH13[3]	[30:24]	RW	Filtering width of EXT_INT13[3]	0x00
FLTEN13[2]	[23]	RW	Filter Enable for EXT_INT13[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH13[2]	[22:16]	RW	Filtering width of EXT_INT13[2]	0x00
FLTEN13[1]	[15]	RW	Filter Enable for EXT_INT13[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH13[1]	[14:8]	RW	Filtering width of EXT_INT13[1]	0x00
FLTEN13[0]	[7]	RW	Filter Enable for EXT_INT13[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH13[0]	[6:0]	RW	Filtering width of EXT_INT13[0]	0x00

6.2.2.109 EXT_INT13_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0864, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN13[7]	[31]	RW	Filter Enable for EXT_INT13[7] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH13[7]	[30:24]	RW	Filtering width of EXT_INT13[7]	0x00
FLTEN13[6]	[23]	RW	Filter Enable for EXT_INT13[6] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH13[6]	[22:16]	RW	Filtering width of EXT_INT13[6]	0x00
FLTEN13[5]	[15]	RW	Filter Enable for EXT_INT13[5] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH13[5]	[14:8]	RW	Filtering width of EXT_INT13[5]	0x00
FLTEN13[4]	[7]	RW	Filter Enable for EXT_INT13[4] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH13[4]	[6:0]	RW	Filtering width of EXT_INT13[4]	0x00

6.2.2.110 EXT_INT14_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0868, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN14[3]	[31]	RW	Filter Enable for EXT_INT14[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH14[3]	[30:24]	RW	Filtering width of EXT_INT14[3]	0x00
FLTEN14[2]	[23]	RW	Filter Enable for EXT_INT14[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH14[2]	[22:16]	RW	Filtering width of EXT_INT14[2]	0x00
FLTEN14[1]	[15]	RW	Filter Enable for EXT_INT14[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH14[1]	[14:8]	RW	Filtering width of EXT_INT14[1]	0x00
FLTEN14[0]	[7]	RW	Filter Enable for EXT_INT14[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH14[0]	[6:0]	RW	Filtering width of EXT_INT14[0]	0x00

6.2.2.111 EXT_INT14_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x086C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN14[7]	[31]	RW	Filter Enable for EXT_INT14[7] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH14[7]	[30:24]	RW	Filtering width of EXT_INT14[7]	0x00
FLTEN14[6]	[23]	RW	Filter Enable for EXT_INT14[6] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH14[6]	[22:16]	RW	Filtering width of EXT_INT14[6]	0x00
FLTEN14[5]	[15]	RW	Filter Enable for EXT_INT14[5] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH14[5]	[14:8]	RW	Filtering width of EXT_INT14[5]	0x00
FLTEN14[4]	[7]	RW	Filter Enable for EXT_INT14[4] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH14[4]	[6:0]	RW	Filtering width of EXT_INT14[4]	0x00

6.2.2.112 EXT_INT15_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0870, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN15[3]	[31]	RW	Filter Enable for EXT_INT15[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH15[3]	[30:24]	RW	Filtering width of EXT_INT15[3]	0x00
FLTEN15[2]	[23]	RW	Filter Enable for EXT_INT15[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH15[2]	[22:16]	RW	Filtering width of EXT_INT15[2]	0x00
FLTEN15[1]	[15]	RW	Filter Enable for EXT_INT15[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH15[1]	[14:8]	RW	Filtering width of EXT_INT15[1]	0x00
FLTEN15[0]	[7]	RW	Filter Enable for EXT_INT15[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH15[0]	[6:0]	RW	Filtering width of EXT_INT15[0]	0x00

6.2.2.113 EXT_INT15_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0874, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN15[7]	[31]	RW	Filter Enable for EXT_INT15[7] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH15[7]	[30:24]	RW	Filtering width of EXT_INT15[7]	0x00
FLTEN15[6]	[23]	RW	Filter Enable for EXT_INT15[6] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH15[6]	[22:16]	RW	Filtering width of EXT_INT15[6]	0x00
FLTEN15[5]	[15]	RW	Filter Enable for EXT_INT15[5] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH15[5]	[14:8]	RW	Filtering width of EXT_INT15[5]	0x00
FLTEN15[4]	[7]	RW	Filter Enable for EXT_INT15[4] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH15[4]	[6:0]	RW	Filtering width of EXT_INT15[4]	0x00

6.2.2.114 EXT_INT16_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0878, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN16[3]	[31]	RW	Filter Enable for EXT_INT16[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH16[3]	[30:24]	RW	Filtering width of EXT_INT16[3]	0x00
FLTEN16[2]	[23]	RW	Filter Enable for EXT_INT16[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH16[2]	[22:16]	RW	Filtering width of EXT_INT16[2]	0x00
FLTEN16[1]	[15]	RW	Filter Enable for EXT_INT16[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH16[1]	[14:8]	RW	Filtering width of EXT_INT16[1]	0x00
FLTEN16[0]	[7]	RW	Filter Enable for EXT_INT16[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH16[0]	[6:0]	RW	Filtering width of EXT_INT16[0]	0x00

6.2.2.115 EXT_INT16_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x087C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0000
FLTEN16[5]	[15]	RW	Filter Enable for EXT_INT16[5] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH16[5]	[14:8]	RW	Filtering width of EXT_INT16[5]	0x00
FLTEN16[4]	[7]	RW	Filter Enable for EXT_INT16[4] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH16[4]	[6:0]	RW	Filtering width of EXT_INT16[4]	0x00

6.2.2.116 EXT_INT21_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0880, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN17[3]	[31]	RW	Filter Enable for EXT_INT21[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH17[3]	[30:24]	RW	Filtering width of EXT_INT21[3]	0x00
FLTEN17[2]	[23]	RW	Filter Enable for EXT_INT21[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH17[2]	[22:16]	RW	Filtering width of EXT_INT21[2]	0x00
FLTEN17[1]	[15]	RW	Filter Enable for EXT_INT21[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH17[1]	[14:8]	RW	Filtering width of EXT_INT21[1]	0x00
FLTEN17[0]	[7]	RW	Filter Enable for EXT_INT21[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH17[0]	[6:0]	RW	Filtering width of EXT_INT21[0]	0x00

6.2.2.117 EXT_INT21_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0884, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN17[7]	[31]	RW	Filter Enable for EXT_INT21[7] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH17[7]	[30:24]	RW	Filtering width of EXT_INT21[7]	0x00
FLTEN17[6]	[23]	RW	Filter Enable for EXT_INT21[6] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH17[6]	[22:16]	RW	Filtering width of EXT_INT21[6]	0x00
FLTEN17[5]	[15]	RW	Filter Enable for EXT_INT21[5] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH17[5]	[14:8]	RW	Filtering width of EXT_INT21[5]	0x00
FLTEN17[4]	[7]	RW	Filter Enable for EXT_INT21[4] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH17[4]	[6:0]	RW	Filtering width of EXT_INT21[4]	0x00

6.2.2.118 EXT_INT22_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0888, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN18[3]	[31]	RW	Filter Enable for EXT_INT22[3] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH18[3]	[30:24]	RW	Filtering width of EXT_INT22[3]	0x00
FLTEN18[2]	[23]	RW	Filter Enable for EXT_INT22[2] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH18[2]	[22:16]	RW	Filtering width of EXT_INT22[2]	0x00
FLTEN18[1]	[15]	RW	Filter Enable for EXT_INT22[1] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH18[1]	[14:8]	RW	Filtering width of EXT_INT22[1]	0x00
FLTEN18[0]	[7]	RW	Filter Enable for EXT_INT22[0] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH18[0]	[6:0]	RW	Filtering width of EXT_INT22[0]	0x00

6.2.2.119 EXT_INT22_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x088C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
FLTEN18[4]	[7]	RW	Filter Enable for EXT_INT22[4] 0x0 = Disables filter 0x1 = Enables filter	0x0
FLTWIDTH18[4]	[6:0]	RW	Filtering width of EXT_INT22[4]	0x00

6.2.2.120 EXT_INT1_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0900, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT1_MASK[7]	[7]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT1_MASK[6]	[6]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT1_MASK[5]	[5]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT1_MASK[4]	[4]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT1_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT1_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT1_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT1_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.121 EXT_INT2_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0904, Reset Value = 0x0000_003F

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	0x0000000
EXT_INT2_MASK[5]	[5]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT2_MASK[4]	[4]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT2_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT2_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT2_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT2_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.122 EXT_INT3_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0908, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT3_MASK[7]	[7]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT3_MASK[6]	[6]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT3_MASK[5]	[5]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT3_MASK[4]	[4]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT3_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT3_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT3_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT3_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.123 EXT_INT4_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x090C, Reset Value = 0x0000_001F

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0000000
EXT_INT4_MASK[4]	[4]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT4_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT4_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT4_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT4_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.124 EXT_INT5_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0910, Reset Value = 0x0000_001F

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0000000
EXT_INT5_MASK[4]	[4]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT5_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT5_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT5_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT5_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.125 EXT_INT6_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0914, Reset Value = 0x0000_000F

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0000000
EXT_INT6_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT6_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT6_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT6_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.126 EXT_INT7_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0918, Reset Value = 0x0000_000F

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0000000
EXT_INT7_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT7_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT7_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT7_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.127 EXT_INT13_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0930, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT13_MASK[7]	[7]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT13_MASK[6]	[6]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT13_MASK[5]	[5]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT13_MASK[4]	[4]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT13_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT13_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT13_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT13_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.128 EXT_INT14_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0934, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT14_MASK[7]	[7]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT14_MASK[6]	[6]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT14_MASK[5]	[5]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT14_MASK[4]	[4]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT14_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT14_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT14_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT14_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.129 EXT_INT15_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0938, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT15_MASK[7]	[7]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT15_MASK[6]	[6]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT15_MASK[5]	[5]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT15_MASK[4]	[4]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT15_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT15_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT15_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT15_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.130 EXT_INT16_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x093C, Reset Value = 0x0000_003F

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	0x0000000
EXT_INT16_MASK[5]	[5]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT16_MASK[4]	[4]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT16_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT16_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT16_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT16_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.131 EXT_INT21_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0940, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT21_MASK[7]	[7]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT21_MASK[6]	[6]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT21_MASK[5]	[5]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT21_MASK[4]	[4]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT21_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT21_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT21_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT21_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.132 EXT_INT22_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0944, Reset Value = 0x0000_001F

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0000000
EXT_INT22_MASK[4]	[4]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT22_MASK[3]	[3]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT22_MASK[2]	[2]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT22_MASK[1]	[1]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1
EXT_INT22_MASK[0]	[0]	RW	0x0 = Enables interrupt 0x1 = Masked	0x1

6.2.2.133 EXT_INT1_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A00, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT1_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT1_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 =Interrupt occurs	0x0
EXT_INT1_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT1_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT1_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT1_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT1_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT1_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.134 EXT_INT2_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A04, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	0x0000000
EXT_INT2_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT2_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT2_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT2_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT2_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT2_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.135 EXT_INT3_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A08, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT3_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT3_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT3_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT3_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT3_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT3_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT3_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT3_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.136 EXT_INT4_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A0C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0000000
EXT_INT4_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT4_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT4_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT4_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT4_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.137 EXT_INT5_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A10, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0000000
EXT_INT5_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT5_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT5_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT5_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT5_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.138 EXT_INT6_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A14, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0000000
EXT_INT6_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT6_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT6_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT6_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.139 EXT_INT7_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A18, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0000000
EXT_INT7_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT7_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT7_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT7_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.140 EXT_INT13_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A30, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT13_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT13_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT13_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT13_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT13_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT13_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT13_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT13_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.141 EXT_INT14_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A34, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT14_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT14_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT14_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT14_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT14_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT14_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT14_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT14_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.142 EXT_INT15_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A38, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT15_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT15_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT15_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT15_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT15_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT15_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT15_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT15_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.143 EXT_INT16_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A3C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	0x0000000
EXT_INT16_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT16_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT16_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT16_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT16_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT16_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.144 EXT_INT21_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A40, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT21_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT21_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT21_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT21_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT21_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT21_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT21_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT21_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.145 EXT_INT22_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A44, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0000000
EXT_INT22_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT22_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT22_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT22_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT22_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.2.146 EXT_INT_SERVICE_XB

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B08, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	RW	Reserved	0x0000000
SVC_Group_Num	[7:3]	RW	EXT_INT Service group number 0x1 = EXT_INT1 0x2 = EXT_INT2 0x3 = EXT_INT3 0x4 = EXT_INT4 0x5 = EXT_INT5 0x6 = EXT_INT6 0x7 = EXT_INT7 0x8 = EXT_INT13 0x9 = EXT_INT14 0xA = EXT_INT15 0xB = EXT_INT16 0xC = EXT_INT21 0xD = EXT_INT22	0x00
SVC_Num	[2:0]	RW	Interrupt number to be serviced	0x0

6.2.2.147 EXT_INT_SERVICE_PEND_XB

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B0C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	RW	Reserved	0x0000000
SVC_PEND	[7:0]	RW	0x0 = Not occur 0x1 = Interrupt occurs	0x00

6.2.2.148 EXT_INT_GRPFIXPRI_XB

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B10, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0000000
Highest_GRP_NUM	[3:0]	RW	When fixed group priority mode = 0 to 12, then group number should be of the highest priority. 0x0 = EXT_INT1 0x1 = EXT_INT2 0x2 = EXT_INT3 0x3 = EXT_INT4 0x4 = EXT_INT5 0x5 = EXT_INT6 0x6 = EXT_INT7 0x7 = EXT_INT13 0x8 = EXT_INT14 0x9 = EXT_INT15 0xA = EXT_INT16 0xB = EXT_INT21 0xC = EXT_INT22	0x00

6.2.2.149 EXT_INT1_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B14, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 0 (EXT_INT1) when fixed priority mode: 0 to 7	0x0

6.2.2.150 EXT_INT2_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B18, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 1 (EXT_INT2) when fixed priority mode: 0 to 7	0x0

6.2.2.151 EXT_INT3_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B1C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 2 (EXT_INT3) when fixed priority mode: 0 to 7	0x0

6.2.2.152 EXT_INT4_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B20, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 3 (EXT_INT4) when fixed priority mode: 0 to 7	0x0

6.2.2.153 EXT_INT5_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B24, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 4 (EXT_INT5) when fixed priority mode: 0 to 7	0x0

6.2.2.154 EXT_INT6_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B28, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 5 (EXT_INT6) when fixed priority mode: 0 to 7	0x0

6.2.2.155 EXT_INT7_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B2C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 6 (EXT_INT7) when fixed priority mode: 0 to 7	0x0

6.2.2.156 EXT_INT13_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B44, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 7 (EXT_INT13) when fixed priority mode: 0 to 7	0x0

6.2.2.157 EXT_INT14_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B48, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 8 (EXT_INT14) when fixed priority mode: 0 to 7	0x0

6.2.2.158 EXT_INT15_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B4C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 9 (EXT_INT15) when fixed priority mode: 0 to 7	0x0

6.2.2.159 EXT_INT16_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B50, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 10 (EXT_INT16) when fixed priority mode: 0 to 7	0x0

6.2.2.160 EXT_INT21_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B54, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 11 (EXT_INT21) when fixed priority mode: 0 to 7	0x0

6.2.2.161 EXT_INT22_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B58, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 12 (EXT_INT22) when fixed priority mode: 0 to 7	0x0

6.2.2.162 PDNEN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0F80, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	–	Reserved	0x00
PDNEN_CFG	[1]	RW	0 = Automatically by power down mode 1 = by PDNEN bit	0x0
PDNEN	[0]	RW	Power down mode pad state enable register. 0 = PADs Controlled by normal mode 1 = PADs Controlled by Power Down mode control registers This bit is set to "1" automaticially when system enters into Power down mode and can be cleared by writing "0" to this bit or cold reset. After wake up from Power down mode, this bit maintains value "1" until writing "0"	0x0

6.2.3 Part 2

For the following SFRs, Sets the value does not take effect immediately. It takes at least 800 APB clocks for the value to take effect after the SFR is actually changed The SFRs are:

GPK0PUD, GPK0DRV, GPK1PUD, GPK1DRV, GPK2PUD, GPK2DRV, GPK3PUD, GPK3DRV, GPL0PUD, GPL0DRV, GPL1PUD, GPL1DRV, GPL2PUD, GPL2DRV, GPY0PUD, GPY0DRV, GPY1PUD, GPY1DRV, GPY2PUD, GPY2DRV, GPY3PUD, GPY3DRV, GPY4PUD, GPY4DRV, GPY5PUD, GPY5DRV, GPY6PUD, GPY6DRV, GPM0PUD, GPM0DRV, GPM1PUD, GPM1DRV, GPM2PUD, GPM2DRV, GPM3PUD, GPM3DRV, GPM4PUD, GPM4DRV.

6.2.3.1 GPK0CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPK0CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = SD_0_DATA[3] 0x3 = SD_4_DATA[3] 0x4 to 0xE = Reserved 0xF = EXT_INT23[6]	0x00
GPK0CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = SD_0_DATA[2] 0x3 = SD_4_DATA[2] 0x4 to 0xE = Reserved 0xF = EXT_INT23[5]	0x00
GPK0CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = SD_0_DATA[1] 0x3 = SD_4_DATA[1] 0x4 to 0xE = Reserved 0xF = EXT_INT23[4]	0x00
GPK0CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = SD_0_DATA[0] 0x3 = SD_4_DATA[0] 0x4 to 0xE = Reserved 0xF = EXT_INT23[3]	0x00
GPK0CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = SD_0_CDn 0x3 = SD_4_CDn 0x4 = GNSS_GPIO[8] 0x5 to 0xE = Reserved 0xF = EXT_INT23[2]	0x00
GPK0CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = SD_0_CMD 0x3 = SD_4_CMD 0x4 to 0xE = Reserved 0xF = EXT_INT23[1]	0x00
GPK0CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = SD_0_CLK 0x3 = SD_4_CLK 0x4 to 0xE = Reserved 0xF = EXT_INT23[0]	0x00

6.2.3.2 GPK0DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0044, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPK0DAT[6:0]	[6:0]	RWX	When you configure port as input port, then corresponding bit is pin state. When configuring as output port , the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.3 GPK0PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0048, Reset Value = 0x1555

Name	Bit	Type	Description	Reset Value
GPK0PUD[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x1555

6.2.3.4 GPK0DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x004C, Reset Value = 0x00_2AAA

Name	Bit	Type	Description	Reset Value
GPK0DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 6	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x2AAA

6.2.3.5 GPK0CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPK0[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.6 GPK0PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPK0[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.7 GPK1CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPK1CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = SD_1_DATA[3] 0x3 = SD_0_DATA[7] 0x4 = SD_4_DATA[7] 0x5 to 0xE = Reserved 0xF = EXT_INT24[6]	0x00
GPK1CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = SD_1_DATA[2] 0x3 = SD_0_DATA[6] 0x4 = SD_4_DATA[6] 0x5 to 0xE = Reserved 0xF = EXT_INT24[5]	0x00
GPK1CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = SD_1_DATA[1] 0x3 = SD_0_DATA[5] 0x4 = SD_4_DATA[5] 0x5 to 0xE = Reserved 0xF = EXT_INT24[4]	0x00
GPK1CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = SD_1_DATA[0] 0x3 = SD_0_DATA[4] 0x4 = SD_4_DATA[4] 0x5 to 0xE = Reserved 0xF = EXT_INT24[3]	0x00
GPK1CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = SD_1_CDn 0x3 = GNSS_GPIO[9] 0x4 = SD_4_nRESET_OUT 0x5 to 0xE = Reserved 0xF = EXT_INT24[2]	0x00
GPK1CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = SD_1_CMD 0x3 to 0xE = Reserved 0xF = EXT_INT24[1]	0x00
GPK1CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = SD_1_CLK 0x3 to 0xE = Reserved	0x00

Name	Bit	Type	Description	Reset Value
			0xF = EXT_INT24[0]	

6.2.3.8 GPK1DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0064, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPK1DAT[6:0]	[6:0]	RWX	When you configure port as input port, the corresponding bit is the pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.9 GPK1PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0068, Reset Value = 0x1555

Name	Bit	Type	Description	Reset Value
GPK1PUD[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x1555

6.2.3.10 GPK1DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x006C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPK1DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 6	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.11 GPK1CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0070, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPK1[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.12 GPK1PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0074, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPK1[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.13 GPK2CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPK2CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = SD_2_DATA[3] 0x3 to 0xE = Reserved 0xF = EXT_INT25[6]	0x00
GPK2CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = SD_2_DATA[2] 0x3 to 0xE = Reserved 0xF = EXT_INT25[5]	0x00
GPK2CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = SD_2_DATA[1] 0x3 to 0xE = Reserved 0xF = EXT_INT25[4]	0x00
GPK2CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = SD_2_DATA[0] 0x3 to 0xE = Reserved 0xF = EXT_INT25[3]	0x00
GPK2CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = SD_2_CDn 0x3 = GNSS_GPIO[10] 0x4 to 0xE = Reserved 0xF = EXT_INT25[2]	0x00
GPK2CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = SD_2_CMD 0x3 to 0xE = Reserved 0xF = EXT_INT25[1]	0x00
GPK2CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = SD_2_CLK 0x3 to 0xE = Reserved 0xF = EXT_INT25[0]	0x00

6.2.3.14 GPK2DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0084, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPK2DAT[6:0]	[6:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.15 GPK2PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0088, Reset Value = 0x1555

Name	Bit	Type	Description	Reset Value
GPK2PUD[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x1555

6.2.3.16 GPK2DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x008C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPK2DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 6	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.17 GPK2CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPK2[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.18 GPK2PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0094, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPK2[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Enables Reserved 0x3 = Pull-up	0x00

6.2.3.19 GPK3CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPK3CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = SD_3_DATA[3] 0x3 = SD_2_DATA[7] 0x4 to 0xE = Reserved 0xF = EXT_INT26[6]	0x00
GPK3CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = SD_3_DATA[2] 0x3 = SD_2_DATA[6] 0x4 to 0xE = Reserved 0xF = EXT_INT26[5]	0x00
GPK3CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = SD_3_DATA[1] 0x3 = SD_2_DATA[5] 0x4 to 0xE = Reserved 0xF = EXT_INT26[4]	0x00
GPK3CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = SD_3_DATA[0] 0x3 = SD_2_DATA[4] 0x4 to 0xE = Reserved 0xF = EXT_INT26[3]	0x00
GPK3CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = SD_3_CDn 0x3 = GNSS_GPIO[11] 0x4 to 0xE = Reserved 0xF = EXT_INT26[2]	0x00
GPK3CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = SD_3_CMD 0x3 to 0xE = Reserved 0xF = EXT_INT26[1]	0x00
GPK3CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = SD_3_CLK 0x3 to 0xE = Reserved 0xF = EXT_INT26[0]	0x00

6.2.3.20 GPK3DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x00A4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPK3DAT[6:0]	[6:0]	RWX	When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.21 GPK3PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x00A8, Reset Value = 0x1555

Name	Bit	Type	Description	Reset Value
GPK3PUD[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Disables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x1555

6.2.3.22 GPK3DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x00AC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPK3DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 6	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.23 GPK3CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x00B0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPK3[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.24 GPK3PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x00B4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPK3[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.25 GPL0CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPL0CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_RF_RSTN 0x3 to 0xE = Reserved 0xF = EXT_INT27[6]	0x00
GPL0CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 to 0xE = Reserved 0xF = EXT_INT27[5]	0x00
GPL0CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_QMAG 0x3 to 0xE = Reserved 0xF = EXT_INT27[4]	0x00
GPL0CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_QSIGN 0x3 to 0xE = Reserved 0xF = EXT_INT27[3]	0x00
GPL0CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_IMAG 0x3 to 0xE = Reserved 0xF = EXT_INT27[2]	0x00
GPL0CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_ISIGN 0x3 to 0xE = Reserved 0xF = EXT_INT27[1]	0x00
GPL0CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_SYNC 0x3 to 0xE = Reserved 0xF = EXT_INT27[0]	0x00

6.2.3.26 GPL0DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x00C4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPL0DAT[6:0]	[6:0]	RWX	When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.27 GPL0PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x00C8, Reset Value = 0x1555

Name	Bit	Type	Description	Reset Value
GPL0PUD[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Disables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x1555

6.2.3.28 GPL0DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x00CC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPL0DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 6	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.29 GPL0CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x00D0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPL0[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.30 GPL0PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x00D4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPL0[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.31 GPL1CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x00E0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPL1CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_SDA 0x3 to 0xE = Reserved 0xF = EXT_INT28[1]	0x00
GPL1CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_SCL 0x3 to 0xE = Reserved 0xF = EXT_INT28[0]	0x00

6.2.3.32 GPL1DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x00E4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPL1DAT[1:0]	[1:0]	RWX	When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.33 GPL1PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x00E8, Reset Value = 0x0005

Name	Bit	Type	Description	Reset Value
GPL1PUD[n]	[2n + 1:2n] n = 0 to 1	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Disables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0005

6.2.3.34 GPL1DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x00EC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPL1DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 1	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.35 GPL1CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x00F0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPL1[n]	[2n + 1:2n] n = 0 to 1	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.36 GPL1PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x00F4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPL1[n]	[2n + 1:2n] n = 0 to 1	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.37 GPL2CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0100, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPL2CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[7] 0x3 = KP_COL[7] 0x4 to 0xE = Reserved 0xF = EXT_INT29[7]	0x00
GPL2CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[6] 0x3 = KP_COL[6] 0x4 to 0xE = Reserved 0xF = EXT_INT29[6]	0x00
GPL2CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[5] 0x3 = KP_COL[5] 0x4 to 0xE = Reserved 0xF = EXT_INT29[5]	0x00
GPL2CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[4] 0x3 = KP_COL[4] 0x4 to 0xE = Reserved 0xF = EXT_INT29[4]	0x00
GPL2CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[3] 0x3 = KP_COL[3] 0x4 to 0xE = Reserved 0xF = EXT_INT29[3]	0x00
GPL2CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[2] 0x3 = KP_COL[2] 0x4 to 0xE = Reserved 0xF = EXT_INT29[2]	0x00
GPL2CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[1] 0x3 = KP_COL[1] 0x4 to 0xE = Reserved 0xF = EXT_INT29[1]	0x00
GPL2CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output	0x00

Name	Bit	Type	Description	Reset Value
			0x2 = GNSS_GPIO[0] 0x3 = KP_COL[0] 0x4 to 0xE = Reserved 0xF = EXT_INT29[0]	

6.2.3.38 GPL2DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0104, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPL2DAT[7:0]	[7:0]	RWX	When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.39 GPL2PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0108, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPL2PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Disables Pull-up	0x5555

6.2.3.40 GPL2DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x010C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPL2DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.41 GPL2CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0110, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPL2[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.42 GPL2PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0114, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPL2[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.43 GPY0CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0120, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPY0CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_WEn 0x4 to 0xF = Reserved	0x00
GPY0CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_OEn 0x4 to 0xF = Reserved	0x00
GPY0CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = SROM_CSn[3] 0x3 = NF_CSn[1] 0x4= Reserved 0x5 = OND_CSn[1] 0x4 to 0xF = Reserved	0x00
GPY0CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = SROM_CSn[2] 0x3 = NF_CSn[0] 0x4= Reserved 0x5 = OND_CSn[0] 0x4 to 0xF = Reserved	0x00
GPY0CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = SROM_CSn[1] 0x3 = NF_CSn[3] 0x4 to 0xF = Reserved	0x00
GPY0CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = SROM_CSn[0] 0x3 = NF_CSn[2] 0x4 to 0xF = Reserved	0x00

6.2.3.44 GPY0DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0124, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPY0DAT[5:0]	[5:0]	RWX	When you configure port as input port, the corresponding bit is the pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.45 GPY0PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0128, Reset Value = 0x0FFF

Name	Bit	Type	Description	Reset Value
GPY0PUD[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0FFF

6.2.3.46 GPY0DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x012C, Reset Value = 0x00_0AAA

Name	Bit	Type	Description	Reset Value
GPY0DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 5	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0AAA

6.2.3.47 GPY0CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0130, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY0[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.48 GPY0PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0134, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY0[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.49 GPY1CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0140, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPY1CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA_RDn 0x4 to 0xF = Reserved	0x00
GPY1CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = SROM_WAITn 0x4 to 0xF = Reserved	0x00
GPY1CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_BEn[1] 0x4 to 0xF = Reserved	0x00
GPY1CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_BEn[0] 0x4 to 0xF = Reserved	0x00

6.2.3.50 GPY1DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0144, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPY1DAT[3:0]	[3:0]	RWX	When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.51 GPY1PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0148, Reset Value = 0x00FF

Name	Bit	Type	Description	Reset Value
GPY1PUD[n]	[2n + 1:2n] n = 0 to 3	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Disables Pull-up	0x00FF

6.2.3.52 GPY1DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x014C, Reset Value = 0x00_00AA

Name	Bit	Type	Description	Reset Value
GPY1DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 3	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x00AA

6.2.3.53 GPY1CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0150, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY1[n]	[2n + 1:2n] n = 0 to 3	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.54 GPY1PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0154, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY1[n]	[2n + 1:2n] n = 0 to 3	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.55 GPY2CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0160, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPY2CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = NF_RnB[3] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY2CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = NF_RnB[2] 0x3= Reserved 0x4= Reserved 0x5 = OND_RPn 0x6 to 0xE = Reserved 0xF = –	0x00
GPY2CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = NF_RnB[1] 0x3= Reserved 0x4= Reserved 0x5 = OND_INT[1] 0x6 to 0xE = Reserved 0xF = –	0x00
GPY2CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = NF_RnB[0] 0x3= Reserved 0x4= Reserved 0x5 = OND_INT[0] 0x6 to 0xE = Reserved 0xF = –	0x00
GPY2CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = NF_ALE 0x3= Reserved 0x4= Reserved 0x5 = OND_SMCLK 0x6 to 0xE = Reserved 0xF = –	0x00
GPY2CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = NF_CLE 0x3= Reserved 0x4= Reserved 0x5 = OND_ADDRVALID 0x6 to 0xE = Reserved	0x00

Name	Bit	Type	Description	Reset Value
			0xF = –	

6.2.3.56 GPY2DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0164, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPY2DAT[5:0]	[5:0]	RWX	When you configure port as input port, the corresponding bit is the pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.57 GPY2PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0168, Reset Value = 0x0FFF

Name	Bit	Type	Description	Reset Value
GPY2PUD[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0 FFF

6.2.3.58 GPY2DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x016C, Reset Value = 0x00_0AAA

Name	Bit	Type	Description	Reset Value
GPY2DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 5	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0AAA

6.2.3.59 GPY2CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0170, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY2[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.60 GPY2PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0174, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY2[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.61 GPY3CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0180, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPY3CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[7] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY3CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output, 0x2 = EBI_ADDR[6] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY3CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[5] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY3CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[4] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY3CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[3] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY3CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[2] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY3CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[1] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY3CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[0] 0x3 to 0xE = Reserved 0xF = –	0x00

6.2.3.62 GPY3DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0184, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPY3DAT[7:0]	[7:0]	RWX	When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.63 GPY3PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0188, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPY3PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.3.64 GPY3DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x018C, Reset Value = 0x00_AAAA

Name	Bit	Type	Description	Reset Value
GPY3DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0xAAAA

6.2.3.65 GPY3CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0190, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY3[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.66 GPY3PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0194, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY3[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.67 GPY4CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x01A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPY4CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[15] 0x3= Reserved 0x4 = XhsiCAREADY 0x5 to 0xE = Reserved, 0xF = –	0x00
GPY4CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[14] 0x3= Reserved 0x4 = XhsiACFLAG 0x5 to 0xE = Reserved, 0xF = –	0x00
GPY4CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[13] 0x3= Reserved 0x4 = XhsiACDATA 0x5 to 0xE = Reserved 0xF = –	0x00
GPY4CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[12] 0x3= Reserved 0x4 = XhsiACWAKE 0x5 to 0xE = Reserved 0xF = –	0x00
GPY4CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[11] 0x3= Reserved 0x4 = XhsiACREADY 0x5 to 0xE = Reserved 0xF = –	0x00
GPY4CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[10] 0x3= Reserved 0x4 = XhsiCAFLAG 0x5 to 0xE = Reserved 0xF = –	0x00
GPY4CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output	0x00

Name	Bit	Type	Description	Reset Value
			0x2 = EBI_ADDR[9] 0x3= Reserved 0x4 = XhsiCADATA 0x5 to 0xE = Reserved 0xF = –	
GPY4CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[8] 0x3= Reserved 0x4 = XhsiCAWAKE 0x5 to 0xE = Reserved 0xF = –	0x00

6.2.3.68 GPY4DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x01A4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPY4DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.69 GPY4PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x01A8, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPY4PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.3.70 GPY4DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x01AC, Reset Value = 0x00_AAAA

Name	Bit	Type	Description	Reset Value
GPY4DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0xAAAA

6.2.3.71 GPY4CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x01B0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY4[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.72 GPY4PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x01B4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY4[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.73 GPY5CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x01C0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPY5CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[7] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY5CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[6] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY5CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[5] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY5CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[4] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY5CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[3] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY5CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[2] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY5CON[1]	[7:4]	RW	0x0 = Input]0x1 = Output 0x2 = EBI_DATA[1] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY5CON[0]	[3:0]	RW	0x0 = Input]0x1 = Output 0x2 = EBI_DATA[0] 0x3 to 0xE = Reserved 0xF = –	0x00

6.2.3.74 GPY5DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x01C4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPY5DAT[7:0]	[7:0]	RWX	When you configure port as input port, then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.75 GPY5PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x01C8, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPY5PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Disables Pull-up	0x5555

6.2.3.76 GPY5DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x01CC, Reset Value = 0x00_AAAA

Name	Bit	Type	Description	Reset Value
GPY5DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0xAAAA

6.2.3.77 GPY5CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x01D0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY5[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.78 GPY5PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x01D4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY5[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Disables Pull-up	0x00

6.2.3.79 GPY6CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x01E0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPY6CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[15] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY6CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[14] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY6CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[13] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY6CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[12] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY6CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[11] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY6CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[10] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY6CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[9] 0x3 to 0xE = Reserved 0xF = –	0x00
GPY6CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = EBI_DATA[8] 0x3 to 0xE = Reserved 0xF = –	0x00

6.2.3.80 GPY6DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x01E4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPY6DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.81 GPY6PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x01E8, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPY6PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.3.82 GPY6DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x01EC, Reset Value = 0x00_AAAA

Name	Bit	Type	Description	Reset Value
GPY6DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0xAAAA

6.2.3.83 GPY6CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x01F0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY6[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.84 GPY6PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x01F4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPY6[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.85 ETC0PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0208, Reset Value = 0x0400

Name	Bit	Type	Description	Reset Value
ETC0PUD[n]	[2n + 1:2n] n = 0 to 5	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0400

ETC0PUD[1:0] controls XjTRSTn.

ETC0PUD[3:2] controls XjTMS.

ETC0PUD[5:4] controls XjTCK.

ETC0PUD[7:6] controls XjTDI.

ETC0PUD[9:8] controls XjTDO.

ETC0PUD[11:10] controls XjDBGSEL.

6.2.3.86 ETC0DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x020C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
ETC0DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 5	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

ETC0DRV[1:0] controls XjTRSTn.

ETC0DRV[3:2] controls XjTMS.

ETC0DRV[5:4] controls XjTCK.

ETC0DRV[7:6] controls XjTDI.

ETC0DRV[9:8] controls XjTDO.

ETC0DRV[11:10] controls XjDBGSEL.

6.2.3.87 ETC6PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0228, Reset Value = 0xC000

Name	Bit	Type	Description	Reset Value
ETC6PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0xC000

ETC6PUD[1:0] controls XnRESET.

ETC6PUD[3:2] controls XCLKOUT.

ETC6PUD[5:4] controls XnRSTOUT.

ETC6PUD[9:8] controls XRTCCLKO.

ETC6PUD[11:10] controls XuotgDRVVBUS.

ETC6PUD[13:12] controls XuhostPWREN.

ETC6PUD[15:14] controls XuhostOVERCUR.

6.2.3.88 ETC6DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x022C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
ETC6DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

ETC6DRV[1:0] controls XnRESET.

ETC6DRV[3:2] controls XCLKOUT.

ETC6DRV[5:4] controls XnRSTOUT.

ETC6DRV[7:6] controls XnWRESET.

ETC6DRV[9:8] controls XRTCCLKO.

ETC6DRV[11:10] controls XuotgDRVVBUS.

ETC6DRV[13:12] controls XuhostPWREN.

ETC6DRV[15:14] controls XuhostOVERCUR.

6.2.3.89 GPM0CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0260, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPM0CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[6] 0x4 = XhsiCAFLAG 0x5 = TraceData[6] 0x6 to 0xE = Reserved 0xF = EXT_INT8[7]	0x00
GPM0CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[5] 0x4 = XhsiCADATA 0x5 = TraceData[5] 0x6 to 0xE = Reserved 0xF = EXT_INT8[6]	0x00
GPM0CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output, 0x2 = Reserved 0x3 = CAM_B_DATA[4], 0x4 = XhsiCAWAKE, 0x5 = TraceData[4], 0x6 to 0xE = Reserved, 0xF = EXT_INT8[5]	0x00
GPM0CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[3] 0x4 = TS_ERROR 0x5 = TraceData[3] 0x6 to 0xE = Reserved 0xF = EXT_INT8[4]	0x00
GPM0CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[2] 0x4 = TS_DATA 0x5 = TraceData[2] 0x6 to 0xE = Reserved 0xF = EXT_INT8[3]	0x00
GPM0CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[1]	0x00

Name	Bit	Type	Description	Reset Value
			0x4 = TS_VAL 0x5 = TraceData[1] 0x6 to 0xE = Reserved 0xF = EXT_INT8[2]	
GPM0CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[0] 0x4 = TS_SYNC 0x5 = TraceData[0] 0x6 to 0xE = Reserved 0xF = EXT_INT8[1]	0x00
GPM0CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_PCLK 0x4 = TS_CLK 0x5 = TraceClk 0x6 to 0xE = Reserved 0xF = EXT_INT8[0]	0x00

6.2.3.90 GPM0DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0264, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPM0DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.91 GPM0PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0268, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPM0PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.3.92 GPM0DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x026C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPM0DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.93 GPM0CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0270, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPM0[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.94 GPM0PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0274, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPM0[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.95 GPM1CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0280, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPM1CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_BAY_RGB[13] 0x3= Reserved 0x4= Reserved 0x5 = TraceData[12] 0x6 to 0xE = Reserved 0xF = EXT_INT9[6]	0x00
GPM1CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_BAY_RGB[12] 0x3= Reserved 0x4= Reserved 0x5 = TraceData[11] 0x6 to 0xE = Reserved 0xF = EXT_INT9[5]	0x00
GPM1CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_BAY_RGB[11] 0x3= Reserved 0x4 = XhsiCAREADY 0x5 = TraceData[10] 0x6 to 0xE = Reserved 0xF = EXT_INT9[4]	0x00
GPM1CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_BAY_RGB[10] 0x3= Reserved 0x4 = XhsiACFLAG 0x5 = TraceData[9] 0x6 to 0xE = Reserved 0xF = EXT_INT9[3]	0x00
GPM1CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_BAY_RGB[9] 0x3= Reserved 0x4 = XhsiACDATA 0x5 = TraceData[8] 0x6 to 0xE = Reserved 0xF = EXT_INT9[2]	0x00
GPM1CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_BAY_RGB[8] 0x3 = CAM_B_FIELD	0x00

Name	Bit	Type	Description	Reset Value
			0x4 = XhsiACWAKE 0x5 = TraceCtl 0x6 to 0xE = Reserved 0xF = EXT_INT9[1]	
GPM1CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[7] 0x4 = XhsiACREADY 0x5 = TraceData[7] 0x6 to 0xE = Reserved 0xF = EXT_INT9[0]	0x00

6.2.3.96 GPM1DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0284, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPM1DAT[6:0]	[6:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.97 GPM1PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0288, Reset Value = 0x1555

Name	Bit	Type	Description	Reset Value
GPM1PUD[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x1555

6.2.3.98 GPM1DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x028C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPM1DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 6	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.99 GPM1CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0290, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPM1[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.100 GPM1PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0294, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPM1[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.101 GPM2CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x02A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPM2CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[1] 0x3 = MPWM2_OUT_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT10[4]	0x00
GPM2CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[0] 0x3 = MPWM1_OUT_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT10[3]	0x00
GPM2CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_CLKOUT 0x4= Reserved 0x5 = TraceData[15] 0x6 to 0xE = Reserved 0xF = EXT_INT10[2]	0x00
GPM2CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_HREF 0x4= Reserved 0x5 = TraceData[14] 0x6 to 0xE = Reserved 0xF = EXT_INT10[1]	0x00
GPM2CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_VSYNC 0x4= Reserved 0x5 = TraceData[13] 0x6 to 0xE = Reserved 0xF = EXT_INT10[0]	0x00

6.2.3.102 GPM2DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x02A4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPM2DAT[4:0]	[4:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.103 GPM2PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x02A8, Reset Value = 0x0155

Name	Bit	Type	Description	Reset Value
GPM2PUD[n]	[2n + 1:2n] n = 0 to 4	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0155

6.2.3.104 GPM2DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x02AC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPM2DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 4	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.105 GPM2CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x02B0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPM2[n]	[2n + 1:2n] n = 0 to 4	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.106 GPM2PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x02B4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPM2[n]	[2n + 1:2n] n = 0 to 4	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.107 GPM3CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x02C0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPM3CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[9] 0x3 = RXD_UART_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT11[7]	0x00
GPM3CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[8] 0x3 = nCTS_UART_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT11[6]	0x00
GPM3CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[7] 0x3 = TXD_UART_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT11[5]	0x00
GPM3CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[6] 0x3 = nRTS_UART_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT11[4]	0x00
GPM3CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[5] 0x3 = MPWM6_OUT_ISP 0x4 = CAM_SPI1_MOSI 0x5 to 0xE = Reserved 0xF = EXT_INT11[3]	0x00
GPM3CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[4] 0x3 = MPWM5_OUT_ISP 0x4 = CAM_SPI1_MISO 0x5 to 0xE = Reserved 0xF = EXT_INT11[2]	0x00
GPM3CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[3] 0x3 = MPWM4_OUT_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT11[1]	0x00

Name	Bit	Type	Description	Reset Value
GPM3CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[2] 0x3 = MPWM3_OUT_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT11[0]	0x00

6.2.3.108 GPM3DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x02C4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPM3DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.109 GPM3PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x02C8, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPM3PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.3.110 GPM3DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x02CC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPM3DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.111 GPM3CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x02D0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPM3[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.112 GPM3PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x02D4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPM3[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.113 GPM4CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x02E0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPM4CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_SPI_MOSI 0x3 = CAM_GPIO[17] 0x4 to 0xE = Reserved 0xF = EXT_INT12[7]	0x00
GPM4CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_SPI_MISO 0x3 = CAM_GPIO[16] 0x4 to 0xE = Reserved 0xF = EXT_INT12[6]	0x00
GPM4CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_SPI_nSS 0x3 = CAM_GPIO[15] 0x4 to 0xE = Reserved 0xF = EXT_INT12[5]	0x00
GPM4CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_SPI_CLK 0x3 = CAM_GPIO[14] 0x4 to 0xE = Reserved 0xF = EXT_INT12[4]	0x00
GPM4CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_I2C1_SDA 0x3 = CAM_GPIO[13] 0x4 = CAM_SPI1_nSS 0x5 to 0xE = Reserved 0xF = EXT_INT12[3]	0x00
GPM4CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_I2C1_SCL 0x3 = CAM_GPIO[12] 0x4 = CAM_SPI1_CLK 0x5 to 0xE = Reserved 0xF = EXT_INT12[2]	0x00
GPM4CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_I2C0_SDA 0x3 = CAM_GPIO[11] 0x4 to 0xE = Reserved 0xF = EXT_INT12[1]	0x00

Name	Bit	Type	Description	Reset Value
GPM4CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = CAM_I2C0_SCL 0x3 = CAM_GPIO[10] 0x4 to 0xE = Reserved 0xF = EXT_INT12[0]	0x00

6.2.3.114 GPM4DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x02E4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPM4DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.115 GPM4PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x02E8, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPM4PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.3.116 GPM4DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x02EC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPM4DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.117 GPM4CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x02F0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPM4[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.3.118 GPM4PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x02F4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPM4[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.3.119 EXT_INT23CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0708, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT23_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT23[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT23_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT23[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT23_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT23[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT23_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT23[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT23_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT23[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7]	–	Reserved	0x0
EXT_INT23_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT23[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT23_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT23[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.120 EXT_INT24CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x070C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT24_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT24[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT24_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT24[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT24_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT24[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT24_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT24[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT24_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT24[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7]	–	Reserved	0x0
EXT_INT24_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT24[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT24_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT24[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.121 EXT_INT25CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0710, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT25_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT25[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT25_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT25[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT25_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT25[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT25_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT25[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT25_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT25[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7]	–	Reserved	0x0
EXT_INT25_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT25[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT25_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT25[0] 0x0 = Low level 0x1 = High level, 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.122 EXT_INT26CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0714, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT26_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT26[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT26_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT26[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT26_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT26[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT26_CON[3]	[14:12]	W	Sets signaling method of EXT_INT26[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT26_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT26[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7]	–	Reserved	0x0
EXT_INT26_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT26[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT26_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT26[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.123 EXT_INT27CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0718, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT27_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT27[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT27_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT27[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT27_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT27[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT27_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT27[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT27_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT27[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7]	–	Reserved	0x0
EXT_INT27_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT27[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT27_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT27[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.124 EXT_INT28CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x071C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
RSVD	[7]	–	Reserved	0x0
EXT_INT28_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT28[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT28_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT28[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.125 EXT_INT29CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0720, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT29_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT29[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT29_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT29[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT29_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT29[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT29_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT29[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT29_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT29[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT29_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT29[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT29_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT29[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT29_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT29[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.126 EXT_INT8CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0724, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT8_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT8[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT8_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT8[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT8_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT8[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT8_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT8[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT8_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT8[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT8_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT8[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT8_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT8[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT8_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT8[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.127 EXT_INT9CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0728, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT9_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT9[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT9_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT9[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT9_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT9[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT9_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT9[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT9_CON[2]	[10:8]	W	Sets signaling method of EXT_INT9[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7]	–	Reserved	0x0
EXT_INT9_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT9[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT9_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT9[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.128 EXT_INT10CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x072C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	0x000
RSVD	[19]	–	Reserved	0x0
EXT_INT10_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT10[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT10_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT10[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT10_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT10[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT10_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT10[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT10_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT10[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.129 EXT_INT11CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0730, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT11_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT11[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT11_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT11[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT11_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT11[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT11_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT11[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT11_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT11[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Rising edge triggered 0x4 = Both edge triggered 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT11_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT11[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT11_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT11[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT11_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT11[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.130 EXT_INT12CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0734, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT12_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT12[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT12_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT12[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT12_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT12[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT12_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT12[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT12_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT12[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT12_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT12[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT12_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT12[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT12_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT12[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.131 EXT_INT23_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0810, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN3[3]	[31]	RW	Filter Enable for EXT_INT23[3] 0x0 = Disables Filter 0x1 = Enabled Filter	0x0
FLTWIDTH3[3]	[30:24]	RW	Filtering width of EXT_INT23[3]	0x00
FLTEN3[2]	[23]	RW	Filter Enable for EXT_INT23[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[2]	[22:16]	RW	Filtering width of EXT_INT23[2]	0x00
FLTEN3[1]	[15]	RW	Filter Enable for EXT_INT23[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[1]	[14:8]	RW	Filtering width of EXT_INT23[1]	0x00
FLTEN3[0]	[7]	RW	Filter Enable for EXT_INT23[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[0]	[6:0]	RW	Filtering width of EXT_INT23[0]	0x00

6.2.3.132 EXT_INT23_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0814, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
FLTEN3[6]	[23]	RW	Filter Enable for EXT_INT23[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[6]	[22:16]	RW	Filtering width of EXT_INT23[6]	0x00
FLTEN3[5]	[15]	RW	Filter Enable for EXT_INT23[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[5]	[14:8]	RW	Filtering width of EXT_INT23[5]	0x00
FLTEN3[4]	[7]	RW	Filter Enable for EXT_INT23[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[4]	[6:0]	RW	Filtering width of EXT_INT23[4]	0x00

6.2.3.133 EXT_INT24_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0818, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN4[3]	[31]	RW	Filter Enable for EXT_INT24[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[3]	[30:24]	RW	Filtering width of EXT_INT24[3]	0x00
FLTEN4[2]	[23]	RW	Filter Enable for EXT_INT24[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[2]	[22:16]	RW	Filtering width of EXT_INT24[2]	0x00
FLTEN4[1]	[15]	RW	Filter Enable for EXT_INT24[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[1]	[14:8]	RW	Filtering width of EXT_INT24[1]	0x00
FLTEN4[0]	[7]	RW	Filter Enable for EXT_INT24[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[0]	[6:0]	RW	Filtering width of EXT_INT24[0]	0x00

6.2.3.134 EXT_INT24_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x081C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
FLTEN4[6]	[23]	RW	Filter Enable for EXT_INT24[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[6]	[22:16]	RW	Filtering width of EXT_INT24[6]	0x00
FLTEN4[5]	[15]	RW	Filter Enable for EXT_INT24[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[5]	[14:8]	RW	Filtering width of EXT_INT24[5]	0x00
FLTEN4[4]	[7]	RW	Filter Enable for EXT_INT24[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[4]	[6:0]	RW	Filtering width of EXT_INT24[4]	0x00

6.2.3.135 EXT_INT25_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0820, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN5[3]	[31]	RW	Filter Enable for EXT_INT25[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH5[3]	[30:24]	RW	Filtering width of EXT_INT25[3]	0x00
FLTEN5[2]	[23]	RW	Filter Enable for EXT_INT25[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH5[2]	[22:16]	RW	Filtering width of EXT_INT25[2]	0x00
FLTEN5[1]	[15]	RW	Filter Enable for EXT_INT25[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH5[1]	[14:8]	RW	Filtering width of EXT_INT25[1]	0x00
FLTEN5[0]	[7]	RW	Filter Enable for EXT_INT25[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH5[0]	[6:0]	RW	Filtering width of EXT_INT25[0]	0x00

6.2.3.136 EXT_INT25_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0824, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
FLTEN5[6]	[23]	RW	Filter Enable for EXT_INT25[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH5[6]	[22:16]	RW	Filtering width of EXT_INT25[6]	0x00
FLTEN5[5]	[15]	RW	Filter Enable for EXT_INT25[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH5[5]	[14:8]	RW	Filtering width of EXT_INT25[5]	0x00
FLTEN5[4]	[7]	RW	Filter Enable for EXT_INT25[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH5[4]	[6:0]	RW	Filtering width of EXT_INT25[4]	0x00

6.2.3.137 EXT_INT26_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0828, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN6[3]	[31]	RW	Filter Enable for EXT_INT26[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH6[3]	[30:24]	RW	Filtering width of EXT_INT26[3]	0x00
FLTEN6[2]	[23]	RW	Filter Enable for EXT_INT26[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH6[2]	[22:16]	RW	Filtering width of EXT_INT26[2]	0x00
FLTEN6[1]	[15]	RW	Filter Enable for EXT_INT26[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH6[1]	[14:8]	RW	Filtering width of EXT_INT26[1]	0x00
FLTEN6[0]	[7]	RW	Filter Enable for EXT_INT26[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH6[0]	[6:0]	RW	Filtering width of EXT_INT26[0]	0x00

6.2.3.138 EXT_INT26_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x082C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
FLTEN6[6]	[23]	RW	Filter Enable for EXT_INT26[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH6[6]	[22:16]	RW	Filtering width of EXT_INT26[6]	0x00
FLTEN6[5]	[15]	RW	Filter Enable for EXT_INT26[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH6[5]	[14:8]	RW	Filtering width of EXT_INT26[5]	0x00
FLTEN6[4]	[7]	RW	Filter Enable for EXT_INT26[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH6[4]	[6:0]	RW	Filtering width of EXT_INT26[4]	0x00

6.2.3.139 EXT_INT27_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0830, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN7[3]	[31]	RW	Filter Enable for EXT_INT27[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH7[3]	[30:24]	RW	Filtering width of EXT_INT27[3]	0x00
FLTEN7[2]	[23]	RW	Filter Enable for EXT_INT27[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH7[2]	[22:16]	RW	Filtering width of EXT_INT27[2]	0x00
FLTEN7[1]	[15]	RW	Filter Enable for EXT_INT27[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH7[1]	[14:8]	RW	Filtering width of EXT_INT27[1]	0x00
FLTEN7[0]	[7]	RW	Filter Enable for EXT_INT27[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH7[0]	[6:0]	RW	Filtering width of EXT_INT27[0]	0x00

6.2.3.140 EXT_INT27_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0834, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
FLTEN7[6]	[23]	RW	Filter Enable for EXT_INT27[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH7[6]	[22:16]	RW	Filtering width of EXT_INT27[6]	0x00
FLTEN7[5]	[15]	RW	Filter Enable for EXT_INT27[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH7[5]	[14:8]	RW	Filtering width of EXT_INT27[5]	0x00
FLTEN7[4]	[7]	RW	Filter Enable for EXT_INT27[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH7[4]	[6:0]	RW	Filtering width of EXT_INT27[4]	0x00

6.2.3.141 EXT_INT28_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0838, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
FLTEN8[1]	[15]	RW	Filter Enable for EXT_INT28[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH8[1]	[14:8]	RW	Filtering width of EXT_INT28[1]	0x00
FLTEN8[0]	[7]	RW	Filter Enable for EXT_INT28[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH8[0]	[6:0]	RW	Filtering width of EXT_INT28[0]	0x00

6.2.3.142 EXT_INT28_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x083C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:0]	–	Reserved	0x00000000

6.2.3.143 EXT_INT29_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0840, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN9[3]	[31]	RW	Filter Enable for EXT_INT29[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH9[3]	[30:24]	RW	Filtering width of EXT_INT29[3]	0x00
FLTEN9[2]	[23]	RW	Filter Enable for EXT_INT29[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH9[2]	[22:16]	RW	Filtering width of EXT_INT29[2]	0x00
FLTEN9[1]	[15]	RW	Filter Enable for EXT_INT29[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH9[1]	[14:8]	RW	Filtering width of EXT_INT29[1]	0x00
FLTEN9[0]	[7]	RW	Filter Enable for EXT_INT29[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH9[0]	[6:0]	RW	Filtering width of EXT_INT29[0]	0x00

6.2.3.144 EXT_INT29_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0844, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN9[7]	[31]	RW	Filter Enable for EXT_INT29[7] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH9[7]	[30:24]	RW	Filtering width of EXT_INT29[7]	0x00
FLTEN9[6]	[23]	RW	Filter Enable for EXT_INT29[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH9[6]	[22:16]	RW	Filtering width of EXT_INT29[6]	0x00
FLTEN9[5]	[15]	RW	Filter Enable for EXT_INT29[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH9[5]	[14:8]	RW	Filtering width of EXT_INT29[5]	0x00
FLTEN9[4]	[7]	RW	Filter Enable for EXT_INT29[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH9[4]	[6:0]	RW	Filtering width of EXT_INT29[4]	0x00

6.2.3.145 EXT_INT8_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0848, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN10[3]	[31]	RW	Filter Enable for EXT_INT8[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH10[3]	[30:24]	RW	Filtering width of EXT_INT8[3]	0x00
FLTEN10[2]	[23]	RW	Filter Enable for EXT_INT8[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH10[2]	[22:16]	RW	Filtering width of EXT_INT8[2]	0x00
FLTEN10[1]	[15]	RW	Filter Enable for EXT_INT8[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH10[1]	[14:8]	RW	Filtering width of EXT_INT8[1]	0x00
FLTEN10[0]	[7]	RW	Filter Enable for EXT_INT8[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH10[0]	[6:0]	RW	Filtering width of EXT_INT8[0]	0x00

6.2.3.146 EXT_INT8_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x084C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN10[7]	[31]	RW	Filter Enable for EXT_INT8[7] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH10[7]	[30:24]	RW	Filtering width of EXT_INT8[7]	0x00
FLTEN10[6]	[23]	RW	Filter Enable for EXT_INT8[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH10[6]	[22:16]	RW	Filtering width of EXT_INT8[6]	0x00
FLTEN10[5]	[15]	RW	Filter Enable for EXT_INT8[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH10[5]	[14:8]	RW	Filtering width of EXT_INT8[5]	0x00
FLTEN10[4]	[7]	RW	Filter Enable for EXT_INT8[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH10[4]	[6:0]	RW	Filtering width of EXT_INT8[4]	0x00

6.2.3.147 EXT_INT9_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0850, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN11[3]	[31]	RW	Filter Enable for EXT_INT9[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH11[3]	[30:24]	RW	Filtering width of EXT_INT9[3]	0x00
FLTEN11[2]	[23]	RW	Filter Enable for EXT_INT9[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH11[2]	[22:16]	RW	Filtering width of EXT_INT9[2]	0x00
FLTEN11[1]	[15]	RW	Filter Enable for EXT_INT9[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH11[1]	[14:8]	RW	Filtering width of EXT_INT9[1]	0x00
FLTEN11[0]	[7]	RW	Filter Enable for EXT_INT9[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH11[0]	[6:0]	RW	Filtering width of EXT_INT9[0]	0x00

6.2.3.148 EXT_INT9_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0854, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
FLTEN11[6]	[23]	RW	Filter Enable for EXT_INT9[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH11[6]	[22:16]	RW	Filtering width of EXT_INT9[6]	0x00
FLTEN11[5]	[15]	RW	Filter Enable for EXT_INT9[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH11[5]	[14:8]	RW	Filtering width of EXT_INT9[5]	0x00
FLTEN11[4]	[7]	RW	Filter Enable for EXT_INT9[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH11[4]	[6:0]	RW	Filtering width of EXT_INT9[4]	0x00

6.2.3.149 EXT_INT10_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0858, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN12[3]	[31]	RW	Filter Enable for EXT_INT10[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH12[3]	[30:24]	RW	Filtering width of EXT_INT10[3]	0x00
FLTEN12[2]	[23]	RW	Filter Enable for EXT_INT10[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH12[2]	[22:16]	RW	Filtering width of EXT_INT10[2]	0x00
FLTEN12[1]	[15]	RW	Filter Enable for EXT_INT10[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH12[1]	[14:8]	RW	Filtering width of EXT_INT10[1]	0x00
FLTEN12[0]	[7]	RW	Filter Enable for EXT_INT10[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH12[0]	[6:0]	RW	Filtering width of EXT_INT10[0]	0x00

6.2.3.150 EXT_INT10_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x085C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
FLTEN12[4]	[7]	RW	Filter Enable for EXT_INT10[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH12[4]	[6:0]	RW	Filtering width of EXT_INT10[4]	0x00

6.2.3.151 EXT_INT11_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0860, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN13[3]	[31]	RW	Filter Enable for EXT_INT11[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH13[3]	[30:24]	RW	Filtering width of EXT_INT11[3]	0x00
FLTEN13[2]	[23]	RW	Filter Enable for EXT_INT11[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH13[2]	[22:16]	RW	Filtering width of EXT_INT11[2]	0x00
FLTEN13[1]	[15]	RW	Filter Enable for EXT_INT11[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH13[1]	[14:8]	RW	Filtering width of EXT_INT11[1]	0x00
FLTEN13[0]	[7]	RW	Filter Enable for EXT_INT11[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH13[0]	[6:0]	RW	Filtering width of EXT_INT11[0]	0x00

6.2.3.152 EXT_INT11_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0864, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN13[7]	[31]	RW	Filter Enable For EXT_INT11[7] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH13[7]	[30:24]	RW	Filtering width of EXT_INT11[7]	0x00
FLTEN13[6]	[23]	RW	Filter Enable for EXT_INT11[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH13[6]	[22:16]	RW	Filtering width of EXT_INT11[6]	0x00
FLTEN13[5]	[15]	RW	Filter Enable for EXT_INT11[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH13[5]	[14:8]	RW	Filtering width of EXT_INT11[5]	0x00
FLTEN13[4]	[7]	RW	Filter Enable for EXT_INT11[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH13[4]	[6:0]	RW	Filtering width of EXT_INT11[4]	0x00

6.2.3.153 EXT_INT12_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0868, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN14[3]	[31]	RW	Filter Enable for EXT_INT12[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH14[3]	[30:24]	RW	Filtering width of EXT_INT12[3]	0x00
FLTEN14[2]	[23]	RW	Filter Enable for EXT_INT12[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH14[2]	[22:16]	RW	Filtering width of EXT_INT12[2]	0x00
FLTEN14[1]	[15]	RW	Filter Enable for EXT_INT12[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH14[1]	[14:8]	RW	Filtering width of EXT_INT12[1]	0x00
FLTEN14[0]	[7]	RW	Filter Enable for EXT_INT12[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH14[0]	[6:0]	RW	Filtering width of EXT_INT12[0]	0x00

6.2.3.154 EXT_INT12_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x086C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN14[7]	[31]	RW	Filter Enable for EXT_INT12[7] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH14[7]	[30:24]	RW	Filtering width of EXT_INT12[7]	0x00
FLTEN14[6]	[23]	RW	Filter Enable for EXT_INT12[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH14[6]	[22:16]	RW	Filtering width of EXT_INT12[6]	0x00
FLTEN14[5]	[15]	RW	Filter Enable for EXT_INT12[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH14[5]	[14:8]	RW	Filtering width of EXT_INT12[5]	0x00
FLTEN14[4]	[7]	RW	Filter Enable for EXT_INT12[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH14[4]	[6:0]	RW	Filtering width of EXT_INT12[4]	0x00

6.2.3.155 EXT_INT23_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0908, Reset Value = 0x0000_007F

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT23_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT23_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT23_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT23_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT23_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT23_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT23_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.156 EXT_INT24_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x090C, Reset Value = 0x0000_007F

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT24_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT24_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT24_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT24_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT24_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT24_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT24_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.157 EXT_INT25_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0910, Reset Value = 0x0000_007F

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT25_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT25_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT25_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT25_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT25_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT25_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT25_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.158 EXT_INT26_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0914, Reset Value = 0x0000_007F

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT26_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT26_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT26_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT26_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT26_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT26_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT26_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.159 EXT_INT27_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0918, Reset Value = 0x0000_007F

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT27_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT27_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT27_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT27_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT27_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT27_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT27_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.160 EXT_INT28_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x091C, Reset Value = 0x0000_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x00000000
EXT_INT28_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT28_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.161 EXT_INT29_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0920, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT29_MASK[7]	[7]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT29_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT29_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT29_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT29_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT29_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT29_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT29_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.162 EXT_INT8_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0924, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT8_MASK[7]	[7]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT8_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT8_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT8_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT8_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT8_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT8_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT8_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.163 EXT_INT9_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0928, Reset Value = 0x0000_007F

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT9_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT9_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT9_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT9_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT9_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT9_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT9_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.164 EXT_INT10_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x092C, Reset Value = 0x0000_001F

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0000000
EXT_INT10_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT10_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT10_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT10_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT10_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.165 EXT_INT11_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0930, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT11_MASK[7]	[7]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT11_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT11_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT11_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT11_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT11_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT11_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT11_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.166 EXT_INT12_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0934, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT12_MASK[7]	[7]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT12_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT12_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT12_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT12_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT12_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT12_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT12_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.167 EXT_INT23_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A08, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT23_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 =Interrupt occurs	0x0
EXT_INT23_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT23_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT23_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT23_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT23_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT23_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.3.168 EXT_INT24_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A0C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT24_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT24_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT24_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT24_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT24_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT24_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = interrupt occurs	0x0
EXT_INT24_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.3.169 EXT_INT25_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A10, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT25_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT25_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT25_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT25_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT25_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT25_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT25_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.3.170 EXT_INT26_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A14, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT26_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT26_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT26_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT26_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT26_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT26_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = interrupt occurs	0x0
EXT_INT26_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.3.171 EXT_INT27_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A18, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT27_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT27_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = interrupt occurs	0x0
EXT_INT27_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = interrupt occurs	0x0
EXT_INT27_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = interrupt occurs	0x0
EXT_INT27_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT27_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = interrupt occurs	0x0
EXT_INT27_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.3.172 EXT_INT28_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A1C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x00000000
EXT_INT28_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT28_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = interrupt occurs	0x0

6.2.3.173 EXT_INT29_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A20, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT29_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = interrupt occurs	0x0
EXT_INT29_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = interrupt occurs	0x0
EXT_INT29_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT29_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = interrupt occurs	0x0
EXT_INT29_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT29_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT29_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT29_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.3.174 EXT_INT8_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A24, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT8_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT8_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT8_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = interrupt occurs	0x0
EXT_INT8_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT8_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT8_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT8_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT8_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.3.175 EXT_INT9_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A28, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT9_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT9_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT9_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT9_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT9_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT9_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT9_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.3.176 EXT_INT10_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A2C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0000000
EXT_INT10_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT10_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT10_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT10_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT10_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.3.177 EXT_INT11_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A30, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT11_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT11_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT11_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT11_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT11_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT11_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT11_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT11_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.3.178 EXT_INT12_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A34, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT12_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT12_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT12_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT12_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT12_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT12_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT12_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0
EXT_INT12_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt occurs	0x0

6.2.3.179 EXT_INT_SERVICE_XA

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B08, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	RW	Reserved	0x0000000
SVC_Group_Num	[7:3]	RW	EXT_INT Service group number 0x1 = EXT_INT23 0x2 = EXT_INT24 0x3 = EXT_INT25 0x4 = EXT_INT26 0x5 = EXT_INT27 0x6 = EXT_INT28 0x7 = EXT_INT29 0x8 = EXT_INT8 0x9 = EXT_INT9 0xA = EXT_INT10 0xB = EXT_INT11 0xC = EXT_INT12	0x00
SVC_Num	[2:0]	RW	Interrupt number to be serviced	0x0

6.2.3.180 EXT_INT_SERVICE_PEND_XA

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B0C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	RW	Reserved	0x0000000
SVC_PEND	[7:0]	RW	0x0 = Not occur 0x1 = Interrupt occurs	0x00

6.2.3.181 EXT_INT_GRPFIXPRI_XA

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B10, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0000000
Highest_GRP_NUM	[3:0]	RW	When fixed group priority mode = 0 to 11, then group number should be of the highest priority. 0x0 = EXT_INT23 0x1 = EXT_INT24 0x2 = EXT_INT25 0x3 = EXT_INT26 0x4 = EXT_INT27 0x5 = EXT_INT28 0x6 = EXT_INT29 0x7 = EXT_INT8 0x8 = EXT_INT9 0x9 = EXT_INT10 0xA = EXT_INT11 0xB = EXT_INT12	0x00

6.2.3.182 EXT_INT23_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B1C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 0 (EXT_INT23) when fixed priority mode: 0 to 7	0x0

6.2.3.183 EXT_INT24_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B20, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 1 (EXT_INT24) when fixed priority mode: 0 to 7	0x0

6.2.3.184 EXT_INT25_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B24, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 2 (EXT_INT25) when fixed priority mode: 0 to 7	0x0

6.2.3.185 EXT_INT26_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B28, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 3 (EXT_INT26) when fixed priority mode: 0 to 7	0x0

6.2.3.186 EXT_INT27_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B2C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 4 (EXT_INT27) when fixed priority mode: 0 to 7	0x0

6.2.3.187 EXT_INT28_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B30, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 5 (EXT_INT28) when fixed priority mode: 0 to 7	0x0

6.2.3.188 EXT_INT29_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B34, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 6 (EXT_INT29) when fixed priority mode: 0 to 7	0x0

6.2.3.189 EXT_INT8_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B38, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 7 (EXT_INT8) when fixed priority mode: 0 to 7	0x0

6.2.3.190 EXT_INT9_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B3C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 8 (EXT_INT9) when fixed priority mode: 0 to 7	0x0

6.2.3.191 EXT_INT10_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B40, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 9 (EXT_INT10) when fixed priority mode: 0 to 7	0x0

6.2.3.192 EXT_INT11_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B44, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 10 (EXT_INT11) when fixed priority mode: 0 to 7	0x0

6.2.3.193 EXT_INT12_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B48, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 11 (EXT_INT12) when fixed priority mode: 0 to 7	0x0

6.2.3.194 GPX0CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C00, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPX0CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = Reserved 0x4 = Reserved 0x5 = ALV_DBG[3] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[7]	0x00
GPX0CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = Reserved 0x4 = Reserved 0x5 = ALV_DBG[2] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[6]	0x00
GPX0CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = Reserved 0x4 = Reserved 0x5 = ALV_DBG[1] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[5]	0x00
GPX0CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = AUD_TRSTn 0x4 = GNSS_TRSTn 0x5 = ALV_DBG[0] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[4]	0x00
GPX0CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = AUD_TDO 0x4 = GNSS_TDO 0x5 = ALV_TDO 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[3]	0x00
GPX0CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = AUD_TDI	0x00

Name	Bit	Type	Description	Reset Value
			0x4 = GNSS_TDI 0x5 = ALV_TDI 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[2]	
GPX0CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = AUD_TMS 0x4 = GNSS_TMS 0x5 = ALV_TMS 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[1]	0x00
GPX0CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = AUD_TCK 0x4 = GNSS_TCK 0x5 = ALV_TCK 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[0]	0x00

6.2.3.195 GPX0DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C04, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPX0DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.196 GPX0PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C08, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPX0PUD[n]	[2n + 1:2n] N = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.3.197 GPX0DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C0C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPX0DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] N = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.198 GPX1CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C20, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPX1CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[7] 0x4 = Reserved 0x5 = ALV_DBG[11] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[7]	0x00
GPX1CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[6] 0x4 = Reserved 0x5 = ALV_DBG[10] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[6]	0x00
GPX1CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[5] 0x4 = Reserved 0x5 = ALV_DBG[9] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[5]	0x00
GPX1CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[4] 0x4 = Reserved 0x5 = ALV_DBG[8] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[4]	0x00
GPX1CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[3] 0x4 = Reserved 0x5 = ALV_DBG[7] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[3]	0x00
GPX1CON[2]	[11:8]	RW	0x0 = Input, 0x1 = Output, 0x2 = Reserved 0x3 = KP_COL[2]	0x00

Name	Bit	Type	Description	Reset Value
			0x4 = Reserved 0x5 = ALV_DBG[6] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[2]	
GPX1CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[1] 0x4 = Reserved 0x5 = ALV_DBG[5] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[1]	0x00
GPX1CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[0] 0x4 = Reserved 0x5 = ALV_DBG[4] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[0]	0x00

6.2.3.199 GPX1DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C24, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPX1DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.200 GPX1PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C28, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPX1PUD[n]	[2n + 1:2n] N = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.3.201 GPX1DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C2C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPX1DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] N = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.202 GPX2CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C40, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPX2CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[7] 0x4 = Reserved 0x5 = ALV_DBG[19] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[7]	0x00
GPX2CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[6] 0x4 = Reserved 0x5 = ALV_DBG[18] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[6]	0x00
GPX2CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[5] 0x4 = Reserved 0x5 = ALV_DBG[17] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[5]	0x00
GPX2CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[4] 0x4 = Reserved 0x5 = ALV_DBG[16] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[4]	0x00
GPX2CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[3] 0x4 = Reserved 0x5 = ALV_DBG[15] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[3]	0x00
GPX2CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[2]	0x00

Name	Bit	Type	Description	Reset Value
			0x4 = Reserved 0x5 = ALV_DBG[14] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[2]	
GPX2CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[1] 0x4 = Reserved 0x5 = ALV_DBG[13] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[1]	0x00
GPX2CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[0] 0x4 = Reserved 0x5 = ALV_DBG[12] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[0]	0x00

6.2.3.203 GPX2DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C44, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPX2DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.204 GPX2PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C48, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPX2PUD[n]	[2n + 1:2n] N = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.3.205 GPX2DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C4C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPX2DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] N = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.206 GPX3CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C60, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPX3CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = HDMI_HPD 0x4 = Reserved 0x5 = ALV_DBG[27] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[7]	0x00
GPX3CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = HDMI_CEC 0x4 = Reserved 0x5 = ALV_DBG[26] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[6]	0x00
GPX3CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[13] 0x4 = Reserved 0x5 = ALV_DBG[25] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[5]	0x00
GPX3CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[12] 0x4 = Reserved 0x5 = ALV_DBG[24] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[4]	0x00
GPX3CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[11] 0x4 = Reserved 0x5 = ALV_DBG[23] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[3]	0x00
GPX3CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[10]	0x00

Name	Bit	Type	Description	Reset Value
			0x4 = Reserved 0x5 = ALV_DBG[22] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[2]	
GPX3CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[9] 0x4 = Reserved 0x5 = ALV_DBG[21] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[1]	0x00
GPX3CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[8] 0x4 = Reserved 0x5 = ALV_DBG[20] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[0]	0x00

6.2.3.207 GPX3DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C64, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPX3DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.3.208 GPX3PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C68, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPX3PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.3.209 GPX3DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C6C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPX3DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.3.210 EXT_INT40CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E00, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT40_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT40[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT40_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT40[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT40_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT40[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT40_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT40[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT40_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT40[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT40_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT40[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT40_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT40[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT40_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT40[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.211 EXT_INT41CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E04, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT41_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT41[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT41_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT41[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT41_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT41[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT41_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT41[4] 0x0 = Low level 0x1 = High level 0x2 = Falling edge triggered 0x3 = Rising edge triggered 0x4 = Both edge triggered 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT41_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT41[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT41_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT41[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT41_CON[1]	[6:4]	W	Sets signaling method of EXT_INT41[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT41_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT41[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.212 EXT_INT42CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E08, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT42_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT42[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT42_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT42[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT42_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT42[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT42_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT42[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT42_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT42[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT42_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT42[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT42_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT42[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT42_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT42[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.213 EXT_INT43CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E0C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT43_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT43[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT43_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT43[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT43_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT43[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers ising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT43_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT43[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT43_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT43[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT43_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT43[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT43_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT43[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT43_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT43[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

6.2.3.214 EXT_INT40_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E80, Reset Value = 0x8080_8080

Name	Bit	Type	Description	Reset Value
FLTEN15[3]	[31]	RW	Filter Enable for EXT_INT40[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL15[3]	[30]	RW	Filter Selection for EXT_INT40[3] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH15[3]	[29:24]	RW	Filtering width of EXT_INT40[3] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1.	0x00
FLTEN15[2]	[23]	RW	Filter Enable for EXT_INT40[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL15[2]	[22]	RW	Filter Selection for EXT_INT40[2] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH15[2]	[21:16]	RW	Filtering width of EXT_INT40[2] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1.	0x00
FLTEN15[1]	[15]	RW	Filter Enable for EXT_INT40[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL15[1]	[14]	RW	Filter Selection for EXT_INT40[1] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH15[1]	[13:8]	RW	Filtering width of EXT_INT40[1] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1.	0x00
FLTEN15[0]	[7]	RW	Filter Enable for EXT_INT40[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL15[0]	[6]	RW	Filter Selection for EXT_INT40[0] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH15[0]	[5:0]	RW	Filtering width of EXT_INT40[0] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1.	0x00

6.2.3.215 EXT_INT40_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E84, Reset Value = 0x8080_8080

Name	Bit	Type	Description	Reset Value
FLTEN15[7]	[31]	RW	Filter Enable for EXT_INT40[7] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL15[7]	[30]	RW	Filter Selection for EXT_INT40[7] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH15[7]	[29:24]	RW	Filtering width of EXT_INT40[7] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1.	0x00
FLTEN15[6]	[23]	RW	Filter Enable for EXT_INT40[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL15[6]	[22]	RW	Filter Selection for EXT_INT40[6] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH15[6]	[21:16]	RW	Filtering width of EXT_INT40[6] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1.	0x00
FLTEN15[5]	[15]	RW	Filter Enable for EXT_INT40[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL15[5]	[14]	RW	Filter Selection for EXT_INT40[5] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH15[5]	[13:8]	RW	Filtering width of EXT_INT40[5] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1.	0x00
FLTEN15[4]	[7]	RW	Filter Enable for EXT_INT40[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL15[4]	[6]	RW	Filter Selection for EXT_INT40[4] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH15[4]	[5:0]	RW	Filtering width of EXT_INT40[4] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1.	0x00

6.2.3.216 EXT_INT41_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E88, Reset Value = 0x8080_8080

Name	Bit	Type	Description	Reset Value
FLTEN16[3]	[31]	RW	Filter Enable for EXT_INT41[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL16[3]	[30]	RW	Filter Selection for EXT_INT41[3] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH16[3]	[29:24]	RW	Filtering width of EXT_INT41[3] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1.	0x00
FLTEN16[2]	[23]	RW	Filter Enable for EXT_INT41[2] 0x0 = Disables Filter 0x1 = Enables	0x1
FLTSEL16[2]	[22]	RW	Filter Selection for EXT_INT41[2] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH16[2]	[21:16]	RW	Filtering width of EXT_INT41[2] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1.	0x00
FLTEN16[1]	[15]	RW	Filter Enable for EXT_INT41[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL16[1]	[14]	RW	Filter Selection for EXT_INT41[1] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH16[1]	[13:8]	RW	Filtering width of EXT_INT41[1] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1.	0x00
FLTEN16[0]	[7]	RW	Filter Enable for EXT_INT41[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL16[0]	[6]	RW	Filter Selection for EXT_INT41[0] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH16[0]	[5:0]	RW	Filtering width of EXT_INT41[0] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1.	0x00

6.2.3.217 EXT_INT41_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E8C, Reset Value = 0x8080_8080

Name	Bit	Type	Description	Reset Value
FLTEN16[7]	[31]	RW	Filter Enable for EXT_INT41[7] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL16[7]	[30]	RW	Filter Selection for EXT_INT41[7] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH16[7]	[29:24]	RW	Filtering width of EXT_INT41[7] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1.	0x00
FLTEN16[6]	[23]	RW	Filter Enable for EXT_INT41[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL16[6]	[22]	RW	Filter Selection for EXT_INT41[6] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH16[6]	[21:16]	RW	Filtering width of EXT_INT41[6] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1.	0x00
FLTEN16[5]	[15]	RW	Filter Enable for EXT_INT41[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL16[5]	[14]	RW	Filter Selection for EXT_INT41[5] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH16[5]	[13:8]	RW	Filtering width of EXT_INT41[5] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1.	0x00
FLTEN16[4]	[7]	RW	Filter Enable for EXT_INT41[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL16[4]	[6]	RW	Filter Selection for EXT_INT41[4] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH16[4]	[5:0]	RW	Filtering width of EXT_INT41[4] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1.	0x00

6.2.3.218 EXT_INT42_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E90, Reset Value = 0x8080_8080

Name	Bit	Type	Description	Reset Value
FLTEN17[3]	[31]	RW	Filter Enable for EXT_INT42[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL17[3]	[30]	RW	Filter Selection for EXT_INT42[3] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH17[3]	[29:24]	RW	Filtering width of EXT_INT42[3] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1.	0x00
FLTEN17[2]	[23]	RW	Filter Enable for EXT_INT42[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL17[2]	[22]	RW	Filter Selection for EXT_INT42[2] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH17[2]	[21:16]	RW	Filtering width of EXT_INT42[2] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1.	0x00
FLTEN17[1]	[15]	RW	Filter Enable for EXT_INT42[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL17[1]	[14]	RW	Filter Selection for EXT_INT42[1] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH17[1]	[13:8]	RW	Filtering width of EXT_INT42[1] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1.	0x00
FLTEN17[0]	[7]	RW	Filter Enable for EXT_INT42[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL17[0]	[6]	RW	Filter Selection for EXT_INT42[0] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH17[0]	[5:0]	RW	Filtering width of EXT_INT42[0] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1.	0x00

6.2.3.219 EXT_INT42_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E94, Reset Value = 0x8080_8080

Name	Bit	Type	Description	Reset Value
FLTEN17[7]	[31]	RW	Filter Enable for EXT_INT42[7] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL17[7]	[30]	RW	Filter Selection for EXT_INT42[7] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH17[7]	[29:24]	RW	Filtering width of EXT_INT42[7] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1.	0x00
FLTEN17[6]	[23]	RW	Filter Enable for EXT_INT42[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL17[6]	[22]	RW	Filter Selection for EXT_INT42[6] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH17[6]	[21:16]	RW	Filtering width of EXT_INT42[6] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1.	0x00
FLTEN17[5]	[15]	RW	Filter Enable for EXT_INT42[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL17[5]	[14]	RW	Filter Selection for EXT_INT42[5] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH17[5]	[13:8]	RW	Filtering width of EXT_INT42[5] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1.	0x00
FLTEN17[4]	[7]	RW	Filter Enable for EXT_INT42[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL17[4]	[6]	RW	Filter Selection for EXT_INT42[4] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH17[4]	[5:0]	RW	Filtering width of EXT_INT42[4] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1.	0x00

6.2.3.220 EXT_INT43_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E98, Reset Value = 0x8080_8080

Name	Bit	Type	Description	Reset Value
FLTEN18[3]	[31]	RW	Filter Enable for EXT_INT43[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL18[3]	[30]	RW	Filter Selection for EXT_INT43[3] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH18[3]	[29:24]	RW	Filtering width of EXT_INT43[3] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1.	0x00
FLTEN18[2]	[23]	RW	Filter Enable for EXT_INT43[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL18[2]	[22]	RW	Filter Selection for EXT_INT43[2] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH18[2]	[21:16]	RW	Filtering width of EXT_INT43[2] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1.	0x00
FLTEN18[1]	[15]	RW	Filter Enable for EXT_INT43[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL18[1]	[14]	RW	Filter Selection for EXT_INT43[1] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH18[1]	[13:8]	RW	Filtering width of EXT_INT43[1] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1.	0x00
FLTEN18[0]	[7]	RW	Filter Enable for EXT_INT43[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL18[0]	[6]	RW	Filter Selection for EXT_INT43[0] 0x0 = Delays filter 0x1 = Digital filter(clock count)	0x0
FLTWIDTH18[0]	[5:0]	RW	Filtering width of EXT_INT43[0] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1.	0x00

6.2.3.221 EXT_INT43_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E9C, Reset Value = 0x8080_8080

Name	Bit	Type	Description	Reset Value
FLTEN18[7]	[31]	RW	Filter Enable for EXT_INT43[7] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL18[7]	[30]	RW	Filter Selection for EXT_INT43[7] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH18[7]	[29:24]	RW	Filtering width of EXT_INT43[7] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1.	0x00
FLTEN18[6]	[23]	RW	Filter Enable for EXT_INT43[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL18[6]	[22]	RW	Filter Selection for EXT_INT43[6] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH18[6]	[21:16]	RW	Filtering width of EXT_INT43[6] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1.	0x00
FLTEN18[5]	[15]	RW	Filter Enable for EXT_INT43[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL18[5]	[14]	RW	Filter Selection for EXT_INT43[5] 0x0 = Delays filter 0x1 = Digital filter(clock count)	0x0
FLTWIDTH18[5]	[13:8]	RW	Filtering width of EXT_INT43[5] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1.	0x00
FLTEN18[4]	[7]	RW	Filter Enable for EXT_INT43[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x1
FLTSEL18[4]	[6]	RW	Filter Selection for EXT_INT43[4] 0x0 = Delays filter 0x1 = Digital filter (clock count)	0x0
FLTWIDTH18[4]	[5:0]	RW	Filtering width of EXT_INT43[4] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1.	0x00

6.2.3.222 EXT_INT40_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F00, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT40_MASK[7]	[7]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT40_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT40_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT40_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT40_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT40_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT40_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT40_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.223 EXT_INT41_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F04, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT41_MASK[7]	[7]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT41_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT41_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT41_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT41_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT41_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT41_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT41_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.224 EXT_INT42_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F08, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT42_MASK[7]	[7]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT42_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT42_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT42_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT42_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT42_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT42_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT42_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.225 EXT_INT43_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F0C, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT43_MASK[7]	[7]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT43_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT43_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT43_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT43_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT43_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT43_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT43_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.3.226 EXT_INT40_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F40, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT40_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT40_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT40_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT40_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT40_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT40_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT40_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT40_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0

6.2.3.227 EXT_INT41_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F44, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT41_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT41_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT41_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT41_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT41_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT41_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT41_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT41_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0

6.2.3.228 EXT_INT42_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F48, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT42_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT42_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT42_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT42_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT42_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT42_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT42_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT42_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0

6.2.3.229 EXT_INT43_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F4C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT43_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT43_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT43_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT43_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT43_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT43_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT43_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT43_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0

6.2.3.230 PDNEN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F80, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	–	Reserved	0x00
PDNEN_CFG	[1]	RW	0 = Automatically by power down mode 1 = By PDNEN bit	0x0
PDNEN	[0]	RW	Power down mode pad state enable register. 0 = PADs Controlled by normal mode This bit is set to "1" automatically when system enters into Power down mode and clears by writing "0" to this bit or cold reset. After wake up from Power down mode, this bit maintains value "1" until writing "0" 1 = PADs Controlled by Power Down mode control registers	0x0

6.2.4 Part 3

6.2.4.1 GPZCON

- Base Address: 0x0386_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPZCON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_0_SDO[2] 0x3 = ST_INT 0x4 to 0xE = Reserved 0xF = EXT_INT50[6]	0x00
GPZCON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_0_SDO[1] 0x3 = ST_TICK 0x4 to 0xE = Reserved 0xF = EXT_INT50[5]	0x00
GPZCON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_0_SDO[0] 0x3 = PCM_0_SOUT 0x4 to 0xE = Reserved 0xF = EXT_INT50[4]	0x00
GPZCON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_0_SDI 0x3 = PCM_0_SIN 0x4 to 0xE = Reserved 0xF = EXT_INT50[3]	0x00
GPZCON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_0_LRCK 0x3 = PCM_0_FSYNC 0x4 to 0xE = Reserved 0xF = EXT_INT50[2]	0x00
GPZCON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_0_CDCLK 0x3 = PCM_0_EXTCLK 0x4 to 0xE = Reserved 0xF = EXT_INT50[1]	0x00
GPZCON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = I2S_0_SCLK 0x3 = PCM_0_SCLK 0x4 to 0xE = Reserved 0xF = EXT_INT50[0]	0x00

6.2.4.2 GPZDAT

- Base Address: 0x0386_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPZDAT[6:0]	[6:0]	RWX	When you configure port as input port then corresponding bit is pin state. While configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.4.3 GPZPUD

- Base Address: 0x0386_0000
- Address = Base Address + 0x0008, Reset Value = 0x1555

Name	Bit	Type	Description	Reset Value
GPZPUD[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x1555

6.2.4.4 GPZDRV

- Base Address: 0x0386_0000
- Address = Base Address + 0x000C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPZDRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 6	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.4.5 GPZCONPDN

- Base Address: 0x0386_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPZ[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.4.6 GPZPUDPDN

- Base Address: 0x0386_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPZ[n]	[2n + 1:2n] n = 0 to 6	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.4.7 EXT_INT50CON

- Base Address: 0x0386_0000
- Address = Base Address + 0x0700, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT50_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT50[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT50_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT50[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT50_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT50[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT50_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT50[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT50_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT50[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7]	–	Reserved	0x0
EXT_INT50_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT50[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT50_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT50[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

6.2.4.8 EXT_INT50_FLTCON0

- Base Address: 0x0386_0000
- Address = Base Address + 0x0800, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN1[3]	[31]	RW	Filter Enable for EXT_INT50[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[3]	[30:24]	RW	Filtering width of EXT_INT50[3]	0x00
FLTEN1[2]	[23]	RW	Filter Enable for EXT_INT50[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[2]	[22:16]	RW	Filtering width of EXT_INT50[2]	0x00
FLTEN1[1]	[15]	RW	Filter Enable for EXT_INT50[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[1]	[14:8]	RW	Filtering width of EXT_INT50[1]	0x00
FLTEN1[0]	[7]	RW	Filter Enable for EXT_INT50[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[0]	[6:0]	RW	Filtering width of EXT_INT50[0]	0x00

6.2.4.9 EXT_INT50_FLTCON1

- Base Address: 0x0386_0000
- Address = Base Address + 0x0804, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
FLTEN1[6]	[23]	RW	Filter Enable for EXT_INT50[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[6]	[22:16]	RW	Filtering width of EXT_INT50[6]	0x00
FLTEN1[5]	[15]	RW	Filter Enable for EXT_INT50[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[5]	[14:8]	RW	Filtering width of EXT_INT50[5]	0x00
FLTEN1[4]	[7]	RW	Filter Enable for EXT_INT50[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[4]	[6:0]	RW	Filtering width of EXT_INT50[4]	0x00

6.2.4.10 EXT_INT50_MASK

- Base Address: 0x0386_0000
- Address = Base Address + 0x0900, Reset Value = 0x0000_007F

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT50_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT50_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT50_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT50_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT50_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT50_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT50_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.4.11 EXT_INT50_PEND

- Base Address: 0x0386_0000
- Address = Base Address + 0x0A00, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0000000
EXT_INT50_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT50_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT50_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT50_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0
EXT_INT50_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0
EXT_INT50_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0
EXT_INT50_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0

6.2.4.12 EXT_INT_SERVICE_XD

- Base Address: 0x0386_0000
- Address = Base Address + 0x0B08, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	RW	Reserved	0x0000000
SVC_Group_Num	[7:3]	RW	EXT_INT Service group number 0x1 = EXT_INT50	0x00
SVC_Num	[2:0]	RW	Interrupt number to be serviced	0x0

6.2.4.13 EXT_INT_SERVICE_PEND_XD

- Base Address: 0x0386_0000
- Address = Base Address + 0x0B0C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	RW	Reserved	0x0000000
SVC_PEND	[7:0]	RW	0x0 = Not occur 0x1 = Interrupt Occurs	0x00

6.2.4.14 EXT_INT_GRPFIXPRI_XD

- Base Address: 0x0386_0000
- Address = Base Address + 0x0B10, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0000000
Highest_GRP_NUM	[3:0]	RW	When fixed group priority mode = 0, then group number should be of the highest priority. 0x0 = EXT_INT50	0x00

6.2.4.15 EXT_INT50_FIXPRI

- Base Address: 0x0386_0000
- Address = Base Address + 0x0B14, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 0 (EXT_INT50) when fixed priority mode: 0 to 7	0x0

6.2.4.16 PDNEN

- Base Address: 0x0386_0000
- Address = Base Address + 0x0F80, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	–	Reserved	0x00
PDNEN_CFG	[1]	RW	0 = Automatically by power down mode 1 = By PDNEN bit	0x0
PDNEN	[0]	RW	Power down mode pad state enable register. 0 = PADs Controlled by normal mode This bit is set to "1" automatically when system enters into Power down mode and clears by writing "0" to this bit or cold reset. After wake up from Power down mode, this bit maintains value "1" until writing "0" 1 = PADs Controlled by Power Down mode control registers	0x0

6.2.5 Part 4

6.2.5.1 GPV0CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPV0CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[7] 0x3 to 0xE = Reserved 0xF = EXT_INT30[7]	0x00
GPV0CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[6] 0x3 to 0xE = Reserved 0xF = EXT_INT30[6]	0x00
GPV0CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[5] 0x3 to 0xE = Reserved 0xF = EXT_INT30[5]	0x00
GPV0CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[4] 0x3 to 0xE = Reserved 0xF = EXT_INT30[4]	0x00
GPV0CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[3] 0x3 to 0xE = Reserved 0xF = EXT_INT30[3]	0x00
GPV0CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[2] 0x3 to 0xE = Reserved 0xF = EXT_INT30[2]	0x00
GPV0CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[1] 0x3 to 0xE = Reserved 0xF = EXT_INT30[1]	0x00
GPV0CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[0] 0x3 to 0xE = Reserved 0xF = EXT_INT30[0]	0x00

6.2.5.2 GPV0DAT

- Base Address: 0x106E_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPV0DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.5.3 GPV0PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x0008, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPV0PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.5.4 GPV0DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x000C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPV0DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.5.5 GPV0CONPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPV0[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.5.6 GPV0PUDPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPV0[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.5.7 GPV1CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPV1CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[15] 0x3 to 0xE = Reserved 0xF = EXT_INT31[7]	0x00
GPV1CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[14] 0x3 to 0xE = Reserved 0xF = EXT_INT31[6]	0x00
GPV1CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[13] 0x3 to 0xE = Reserved 0xF = EXT_INT31[5]	0x00
GPV1CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[12] 0x3 to 0xE = Reserved 0xF = EXT_INT31[4]	0x00
GPV1CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[11] 0x3 to 0xE = Reserved 0xF = EXT_INT31[3]	0x00
GPV1CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[10] 0x3 to 0xE = Reserved 0xF = EXT_INT31[2]	0x00
GPV1CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[9] 0x3 to 0xE = Reserved 0xF = EXT_INT31[1]	0x00
GPV1CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_RXD[8] 0x3 to 0xE = Reserved 0xF = EXT_INT31[0]	0x00

6.2.5.8 GPV1DAT

- Base Address: 0x106E_0000
- Address = Base Address + 0x0024, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPV1DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.5.9 GPV1PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x0028, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPV1PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.5.10 GPV1DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x002C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPV1DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.5.11 GPV1CONPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPV1[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.5.12 GPV1PUDPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPV1[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.5.13 ETC7PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x0048, Reset Value = 0x0005

Name	Bit	Type	Description	Reset Value
ETC7PUD[n]	[2n + 1:2n] n = 0 to 1	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0005

ETC7PUD[1:0] controls Xc2cRXCLK[0].

ETC7PUD[3:2] controls Xc2cRXCLK[1].

6.2.5.14 ETC7DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x004C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
ETC7DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 1	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

ETC7DRV[1:0] controls Xc2cRXCLK[0].

ETC7DRV[3:2] controls Xc2cRXCLK[1].

6.2.5.15 GPV2CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPV2CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[7] 0x3 to 0xE = Reserved 0xF = EXT_INT32[7]	0x00
GPV2CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[6] 0x3 to 0xE = Reserved 0xF = EXT_INT32[6]	0x00
GPV2CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[5] 0x3 to 0xE = Reserved 0xF = EXT_INT32[5]	0x00
GPV2CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[4] 0x3 to 0xE = Reserved 0xF = EXT_INT32[4]	0x00
GPV2CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[3] 0x3 to 0xE = Reserved 0xF = EXT_INT32[3]	0x00
GPV2CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[2] 0x3 to 0xE = Reserved 0xF = EXT_INT32[2]	0x00
GPV2CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[1] 0x3 to 0xE = Reserved 0xF = EXT_INT32[1]	0x00
GPV2CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[0] 0x3 to 0xE = Reserved 0xF = EXT_INT32[0]	0x00

6.2.5.16 GPV2DAT

- Base Address: 0x106E_0000
- Address = Base Address + 0x0064, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPV2DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.5.17 GPV2PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x0068, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPV2PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x5555

6.2.5.18 GPV2DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x006C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPV2DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.5.19 GPV2CONPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0070, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPV2[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.5.20 GPV2PUDPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0074, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPV2[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.5.21 GPV3CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPV3CON[7]	[31:28]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[15] 0x3 to 0xE = Reserved 0xF = EXT_INT33[7]	0x00
GPV3CON[6]	[27:24]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[14] 0x3 to 0xE = Reserved 0xF = EXT_INT33[6]	0x00
GPV3CON[5]	[23:20]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[13] 0x3 to 0xE = Reserved 0xF = EXT_INT33[5]	0x00
GPV3CON[4]	[19:16]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[12] 0x3 to 0xE = Reserved 0xF = EXT_INT33[4]	0x00
GPV3CON[3]	[15:12]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[11] 0x3 to 0xE = Reserved 0xF = EXT_INT33[3]	0x00
GPV3CON[2]	[11:8]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[10] 0x3 to 0xE = Reserved 0xF = EXT_INT33[2]	0x00
GPV3CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[9] 0x3 to 0xE = Reserved 0xF = EXT_INT33[1]	0x00
GPV3CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_TXD[8] 0x3 to 0xE = Reserved 0xF = EXT_INT33[0]	0x00

6.2.5.22 GPV3DAT

- Base Address: 0x106E_0000
- Address = Base Address + 0x0084, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPV3DAT[7:0]	[7:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.5.23 GPV3PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x0088, Reset Value = 0x5555

Name	Bit	Type	Description	Reset Value
GPV3PUD[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved, 0x3 = Enables Pull-up	0x5555

6.2.5.24 GPV3DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x008C, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPV3DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 7	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.5.25 GPV3CONPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPV3[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.5.26 GPV3PUDPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0094, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPV3[n]	[2n + 1:2n] n = 0 to 7	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.5.27 ETC8PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x00A8, Reset Value = 0x0005

Name	Bit	Type	Description	Reset Value
ETC8PUD[n]	[2n + 1:2n] n = 0 to 1	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0005

ETC8PUD[1:0] controls Xc2cTXCLK[0].

ETC8PUD[3:2] controls Xc2cTXCLK[1].

6.2.5.28 ETC8DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x00AC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
ETC8DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 1	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

ETC8DRV[1:0] controls Xc2cTXCLK[0].

ETC8DRV[3:2] controls Xc2cTXCLK[1].

6.2.5.29 GPV4CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GPV4CON[1]	[7:4]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_WKREQOUT 0x3 to 0xE = Reserved 0xF = EXT_INT34[1]	0x00
GPV4CON[0]	[3:0]	RW	0x0 = Input 0x1 = Output 0x2 = C2C_WKREQIN 0x3 to 0xE = Reserved 0xF = EXT_INT34[0]	0x00

6.2.5.30 GPV4DAT

- Base Address: 0x106E_0000
- Address = Base Address + 0x00C4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GPV4DAT[1:0]	[1:0]	RWX	When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read.	0x00

6.2.5.31 GPV4PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x00C8, Reset Value = 0x0005

Name	Bit	Type	Description	Reset Value
GPV4PUD[n]	[2n + 1:2n] n = 0 to 1	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x0005

6.2.5.32 GPV4DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x00CC, Reset Value = 0x00_0000

Name	Bit	Type	Description	Reset Value
GPV4DRV[n]	[23:16]	RW	Reserved (Should be zero)	0x00
	[2n + 1:2n] n = 0 to 1	RW	0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x	0x0000

6.2.5.33 GPV4CONPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x00D0, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPV4[n]	[2n + 1:2n] n = 0 to 1	RW	0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state	0x00

6.2.5.34 GPV4PUDPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x00D4, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
GPV4[n]	[2n + 1:2n] n = 0 to 1	RW	0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up	0x00

6.2.5.35 EXT_INT30CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0700, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT30_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT30[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT30_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT30[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT30_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT30[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT30_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT30[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT30_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT30[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT30_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT30[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT30_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT30[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT30_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT30[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

6.2.5.36 EXT_INT31CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0704, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT31_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT31[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT31_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT31[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT31_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT31[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT31_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT31[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT31_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT31[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT31_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT31[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT31_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT31[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT31_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT31[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

6.2.5.37 EXT_INT32CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0708, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT32_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT32[7] 0x0 = Low level 0x1 = High level0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT32_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT32[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT32_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT32[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT32_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT32[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT32_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT32[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0
EXT_INT32_CON[2]	[10:8]	RW	Sets the signaling method of EXT_INT32[2]	0x0

Name	Bit	Type	Description	Reset Value
			0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	
RSVD	[7]	–	Reserved	0x0
EXT_INT32_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT32[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT32_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT32[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

6.2.5.38 EXT_INT33CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x070C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
EXT_INT33_CON[7]	[30:28]	RW	Sets signaling method of EXT_INT33[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[27]	–	Reserved	0x0
EXT_INT33_CON[6]	[26:24]	RW	Sets signaling method of EXT_INT33[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[23]	–	Reserved	0x0
EXT_INT33_CON[5]	[22:20]	RW	Sets signaling method of EXT_INT33[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[19]	–	Reserved	0x0
EXT_INT33_CON[4]	[18:16]	RW	Sets signaling method of EXT_INT33[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[15]	–	Reserved	0x0
EXT_INT33_CON[3]	[14:12]	RW	Sets signaling method of EXT_INT33[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[11]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
EXT_INT33_CON[2]	[10:8]	RW	Sets signaling method of EXT_INT33[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[7]	–	Reserved	0x0
EXT_INT33_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT33[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT33_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT33[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

6.2.5.39 EXT_INT34CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0710, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
RSVD	[7]	–	Reserved	0x0
EXT_INT34_CON[1]	[6:4]	RW	Sets signaling method of EXT_INT34[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0
RSVD	[3]	–	Reserved	0x0
EXT_INT34_CON[0]	[2:0]	RW	Sets signaling method of EXT_INT34[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved	0x0

6.2.5.40 EXT_INT30_FLTCON0

- Base Address: 0x106E_0000
- Address = Base Address + 0x0800, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN1[3]	[31]	RW	Filter Enable for EXT_INT30[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[3]	[30:24]	RW	Filtering width of EXT_INT30[3]	0x00
FLTEN1[2]	[23]	RW	Filter Enable for EXT_INT30[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[2]	[22:16]	RW	Filtering width of EXT_INT30[2]	0x00
FLTEN1[1]	[15]	RW	Filter Enable for EXT_INT30[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[1]	[14:8]	RW	Filtering width of EXT_INT30[1]	0x00
FLTEN1[0]	[7]	RW	Filter Enable for EXT_INT30[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[0]	[6:0]	RW	Filtering width of EXT_INT30[0]	0x00

6.2.5.41 EXT_INT30_FLTCON1

- Base Address: 0x106E_0000
- Address = Base Address + 0x0804, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN1[7]	[31]	RW	Filter Enable for EXT_INT30[7] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[7]	[30:24]	RW	Filtering width of EXT_INT30[7]	0x00
FLTEN1[6]	[23]	RW	Filter Enable for EXT_INT30[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[6]	[22:16]	RW	Filtering width of EXT_INT30[6]	0x00
FLTEN1[5]	[15]	RW	Filter Enable for EXT_INT30[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[5]	[14:8]	RW	Filtering width of EXT_INT30[5]	0x00
FLTEN1[4]	[7]	RW	Filter Enable for EXT_INT30[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH1[4]	[6:0]	RW	Filtering width of EXT_INT30[4]	0x00

6.2.5.42 EXT_INT31_FLTCON0

- Base Address: 0x106E_0000
- Address = Base Address + 0x0808, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN2[3]	[31]	RW	Filter Enable for EXT_INT31[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH2[3]	[30:24]	RW	Filtering width of EXT_INT31[3]	0x00
FLTEN2[2]	[23]	RW	Filter Enable for EXT_INT31[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH2[2]	[22:16]	RW	Filtering width of EXT_INT31[2]	0x00
FLTEN2[1]	[15]	RW	Filter Enable for EXT_INT31[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH2[1]	[14:8]	RW	Filtering width of EXT_INT31[1]	0x00
FLTEN2[0]	[7]	RW	Filter Enable for EXT_INT31[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH2[0]	[6:0]	RW	Filtering width of EXT_INT31[0]	0x00

6.2.5.43 EXT_INT31_FLTCON1

- Base Address: 0x106E_0000
- Address = Base Address + 0x080C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN2[7]	[31]	RW	Filter Enable for EXT_INT31[7] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH2[7]	[30:24]	RW	Filtering width of EXT_INT31[7]	0x00
FLTEN2[6]	[23]	RW	Filter Enable for EXT_INT31[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH2[6]	[22:16]	RW	Filtering width of EXT_INT31[6]	0x00
FLTEN2[5]	[15]	RW	Filter Enable for EXT_INT31[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH2[5]	[14:8]	RW	Filtering width of EXT_INT31[5]	0x00
FLTEN2[4]	[7]	RW	Filter Enable for EXT_INT31[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH2[4]	[6:0]	RW	Filtering width of EXT_INT31[4]	0x00

6.2.5.44 EXT_INT32_FLTCON0

- Base Address: 0x106E_0000
- Address = Base Address + 0x0810, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN3[3]	[31]	RW	Filter Enable for EXT_INT32[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[3]	[30:24]	RW	Filtering width of EXT_INT32[3]	0x00
FLTEN3[2]	[23]	RW	Filter Enable for EXT_INT32[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[2]	[22:16]	RW	Filtering width of EXT_INT32[2]	0x00
FLTEN3[1]	[15]	RW	Filter Enable for EXT_INT32[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[1]	[14:8]	RW	Filtering width of EXT_INT32[1]	0x00
FLTEN3[0]	[7]	RW	Filter Enable for EXT_INT32[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[0]	[6:0]	RW	Filtering width of EXT_INT32[0]	0x00

6.2.5.45 EXT_INT32_FLTCON1

- Base Address: 0x106E_0000
- Address = Base Address + 0x0814, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN3[7]	[31]	RW	Filter Enable for EXT_INT32[7] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[7]	[30:24]	RW	Filtering width of EXT_INT32[7]	0x00
FLTEN3[6]	[23]	RW	Filter Enable for EXT_INT32[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[6]	[22:16]	RW	Filtering width of EXT_INT32[6]	0x00
FLTEN3[5]	[15]	RW	Filter Enable for EXT_INT32[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[5]	[14:8]	RW	Filtering width of EXT_INT32[5]	0x00
FLTEN3[4]	[7]	RW	Filter Enable for EXT_INT32[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH3[4]	[6:0]	RW	Filtering width of EXT_INT32[4]	0x00

6.2.5.46 EXT_INT33_FLTCON0

- Base Address: 0x106E_0000
- Address = Base Address + 0x0818, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN4[3]	[31]	RW	Filter Enable for EXT_INT33[3] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[3]	[30:24]	RW	Filtering width of EXT_INT33[3]	0x00
FLTEN4[2]	[23]	RW	Filter Enable for EXT_INT33[2] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[2]	[22:16]	RW	Filtering width of EXT_INT33[2]	0x00
FLTEN4[1]	[15]	RW	Filter Enable for EXT_INT33[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[1]	[14:8]	RW	Filtering width of EXT_INT33[1]	0x00
FLTEN4[0]	[7]	RW	Filter Enable for EXT_INT33[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[0]	[6:0]	RW	Filtering width of EXT_INT33[0]	0x00

6.2.5.47 EXT_INT33_FLTCON1

- Base Address: 0x106E_0000
- Address = Base Address + 0x081C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FLTEN4[7]	[31]	RW	Filter Enable for EXT_INT33[7] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[7]	[30:24]	RW	Filtering width of EXT_INT33[7]	0x00
FLTEN4[6]	[23]	RW	Filter Enable for EXT_INT33[6] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[6]	[22:16]	RW	Filtering width of EXT_INT33[6]	0x00
FLTEN4[5]	[15]	RW	Filter Enable for EXT_INT33[5] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[5]	[14:8]	RW	Filtering width of EXT_INT33[5]	0x00
FLTEN4[4]	[7]	RW	Filter Enable for EXT_INT33[4] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH4[4]	[6:0]	RW	Filtering width of EXT_INT33[4]	0x00

6.2.5.48 EXT_INT34_FLTCON0

- Base Address: 0x106E_0000
- Address = Base Address + 0x0820, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
FLTEN5[1]	[15]	RW	Filter Enable for EXT_INT34[1] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH5[1]	[14:8]	RW	Filtering width of EXT_INT34[1]	0x00
FLTEN5[0]	[7]	RW	Filter Enable for EXT_INT34[0] 0x0 = Disables Filter 0x1 = Enables Filter	0x0
FLTWIDTH5[0]	[6:0]	RW	Filtering width of EXT_INT34[0]	0x00

6.2.5.49 EXT_INT34_FLTCON1

- Base Address: 0x106E_0000
- Address = Base Address + 0x0824, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:0]	–	Reserved	0x00000000

6.2.5.50 EXT_INT30_MASK

- Base Address: 0x106E_0000
- Address = Base Address + 0x0900, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT30_MASK[7]	[7]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT30_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT30_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT30_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT30_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT30_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT30_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT30_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.5.51 EXT_INT31_MASK

- Base Address: 0x106E_0000
- Address = Base Address + 0x0904, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT31_MASK[7]	[7]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT31_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT31_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT31_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT31_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT31_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT31_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT31_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.5.52 EXT_INT32_MASK

- Base Address: 0x106E_0000
- Address = Base Address + 0x0908, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT32_MASK[7]	[7]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT32_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT32_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT32_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT32_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT32_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT32_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT32_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.5.53 EXT_INT33_MASK

- Base Address: 0x106E_0000
- Address = Base Address + 0x090C, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT33_MASK[7]	[7]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT33_MASK[6]	[6]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT33_MASK[5]	[5]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT33_MASK[4]	[4]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT33_MASK[3]	[3]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT33_MASK[2]	[2]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT33_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT33_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.5.54 EXT_INT34_MASK

- Base Address: 0x106E_0000
- Address = Base Address + 0x0910, Reset Value = 0x0000_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x00000000
EXT_INT34_MASK[1]	[1]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1
EXT_INT34_MASK[0]	[0]	RW	0x0 = Enables Interrupt 0x1 = Masked	0x1

6.2.5.55 EXT_INT30_PEND

- Base Address: 0x106E_0000
- Address = Base Address + 0x0A00, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT30_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT30_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT30_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT30_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT30_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT30_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT30_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT30_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0

6.2.5.56 EXT_INT31_PEND

- Base Address: 0x106E_0000
- Address = Base Address + 0x0A04, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT31_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT31_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT31_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT31_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT31_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT31_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0
EXT_INT31_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0
EXT_INT31_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0

6.2.5.57 EXT_INT32_PEND

- Base Address: 0x106E_0000
- Address = Base Address + 0x0A08, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT32_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT32_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT32_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0
EXT_INT32_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0
EXT_INT32_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0
EXT_INT32_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT32_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT32_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0

6.2.5.58 EXT_INT33_PEND

- Base Address: 0x106E_0000
- Address = Base Address + 0x0A0C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
EXT_INT33_PEND[7]	[7]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT33_PEND[6]	[6]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT33_PEND[5]	[5]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0
EXT_INT33_PEND[4]	[4]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0
EXT_INT33_PEND[3]	[3]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0
EXT_INT33_PEND[2]	[2]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT33_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT33_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0

6.2.5.59 EXT_INT34_PEND

- Base Address: 0x106E_0000
- Address = Base Address + 0x0A10, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x00000000
EXT_INT34_PEND[1]	[1]	RWX	0x0 = Not occur 0x1 = Interrupt Occurs	0x0
EXT_INT34_PEND[0]	[0]	RWX	0x0 = Not occur 0x1 = interrupt Occurs	0x0

6.2.5.60 EXT_INT_SERVICE_XC

- Base Address: 0x106E_0000
- Address = Base Address + 0x0B08, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	RW	Reserved	0x0000000
SVC_Group_Num	[7:3]	RW	EXT_INT Service group number 0x1 = EXT_INT30 0x2 = EXT_INT31 0x3 = EXT_INT32 0x4 = EXT_INT33 0x5 = EXT_INT34	0x00
SVC_Num	[2:0]	RW	Interrupt number to be serviced	0x0

6.2.5.61 EXT_INT_SERVICE_PEND_XC

- Base Address: 0x106E_0000
- Address = Base Address + 0x0B0C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	RW	Reserved	0x0000000
SVC_PEND	[7:0]	RW	0x0 = Not occur 0x1 = Interrupt Occurs	0x00

6.2.5.62 EXT_INT_GRPFIXPRI_XC

- Base Address: 0x106E_0000
- Address = Base Address + 0x0B10, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0000000
Highest_GRP_NUM	[3:0]	RW	Group number of the highest priority when fixed group priority mode: 0 to 4 0x0 = EXT_INT30 0x1 = EXT_INT31 0x2 = EXT_INT32 0x3 = EXT_INT33 0x4 = EXT_INT34	0x00

6.2.5.63 EXT_INT30_FIXPRI

- Base Address: 0x106E_0000
- Address = Base Address + 0x0B14, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 0 (EXT_INT30) when fixed priority mode: 0 to 7	0x0

6.2.5.64 EXT_INT31_FIXPRI

- Base Address: 0x106E_0000
- Address = Base Address + 0x0B18, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 1 (EXT_INT31) when fixed priority mode: 0 to 7	0x0

6.2.5.65 EXT_INT32_FIXPRI

- Base Address: 0x106E_0000
- Address = Base Address + 0x0B1C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 2 (EXT_INT32) when fixed priority mode: 0 to 7	0x0

6.2.5.66 EXT_INT33_FIXPRI

- Base Address: 0x106E_0000
- Address = Base Address + 0x0B20, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 3 (EXT_INT33) when fixed priority mode: 0 to 7	0x0

6.2.5.67 EXT_INT34_FIXPR

- Base Address: 0x106E_0000
- Address = Base Address + 0x0B24, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x00000000
Highest_EINT_NUM	[2:0]	RW	Interrupt number of the highest priority in External Interrupt Group 4 (EXT_INT34) when fixed priority mode: 0 to 7	0x0

6.2.5.68 PDNEN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0F80, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	–	Reserved	0x00
PDNEN_CFG	[1]	RW	0 = Automatically by power down mode 1 = by PDNEN bit	0x0
PDNEN	[0]	RW	Power down mode pad state enable register. 0 = PADs Controlled by normal mode 1 = PADs Controlled by Power Down mode control registers This bit is set to "1" automatically when system enters into Power down mode and clears by writing "0" to this bit or cold reset. After wake up from Power down mode, this bit maintains value "1" until writing "0"	0x0

7 Clock Management Unit

This chapter describes the Clock Management Units (CMUs) of Exynos 4412 SCP. CMUs control Phase Locked Loops (PLLs) and generate system clocks for CPU, buses, and function clocks for individual IPs in Exynos 4412 SCP. They also communicate with the power management unit (PMU) in order to stop clocks before entering certain low power mode to reduce power consumption by minimizing clock toggling.

7.1 Clock Domains

In Exynos 4412 SCP, it clocks the function blocks asynchronously with each other to provide a wider choice of operating frequencies. It also eases physical implementation.

- The CPU block consists of the Cortex-A9 MPCore processor, L2 cache controller, and CoreSight. It operates at voltage levels of 0.875 V ~ 1.30 V. The Cortex-A9 MPCore operates at 200 MHz ~ 1.4 GHz and CoreSight Clock is up to 200MHz. The CMU in CPU block (CMU_CPU) generates all the necessary clocks for IPs in CPU block. It also generates certain control signals for Cortex-A9 MPCore.
- DMC block consists of the DRAM memory controller (DMC), Security sub-system (SSS), and Generic Interrupt Controller (GIC). CMU in DMC block (CMU_DMC) generates 400 MHz DRAM clock, 200 MHz AXI bus clock which is synchronized with the DRAM clock, and 100MHz clock for register accesses. It also generates 200 MHz clock for Accelerator Coherency Port (ACP) bus, which is used for memory coherency checking and connects CPU and SSS bus masters.
- The LEFTBUS and RIGHTBUS blocks contain the global data buses that are clocked at 200 MHz. The global data buses transfer data between the DRAM and various sub-blocks. It also contains global peripheral buses that are clocked at 100 MHz. You can use 100 MHz clock for register accesses.
- CMU_TOP generates clocks for all the remaining function blocks, which include G3D, MFC, LCD0, ISP, CAM, TV, FSYS, MFC, GPS, MAUDIO, PERIL, and PERIR. It generates bus clocks that operate at 400 / 200 / 160 / 133 / 100 MHz. It also generates various special clocks to operate IPs in Exynos 4412 SCP.
- Additionally asynchronous bus bridges are inserted between two different function blocks.

Figure 7-1 illustrates the Exynos 4412 SCP clock domains.

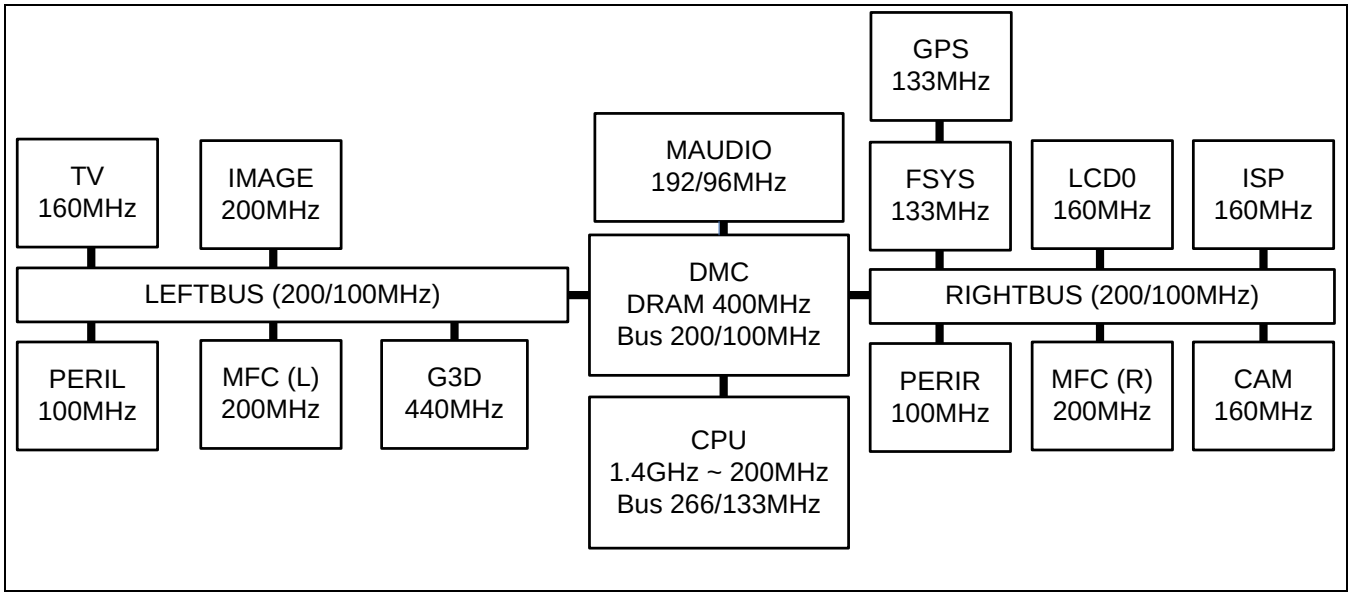


Figure 7-1 Exynos 4412 SCP Clock Domains

Table 7-1 describes the typical operating frequencies for each function block in Exynos 4412 SCP.

Table 7-1 Operating Frequencies in Exynos 4412 SCP

Function Block	Description	Typical Operating Frequency
CPU	Cortex-A9 MPCore It is a Quad Core processor.	200 MHz ~ 1.4 GHz
	CoreSight	200 MHz/100 MHz
DMC	DMC, 2D Graphics Engine	400 MHz(up to 200MHz for G2D)
SSS	Security Sub-System	200 MHz
LEFTBUS	Data Bus/Peripheral Bus	200 MHz/100 MHz
RIGHTBUS	Data Bus/Peripheral Bus	200 MHz/100 MHz
G3D	3D Graphics Engine	440 MHz
MFC	Multi-format Codec	200 MHz
IMAGE	Rotator, MDMA	200 MHz
LCD0	FIMD0, MIE0, MIPI DSI0	160 MHz
ISP	ISP	160 MHz
CAM	FIMC0, FIMC1, FIMC2, FIMC3 JPEG	160 MHz
TV	VP, MIXER, TVENC	160 MHz
FSYS	USB, PCIe, SDMMC, TSI, OneNANDC, SROMC, PDMA0, PDMA1, NFCON, MIPI-HIS, ADC	133 MHz
GPS	GPS	133 MHz
MAUDIO	AudioSS, iROM, iRAM	192 MHz
PERI-L	UART, I2C, SPI, I2S, PCM, SPDIF, PWM, I2CHDMI, Slimbus	100 MHz
PERI-R	CHIPID, SYSREG, PMU/CMU/TMU Bus I/F, MCTimer, WDT, RTC, KEYIF, SECKEY, TZPC	100 MHz

NOTE: Refer to Figure 1-4 in *Chapter 39 Audio Subsystem*, for more details on MAUDIO block clocks.

7.2 Clock Declaration

The top-level clocks in Exynos 4412 SCP are:

- Clocks from clock pads, namely, XRTCXTI, XXTI, and XUSBXTI.
- Clocks from CMUs
  For instance, ARMCLK, ACLK, HCLK, and SCLK
  ARMCLK specifies clock for Cortex-A9 MPCore (up to 800 MHz @1.0 V, 1 GHz @ 1.1 V).
  ACLK, HCLK, PCLK specify bus clocks.
  SCLK (Special clock) specifies all clocks except bus clocks and processor core clock.
- Clocks from USB PHY
- Clocks from HDMI_PHY
- Clocks from GPIO pads

7.2.1 Clocks from Clock Pads

The clock pads derive the clocks. They are:

- **XRTCXTI**: Specifies the clock generated from the crystal pad of 32.768 KHz with XRTCXTI and XRTCXTO pins. XRTCXTI and XRTCXTO are the two pins of crystal pad. RTC uses this clock as a source to the real-time clock. It requires a parallel resistance of 10 MΩ between the XUSBXTI and XUSBXTO pins.
- **XXTI**: Specifies the clock from external oscillator with XXTI pins. XXTI use wide-range OSC pads. When USB PHY is not used in commercial set, CMUs and phase-locked loops, namely, APLL, MPLL, VPLL, and EPLL use this clock as a supply source for appropriate modules. The input frequency of the clock ranges from 12 MHz to 50 MHz. When XXTI pin is not used, the pin should be tied to ground (GND). You can use the XXTI pin only for testing purpose.
- **XUSBXTI:** Specifies the clock from crystal pad with XUSBXTI and XUSBXTO pins. XUSBXTI and XUSBXTO use wide-range OSC pads. This clock is supplied to the USB PHY and the phase locked loops, namely, APLL, MPLL, VPLL, and EPLL. Refer to Chapter 36 USB HOST and Chapter 37 USB DEVICE, for more information. We recommend using a 24 MHz crystal as the iROM design is based on the 24 MHz input clock. It requires parallel resistance of 5 MΩ between the XUSBXTI and XUSBXTO pins.

7.2.2 Clocks from CMU

CMUs generate internal clocks with intermediate frequencies using from clocks from the clock pads. They are:

- Clock pads, namely, XRTCXTI, XXTI, and XUSBXTI
- Four PLLs, namely, APLL, MPLL, EPLL, and VPLL
- USB PHY and HDMI PHY

Some of these clocks are selected, pre-scaled, and provided to the corresponding modules.

We recommend using 24 MHz input clock source for APLL, MPLL, EPLL, and VPLL.

The components to generate internal clocks are:

- APLL uses FINPLL as input to generate frequencies from 22 to 1400 MHz.
- MPLL uses FINPLL as input to generate frequencies from 22 to 1400 MHz.
- EPLL uses FINPLL as input to generate frequencies from 22 to 1400 MHz. This PLL generates a 192 MHz clock for the Audio Sub-system. It divides EPLL output to generate 24 MHz SLIMbus clock.
- VPLL uses FINPLL or SCLK_HDMI24M as input to generate frequencies from 22 to 1400 MHz. This PLL generates 54 MHz video clock or G3D clock.
- USB Device PHY uses XUSBXTI to generate frequencies of 30 and 48 MHz.
- HDMI PHY uses XUSBXTI to generate 54 MHz.

In typical Exynos 4412 SCP applications,

- Cortex-A9 MPCore, CoreSight, and HPM use APLL.
- DRAM, system bus clocks, and other peripheral clocks like audio IPs, and SPI use MPLL and EPLL.
- Video clock uses VPLL.
- G3D uses MPLL or VPLL as input clock source.

Clock controllers allow bypassing of PLLs for low frequency clock. They also provide clock gating to each block, thereby reducing power consumption.

7.3 Clock Relationship

The clock relationship between various clocks are:

- CPU_BLK clocks
  - freq (ARMCLK) = freq (MOUTCORE)/n, where n = 1 to 16
  - freq (ACLK_COREM0) = freq (ARMCLK)/n, where n = 1 to 8
  - freq (ACLK_COREM1) = freq (ARMCLK)/n, where n = 1 to 8
  - freq (PERIPHCLK) = freq (ARMCLK)/n, where n = 1 to 8
  - freq (ATCLK) = freq (MOUTCORE)/n, where n = 1 to 8
  - freq (PCLK_DBG) = freq (ATCLK)/n, where n = 1 to 8
- DMC_BLK clocks
  - freq (SCLK_DMC) = freq (MOUTDMC_BUS)/n, where n = 1 to 8
  - freq (ACLK_DMCD) = freq (SCLK_DMC)/n, where n = 1 to 8
  - freq (ACLK_DMCP) = freq (ACLK_DMCD)/n, where n = 1 to 8
  - freq (ACLK_ACP) = freq (MOUTDMC_BUS)/n, where n = 1 to 8
  - freq (PCLK_ACP) = freq (ACLK_ACP)/n, where n = 1 to 8
  - freq (SCLK_C2C) = freq (MOUTC2C)/n, where n = 1 to 8
  - freq (ACLK_C2C) = freq (SCLK_C2C)/n, where n = 1 to 8
- LEFTBUS_BLK clocks
  - freq (ACLK_GDL) = freq (MOUTGDL)/n, where n = 1 to 8
  - freq (ACLK_GPL) = freq (ACLK_GDL)/n, where n = 1 to 8
- RIGHTBUS_BLK clocks
  - freq (ACLK_GDR) = freq (MOUTGDR)/n, where n = 1 to 8
  - freq (ACLK_GPR) = freq (ACLK_GDR)/n, where n = 1 to 8
- CMU_TOP clocks
  - freq (ACLK_400_MCUISP = freq (MOUTACLK_400_mcuisp)/n, where n = 1 to 8
  - freq (ACLK_200) = freq (MOUTACLK_200)/n, where n = 1 to 8
  - freq (ACLK_100) = freq (MOUTACLK_100)/n, where n = 1 to 16
  - freq (ACLK_160) = freq (MOUTACLK_160)/n, where n = 1 to 8
  - freq (ACLK_133) = freq (MOUTACLK_133)/n, where n = 1 to 8
  - freq (SCLK_ONENAND) = freq (MOUTONENAND)/n, where n = 1 to 8
- MAUDIO_BLK clocks
  - freq (RP_CLK) = freq (MOUTASS)/n, where n = 1 to 16
  - freq (BUS_CLK) = freq (MOUTRP)/n, where n = 1 to 16

NOTE: *Figure 7-4* of *Chapter 39 Audio Subsystem* illustrates the clock names including iROM/iRAM and clock tree diagram of MAUDIO_BLK.

Caution: Ensure that the ratio between the SCLK_DMC and ACLK_DMCD frequency should be 2:1 or 1:1 always. Do not change this ratio during the running state of DMC. You should also ensure that the ratio between the SCLK_C2C and ACLK_C2C frequency should be 2:1. You should not change this ratio during the running state of C2C.

The values for high-performance operation are:

- freq (ARMCLK) = 1400 MHz at 1.3 V
- freq (ACLK_COREM0) = 350 MHz at 1.3V
- freq (ACLK_COREM1) = 188 MHz at 1.3 V
- freq (PERIPHCLK) = 1400 MHz at 1.3 V
- freq (ATCLK) = 214 MHz at 1.3 V
- freq (PCLK_DBG) = 107 MHz at 1.3 V
- freq (SCLK_DMC) = 400 MHz at 1.0 V
- freq (ACLK_DMCD) = 200 MHz at 1.0 V
- freq (ACLK_DMCP) = 100 MHz at 1.0 V
- freq (ACLK_ACP) = 200 MHz at 1.0 V
- freq (PCLK_ACP) = 100 MHz at 1.0 V
- freq (SCLK_C2C) = 400 MHz at 1.0 V
- freq (ACLK_C2C) = 200 MHz at 1.0 V
- freq (ACLK_GDL) = 200 MHz at 1.0 V
- freq (ACLK_GPL) = 100 MHz at 1.0 V
- freq (ACLK_GDR) = 200 MHz at 1.0 V
- freq (ACLK_GPR) = 100 MHz at 1.0 V
- freq (ACLK_400_MCUISP) = 400 MHz at 1.0 V
- freq (ACLK_200) = 160 MHz at 1.0 V
- freq (ACLK_100) = 100 MHz at 1.0 V
- freq (ACLK_160) = 160 MHz at 1.0 V
- freq (ACLK_133) = 133 MHz at 1.0 V
- freq (SCLK_ONENAND) = 160 MHz at 1.0 V

The PLL operations are:

- APLL mainly drives the CPU_BLK clocks. It generates frequencies up to 1.4 GHz with a duty ratio of 49:51. APLL also generates DMC_BLK, LEFTBUS_BLK, RIGHTBUS_BLK, and CMU_TOP clocks as supplement of MPLL.
- MPLL mainly drives the DMC_BLK, LEFTBUS_BLK, RIGHTBUS_BLK, and CMU_TOP clocks. It generates frequencies up to 1 GHz with a duty ratio of 49:51. MPLL also generates CPU_BLK clocks when it blocks APLL for locking during the Dynamic Voltage Frequency Scaling (DVFS).
- EPLL mainly generates an audio clock.
- VPLL mainly generates video system operating clock of 54 MHz, or a G3D clock, or 440 MHz clock at 1.1 V.

7.3.1 Recommended PLL PMS Value for APLL and MPLL

Table 7-2 describes the recommended PLL PMS value for APLL and MPLL.

Table 7-2 APLL and MPLL PMS Value

F_{IN} (MHz)	Target F_{OUT} (MHz)	P	M	S	F_{OUT} (MHz)
24	200	3	100	2	200
24	300	4	200	2	300
24	400	3	100	1	400
24	500	3	125	1	500
24	600	4	200	1	600
24	700	3	175	1	700
24	800	3	100	0	800
24	900	4	150	0	900
24	1000	3	125	0	1000
24	1100	6	275	0	1100
24	1200	4	200	0	1200
24	1300	6	325	0	1300
24	1400	3	175	0	1400

NOTE: Although there is an equation for choosing PMS values, we strongly recommend only the values in the above table. If you have to use other values, please contact us.

7.3.2 Recommended PLL PMS Value for EPLL

Table 7-3 describes the recommended PLL PMS value for EPLL.

Table 7-3 EPLL PMS Value

F_{IN} (MHz)	Target F_{OUT} (MHz)	P	M	S	K	F_{OUT} (MHz)
24	90	2	60	3	0	90
24	180	2	60	2	0	180
24	180.6	3	90	2	19661	108.6
24	180.6336	3	90	2	20762	180.6336
24	192	2	64	2	0	192
24	200	3	100	2	0	200
24	400	3	100	1	0	400
24	408	2	68	1	0	408
24	416	3	104	1	0	416

NOTE:

1. K value description "Positive value (Negative value)":
   Positive values is that you should write to EPLLCON/VPLLCON register.
   Negative value is that you can calculate PLL output frequency with it.
2. Although there is an equation for choosing PMS values, we strongly recommend only the values in the above table.
   If you have to use other values, please contact us.
3. You should set K to "0" in EVT0. This restriction will be fixed in EVT1.

7.3.3 Recommended PLL PMS Value for VPLL

Table 7-4 describes the recommended PLL PMS value for VPLL.

Table 7-4 VPLL PMS Value

F_{IN} (MHz)	Target F_{OUT} (MHz)	P	M	S	K	MFR	MRR	SSCG_EN
24	100	3	100	3	0	–	–	0
	160	3	160	3	0	–	–	0
	266	3	133	2	0	–	–	0
	350	3	175	2	0	-	-	0
	440	3	110	1	0	-	-	0

NOTE:

1. Although there is an equation for choosing PMS values, we strongly recommend only the values in the above table. If you have to use other values, please contact us.
2. K value description "Positive value (Negative value)":
   Positive values is that you should write to EPLLCON/VPLLCON register.
   Negative value is that you can calculate PLL output frequency with it.

7.4 Clock Generation

Figure 7-2 and *Figure 7-3*, illustrates the block diagram of the clock generation logic. The clock generator consists of an external crystal clock that is connected to the oscillation amplifier. The PLL converts the incoming low frequency to a high frequency clock that is required by the Exynos 4412 SCP. The clock generator also includes a built-in logic to stabilize the clock frequency for each system reset. The clock requires a specified time for stabilization.

Figure 7-2 and *Figure 7-3* illustrates the two types of clock MUX. Clock MUX in grey color represents glitch-free clock MUX that is free of glitches while changing the clock selection. Clock MUX in white color represents non-glitch-free clock MUX that can suffer from glitches while changing the clock sources. You have to be careful while using each clock MUX.

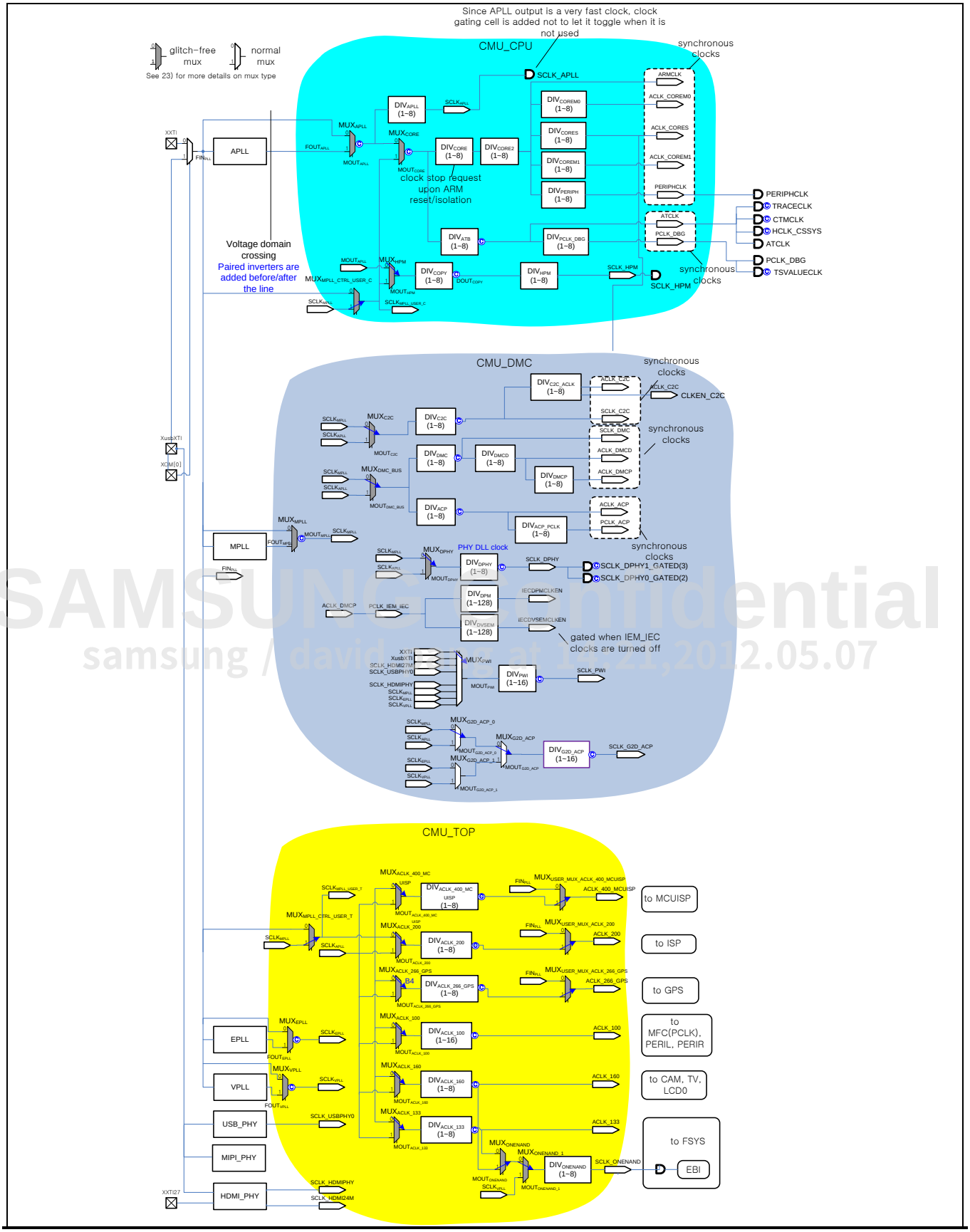
For glitch-free MUX, you should ensure that all clock sources are running while changing the clock selection. If not, it implies that the clock selection process is not complete and it results in clock output having unknown states. The clock MUX status registers are identified with a keyword that starts with CLK_MUX_STAT

For non-glitch-free clock MUX, glitches may occur while changing the clock selection. To prevent glitch signals, we recommend disabling the output of a non-glitch-free MUX before any change of clock selection. After completing the clock change, you can re-enable the output of the non-glitch-free clock MUX. This is done to ensure that there are no glitches resulting due to the clock change selection. The outputs of non-glitch-free MUXES are masked by the clock source mask control registers. The clock source mask control registers are identified with a keyword that starts with CLK_SRC_MASK.

Figure 7-2 and *Figure 7-3* illustrates a clock divider that indicates possible dividing value in parentheses. The dividing values can be changed by clock divider registers during run-time. Some clock dividers have only one dividing value and you are not allowed to change the dividing value.

Figure 7-2 illustrates the Exynos 4412 SCP Clock Generation Circuit (CPU, BUS, DRAM, and ISP Clocks) diagram.

Figure 7-3 illustrates the Exynos 4412 SCP Clock Generation Circuit (Special Clocks) diagram.



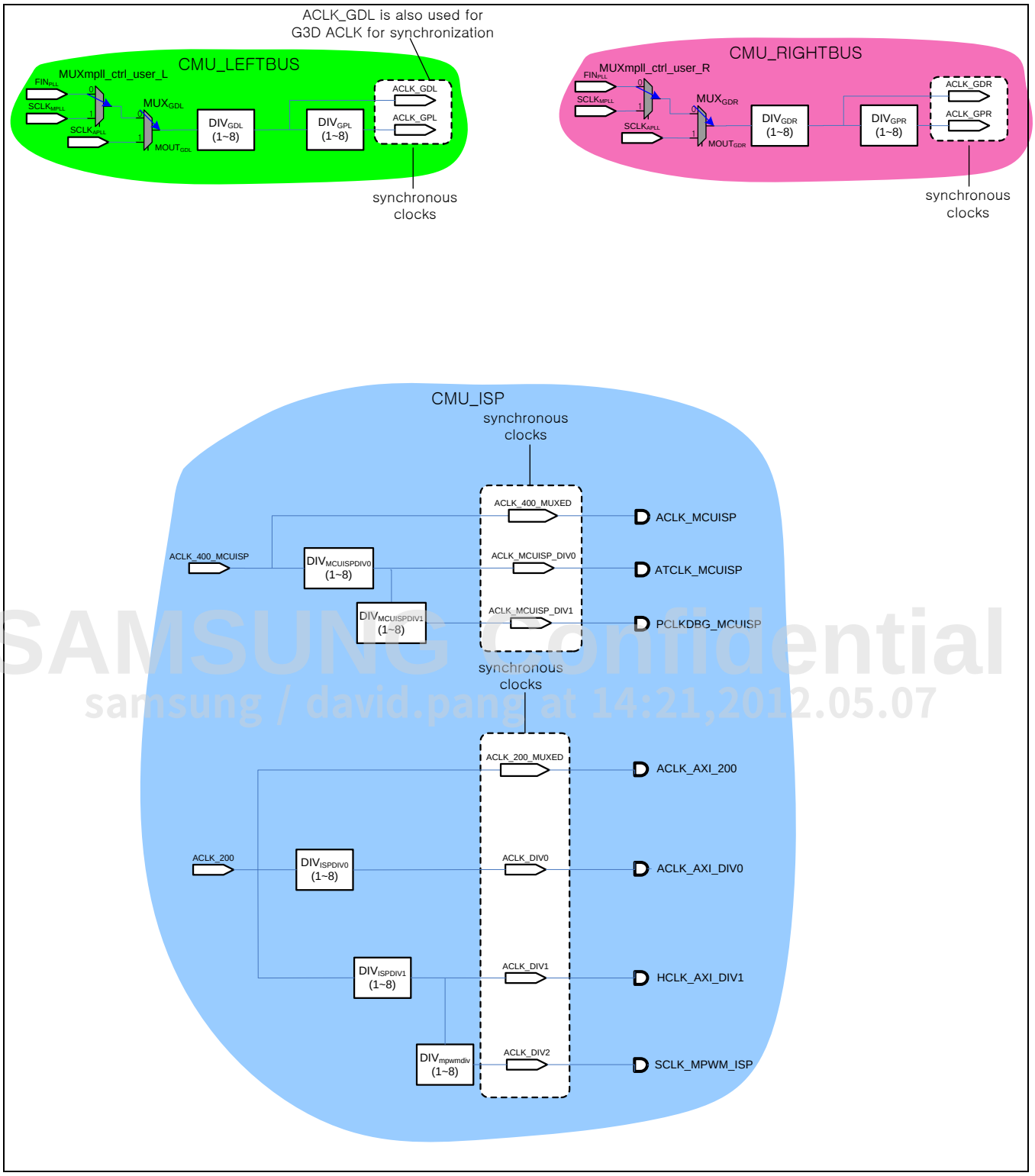
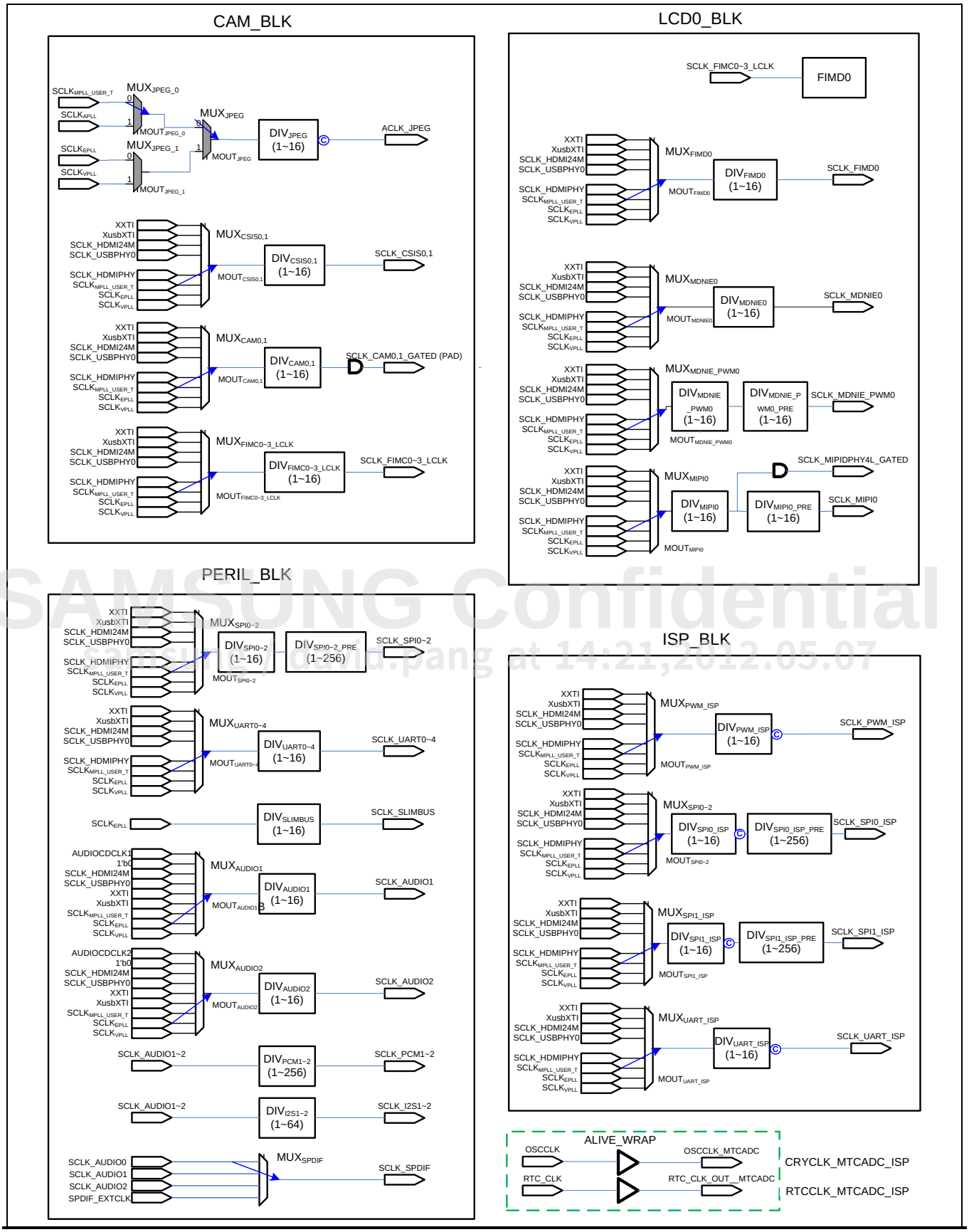


Figure 7-2 Exynos 4412 SCP Clock Generation Circuit (CPU, BUS, DRAM, ISP Clocks)



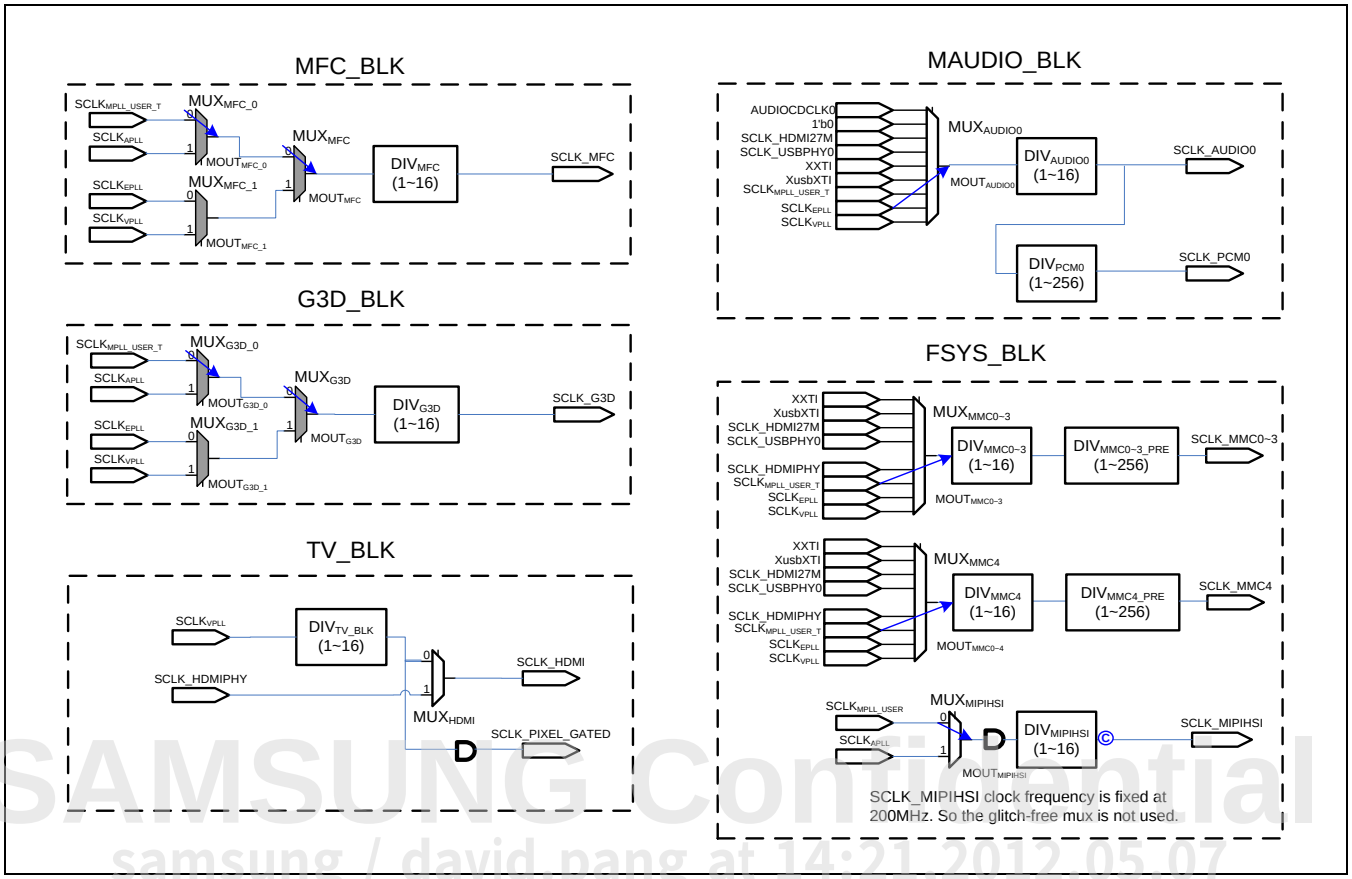


Figure 7-3 Exynos 4412 SCP Clock Generation Circuit (Special Clocks)

NOTE: The SCLKuser_mpll In *Figure 7-3* means SCLKuser_mpll_T.

Caution: In *Figure 7-2* and *Figure 7-3*, the MUX's with grey color are glitch-free. For glitch-free clock MUX, ensure that all clock sources are running while changing the clock selection.
For clock dividers, ensure that input clock is running while changing the divider value.

7.5 Clock Configuration Procedure

The rules for changing the clock configuration are:

- All inputs of a glitch-free MUX should run.
- When a PLL is turned OFF, you should not select the output of PLL.

The basic SFR configuration requires change in system clock divider values that are:

- CLK_DIV_CPU0[31:0] = target value0
- CLK_DIV_DMC0[31:0] = target value1
- CLK_DIV_TOP[31:0] = target value2
- CLK_DIV_LEFTBUS[31:0] = target value3
- CLK_DIV_RIGHTBUS[31:0] = target value4

Change the divider values for special clocks by setting CLK_DIV_XXX SFRs in CMU_TOP

- CLK_DIV_XXX[31:0] = target value

The following sequence shows turn on PLL procedure.

```
Change PLL PMS values
Set PMS values;
                    // Set PDIV, MDIV, and SDIV values (Refer to (A, M, E, V) PLL_CON0 SFRs)
Change other PLL control values
(A,M,E,V)PLL_CON1[31:0]     = target value;
                    // Set K, AFC, MRR, MFR values if necessary (Refer to (A, M, E, V)PLL_CON1 SFRs)
Turn on a PLL
(A,M,E,V)PLL_CON0[31]       = 1;
                    // Turn on a PLL (Refer to (A, M, E, V)PLL_CON0 SFRs)

wait_lock_time;     // Wait until the PLL is locked

MUX_(A, M, E, V)PLL_SEL     = 1;
                    // Select the PLL output clock instead of input reference clock,
                    after PLL output clock is stabilized.
                    (Refer to CLK_SRC_CPU SFR for APLL and MPLL, CLK_SRC_TOP0 for EPLL and VPLL)
                    Once a PLL is turned on, do not turn it off.
```

7.5.1 Clock Gating

Exynos 4412 SCP can disable the clock operation of each IP, if it does not require. This reduces the dynamic power consumption.

The two types of clock gating control register to disable or enable clock operations are:

- Clock gating control register for function blocks
- Clock gating control register for IP

The two clock gating control registers are ANDed to generate the final clock gating enable signal. As a result, if it turns OFF either of the two registers filed, then the resulting clock will stop. For example, to stop the clocks provided to the Mixer module, you should set the CLK_MIXER field in CLK_GATE_IP_TV register to 0 or CLK_TV field in CLK_GATE_BLOCK register to 0. For latter case, all clocks in TV block, not only MIXER clocks, are turned off.

Caution: Ensure that the software does not access the IPs whose clock is gated, as it may cause system failure.

7.5.2 Clock Diving

Whenever clock divider control register is changed, it is recommended to check clock divider status registers before using the new clock output. This guarantees the corresponding divider finishes changing to a new dividing value before its output is used by other modules.

7.6 Special Clock Description

Special Clock Description section describes special clock in Exynos 4412 SCP.

7.6.1 Special Clock Table

Table 7-5 describes the special clocks in Exynos 4412 SCP.

Table 7-5 Special Clocks in Exynos 4412 SCP

Name	Description	Range	Source
SCLK_ONENAND	ONENAND operating clock	160 MHz	ACLK_160, ACLK_133
SCLK_G3D	G3D core operating clock	440MHz	$SCLK_{APLL}$, $SCLK_{MPLL}$, $SCLK_{EPLL}$, $SCLK_{VPLL}$
SCLK_G2D	G2D core operating clock	200MHz	$SCLK_{APLL}$, $SCLK_{MPLL}$, $SCLK_{EPLL}$, $SCLK_{VPLL}$
SCLK_MFC	MFC core operating clock	200 MHz	$SCLK_{APLL}$, $SCLK_{MPLL}$, $SCLK_{EPLL}$, $SCLK_{VPLL}$
SCLK_CAM0, SCLK_CAM1	Reference clock for external CAM device	Range varies in accordance to the CAM specifications.	All possible clock sources
SCLK_CSIS0, SCLK_CSIS1	CSIS operating clock	160 MHz	All possible clock sources
SCLK_FIMC_LCLK0, SCLK_FIMC_LCLK1, SCLK_FIMC_LCLK2, SCLK_FIMC_LCLK3	FIMC core operating clock	160 MHz	All possible clock sources
SCLK_FIMD0	FIMD operating clock	100 MHz	All possible clock sources
SCLK_MDNIE0	MDNIE operating clock	100 MHz	All possible clock sources
SCLK_MDNIE_PWM0	MDNIE PWM clock	100 MHz	All possible clock sources
SCLK_MIPI0	MIPI DSIM clock	100 MHz	All possible clock sources
SCLK_MIPIDPHY4L	MIPI DPHY 4 Lane clock	800 MHz	All possible clock sources
SCLK_HDMI	HDMI LINK clock	148.5 MHz	All possible clock sources
SCLK_PIXEL	HDMI PIXEL clock	148.5 MHz	All possible clock sources
SCLK_SPDIF	SPDIF operating clock	83 MHz	SCLK_AUDIO0, SCLK_AUDIO1, SCLK_AUDIO2
SCLK_MMC0, SCLK_MMC1, SCLK_MMC2, SCLK_MMC3, SCLK_MMC4	HSMMC operating clock	50 MHz	All possible clock sources
SCLK_USBPHY0	USB device clock	48 MHz	USB Device PHY clock out

Name	Description	Range	Source
SCLK_AUDIO0, SCLK_AUDIO1, SCLK_AUDIO2	AUDIO operating clock (I2S)	100 MHz	All possible clock sources, AUDIOCDCLKx
SCLK_PCM0, SCLK_PCM1, SCLK_PCM2	AUDIO operating clock (PCM)	5 MHz	SCLK_AUDIO0, SCLK_AUDIO1, SCLK_AUDIO2
SCLK_PWI	IEM APC operating clock	6 to 30 MHz	All possible clock sources
SCLK_KEY	KEY I/F or TSADC filter clock (fixed clock)	24 MHz	XXTI, XUSBXTI
SCLK_SPI0, SCLK_SPI1, SCLK_SPI2	SPI operating clock	100 MHz	All possible clock sources
SCLK_UART0, SCLK_UART1, SCLK_UART2, SCLK_UART3, SCLK_UART4	UART operating clock	200 MHz	All possible clock sources
SCLK_SLIMBUS	SLIMBUS clock	25 MHz	$SCLK_{EPLL}$
ACLK_JPEG	JPEG core operating clock	160 MHz	$SCLK_{APLL}$, $SCLK_{MPLL}$, $SCLK_{EPLL}$, $SCLK_{VPLL}$
SCLK_PWM_ISP	PWM_ISP operating clock	66 MHz	All possible clock sources
SCLK_SPI0_ISP	SPI0_ISP operating clock	100 MHz	All possible clock sources
SCLK_SPI1_ISP	SPI1_ISP operating clock	100 MHz	All possible clock sources
SCLK_UART_ISP	UART_ISP operating clock	66 MHz	All possible clock sources
SCLK_MIPIHSI	MIPIHSI core operating clock	200 MHz	$SCLK_{APLL}$, $SCLK_{MPLL}$,

- All possible clock sources include XXTI, XUSBXTI, SCLK_HDMI24M, SCLK_USBPHY, SCLK_HDMIPHY, $SCLK_{MPLL}$, $SCLK_{EPLL}$, and $SCLK_{VPLL}$.
- XXTI and XUSBXTI refer to external crystal.
- SCLK_USBPHY refers to USB PHY 48 MHz output clock.
- SCLK_HDMI24M refers to HDMI PHY (24 MHz reference clock for XUSBXTI) output.
- SCLK_HDMIPHY refers to HDMI PHY (PIXEL_CLKO) output clock.
- $SCLK_{MPLL}$, $SCLK_{EPLL}$, and $SCLK_{VPLL}$ refer to the output clock of MPLL, EPLL, and VPLL, respectively.

Table 7-6 describes the I/O clocks in Exynos 4412 SCP.

Table 7-6 I/O Clocks in Exynos 4412 SCP

Name	I/O	Pad	GPIO Function	Range	Description
IOCLK_AC97	Input	Xi2s1SCLK	Func2: AC97BITCLK	12.288 MHz	AC97 Bit Clock
IOCLK_I2S0, IOCLK_I2S1, IOCLK_I2S2	Input	Xi2s0CDCLK Xi2s1CDCLK Xpcm2EXTCLK	Func0: I2S_0_CDCLK Func0: I2S_1_CDCLK Func2: I2S_2_CDCLK	83.4 MHz	I2S CODEC Clock
IOCLK_PCM0, IOCLK_PCM1, IOCLK_PCM2	Input	Xi2s0CDCLK Xi2s1CDCLK Xpcm2EXTCLK	Func1: PCM_0_EXTCLK Func1: PCM_1_EXTCLK Func0: PCM_2_EXTCLK	83.4 MHz	PCM CODEC Clock
IOCLK_SPDIF	Input	Xpcm2EXTCLK	Func1: SPDIF_EXTCLK	36.864 MHz	SPDIF Input Clock

7.7 CLKOUT

You can use the XCLKOUT port to monitor certain clocks in Exynos 4412 SCP. The six CMUs in Exynos 4412 SCP contain the CLKOUT control logic. If necessary, you can select and divide one of the clocks in the CMU. It generates CLKOUT signal from each CMU and feeds this into the power management unit. It is then muxed with CLKOUT signals and XXTI, XUSBXTI, RTC_TICK_SRC, and RTCCLK.

Figure 7-4 illustrates the CLKOUT control logic in Exynos 4412 SCP .

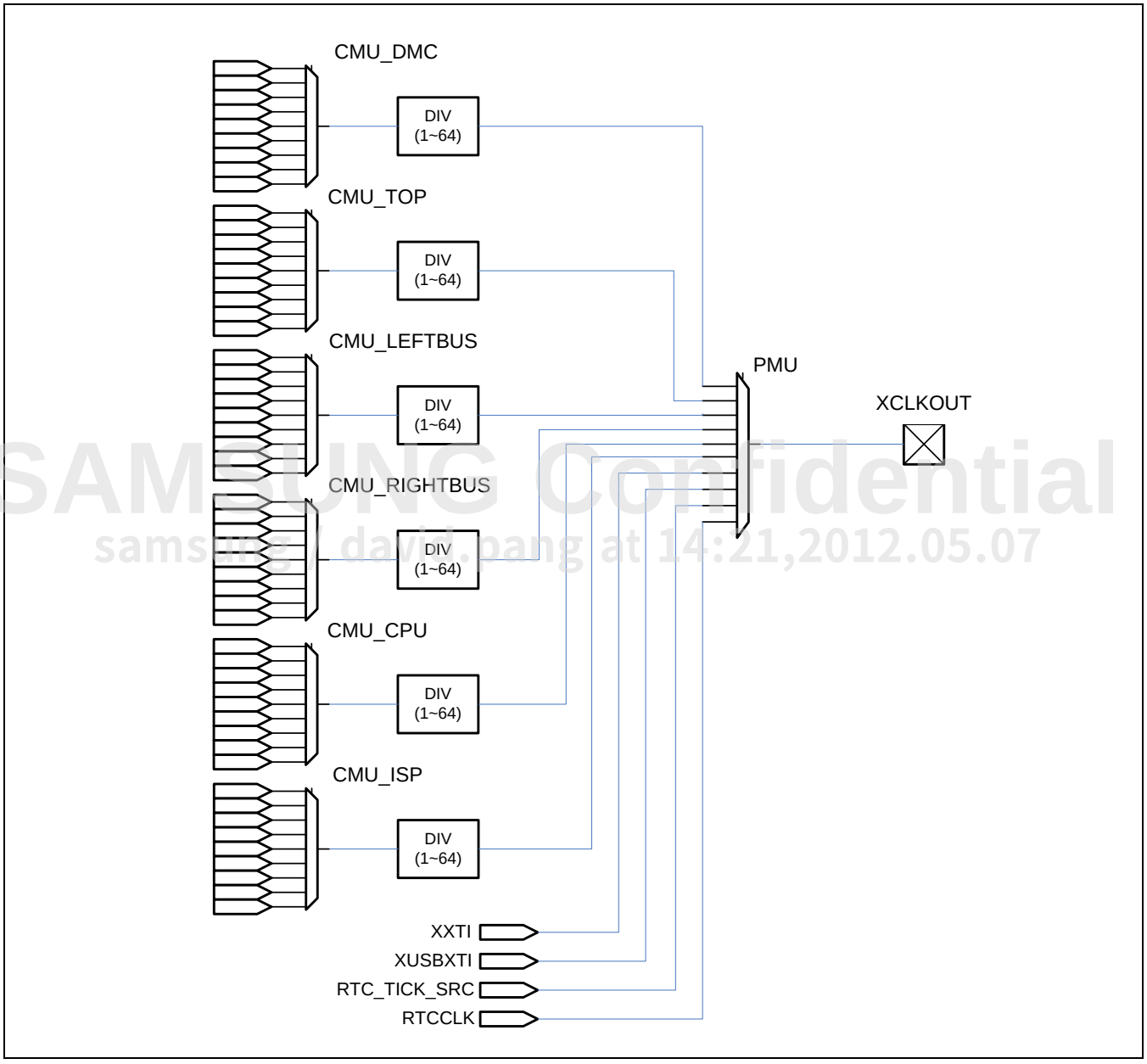


Figure 7-4 Exynos 4412 SCP CLKOUT Control Logic

Table 7-7 describes the CLKOUT input clock selection information.

Table 7-7 CLKOUT Input Clock Selection Information

No.	CMU_CPU	CMU_DMC	CMU_TOP	CMU_RIGHTBUS	CMU_LEFTBUS	CMU_ISP	PMU
0	APLL_ FOUT/2	ACLK_DMCD	EPLL_FOUT	SCLK_MPLL/2	SCLK_MPLL/2	ACLK_ MCUISP	CMU_ DMC
1	Reserved	ACLK_DMCP	VPLL_FOUT	SCLK_APLL/2	SCLK_APLL/2	PCLKDBG _MCUISP	CMU_ TOP
2	Reserved	ACLK_ACP	SCLK_ HDMI24M	ACLK_GDR	ACLK_GDL	ACLK_DIV0	CMU_ LEFTBUS
3	Reserved	PCLK_ACP	SCLK_ USBPHY0	ACLK_GPR	ACLK_GPL	ACLK_DIV1	CMU_ RIGHTBUS
4	ARMCLK/2	SCLK_DMC	Reserved	–	–	SCLK_ MPWM_ISP	CMU_CPU
5	ACLK_ COREM0	SCLK_DPHY	SCLK_ HDMIPHY	–	–	–	XXTI
6	ACLK_ COREM1	MPLL_ FOUT/2	AUDIOCDC LK0	–	–	–	XUSBXTI
7	ACLK_ CORES	SCLK_PWI	AUDIOCDC LK1	–	–	–	RTC_TICK _SRC
8	ATCLK	–	AUDIOCDC LK2	–	–	–	RTCCLK
9	PERIPHCL K	SCLK_C2C	SPDIF_ EXTCLK	–	–	–	–
10	PCLK_ DBG	ACLK_C2C	ACLK_160	–	–	–	–
11	SCLK_ HPM	–	ACLK_133	–	–	–	–
12	–	–	ACLK_200	–	–	–	–
13	–	–	ACLK_100	–	–	–	–
14	–	–	SCLK_MFC	–	–	–	–
15	–	–	SCLK_G3D	–	–	–	–
16	–	–	ACLK_400_ MCUISP	–	–	–	–
17	–	–	CAM_A_ PCLK	–	–	–	–
18	–	–	CAM_B_ PCLK	–	–	–	–
19	–	–	S_RXBYTE CLKHS0_2L	–	–	–	–
20	–	–	S_RXBYTE CLKHS0_4L	–	–	–	–
21	–	–	RX_HALF_	–	–	–	–

No.	CMU_CPU	CMU_DMC	CMU_TOP	CMU_RIGHTBUS	CMU_LEFTBUS	CMU_ISP	PMU
			BYTE_CLK_ CSIS0				
22	–	–	RX_HALF_ BYTE_CLK_ CSIS1	–	–	–	–
23	–	–	SCLK_JPEG	–	–	–	–
24	–	–	SCLK_PWM _ISP	–	–	–	–
25	–	–	SCLK_SPI0 _ISP	–	–	–	–
26	–	–	SCLK_SPI1 _ISP	–	–	–	–
27	–	–	SCLK_ UART_ISP	–	–	–	–
28	–	–	SCLK_ MIPIHSI	–	–	–	–
29	–	–	SCLK_HDMI	–	–	–	–
30	–	–	SCLK_ FIMD0	–	–	–	–
31	–	–	SCLK_ PCM0	–	–	–	–

7.8 I/O Description

Table 7-8 describes the I/O.

Table 7-8 I/O Description

Signal	I/O	Description	Pad	Type
XXTI	Input	External oscillator pad	XXTI	Dedicated
XUSBXTI	Input	Input pad for crystal	XUSBXTI	Dedicated
XUSBXTO	Output	Output pad for crystal	XUSBXTO	Dedicated
EPLLFILTER	Input/Output	Pad for EPLL loop filter capacitance	XEPLLFILTER	Dedicated
VPLLFILTER	Input/Output	Pad for VPLL loop filter capacitance	XVPLLFILTER	Dedicated
XCLKOUT	Output	Clock out pad	XCLKOUT	Dedicated

7.9 Register Description

The clock controller controls PLLs and clock generation units. This section describes the usage of Special Functional Registers (SFRs) in the clock controller. Do not change any reserved area. Any change in the reserved area leads to an unexpected behavior.

The address map of Exynos 4412 SCP clock controller consists of six CMUs. They are, CMU_LEFTBUS, CMU_RIGHTBUS, CMU_TOP, CMU_DMC, CMU_CPU, and CMU_ISP. Each CMU uses an address space of 16 KB for SFRs. The internal structure of address space for each CMU is similar for all CMUs.

The six categories into which the address space is divided are:

- Use 0x000 to 0x1FF for PLL control : PLL lock time and control
- Use 0x200 to 0x4FF for MUX control : MUX selection, output masking, and status
- Use 0x500 to 0x6FF for clock division : Divider ratio and status
- 0x700 to 0x8FF is reserved and you are not allowed to access the region.
- Use 0x900 to 0x9FF for clock gating control : Clock gating of IPs and function blocks
- Use 0xA00 to 0xAFF for CLKOUT : CLKOUT input clock selection and divider ratio

NOTE: The CLK_GATE_IP_XXX registers in the CMU_LEFTBUS and CMU_RIGHTBUS are located at 0x800. Additionally, some CMUs use addresses beyond 0xAFF for other functions such as CPU control functions in CMU_CPU. Refer to register description for more information.

In *Figure 7-5*, XXX in the register name shall be replaced with the function block name, i.e., LEFTBUS, RIGHTBUS, TOP, CAM, TV, MFC, G3D, IMAGE, LCD0, LCD1, MAUDIO, FSYS, PERIL, and PERIR.

Figure 7-5 illustrates the Exynos 4412 SCP clock controller address map.

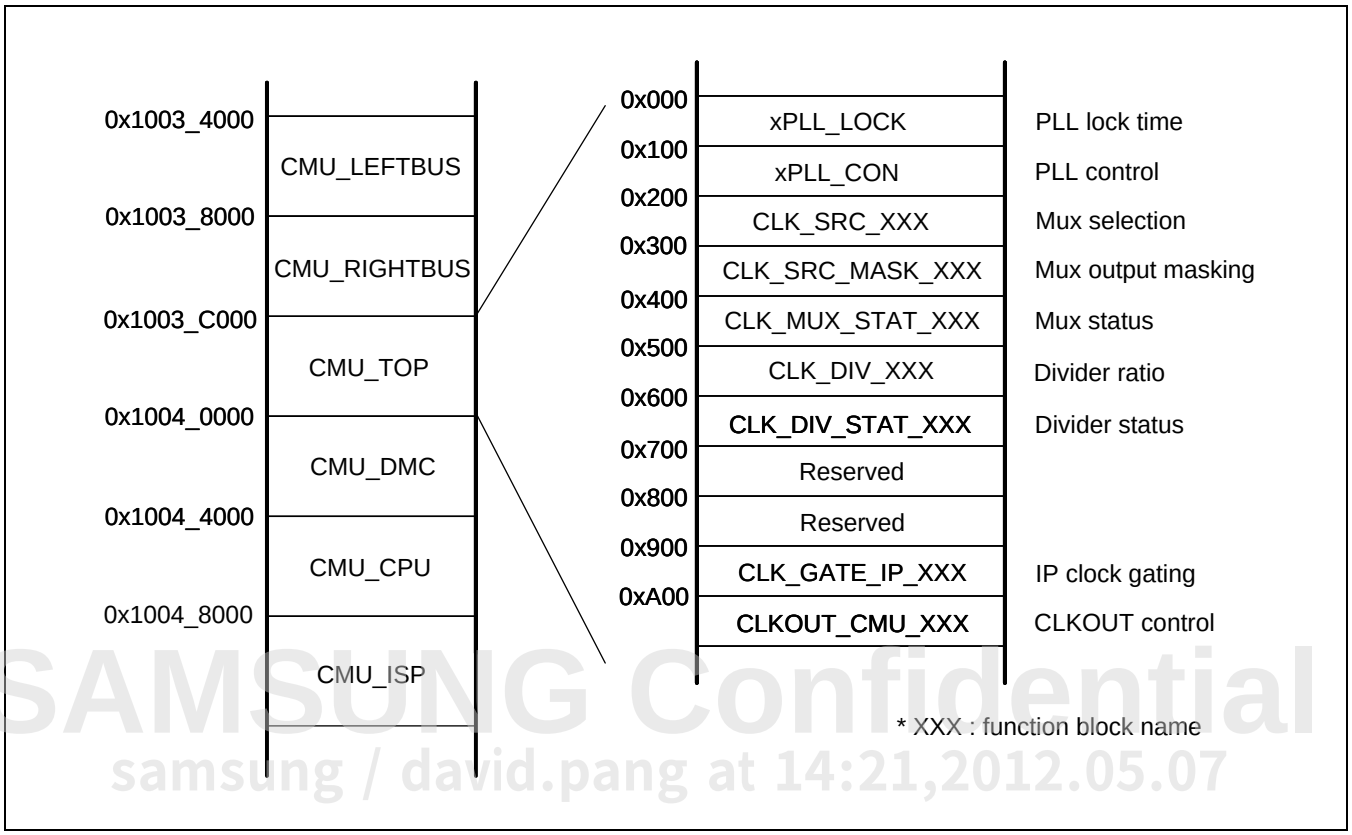


Figure 7-5 Exynos 4412 SCP Clock Controller Address Map

7.9.1 Register Map Summary

- Base Address: 0x1003_0000

Register	Offset	Description	Reset Value
CLK_SRC_LEFTBUS	0x4200	Selects clock source for CMU_LEFTBUS	0x0000_0000
RSVD	0x4204 to 0x43FF	Reserved	Undefined
CLK_MUX_STAT_LEFTBUS	0x4400	Clock MUX status for CMU_LEFTBUS	0x0000_0011
RSVD	0x4404 to 0x44FF	Reserved	Undefined
CLK_DIV_LEFTBUS	0x4500	Sets clock divider ratio for CMU_LEFTBUS	0x0000_0000
RSVD	0x4504 to 0x45FF	Reserved	Undefined
CLK_DIV_STAT_LEFTBUS	0x4600	Clock divider status for CMU_LEFTBUS	0x0000_0000
RSVD	0x4604 to 0x47FC	Reserved	Undefined
CLK_GATE_IP_LEFTBUS	0x4800	Control IP clock gating for LEFTBUS_BLK	0xFFFF_FFFF
RSVD	0x4804 to 0x492C	Reserved	Undefined
CLK_GATE_IP_IMAGE	0x4930	Control IP clock gating for IMAGE_SS	0xFFFF_FFFF
RSVD	0x4934 to 0x49FC	Reserved	Undefined
CLKOUT_CMU_LEFTBUS	0x4A00	CLKOUT control register	0x0001_0000
CLKOUT_CMU_LEFTBUS_DIV_STAT	0x4A04	Clock divider status for CLKOUT	0x0000_0000
RSVD	0x4A08 to 0x81FF	Reserved	Undefined
CLK_SRC_RIGHTBUS	0x8200	Selects clock source for CMU_RIGHTBUS	0x0000_0000
RSVD	0x8204 to 0x83FF	Reserved	Undefined
CLK_MUX_STAT_RIGHTBUS	0x8400	Clock MUX status for CMU_RIGHTBUS	0x0000_0011
RSVD	0x8404 to 0x84FF	Reserved	Undefined
CLK_DIV_RIGHTBUS	0x8500	Sets clock divider ratio for CMU_RIGHTBUS	0x0000_0000
RSVD	0x8504 to 0x85FF	Reserved	Undefined
CLK_DIV_STAT_RIGHTBUS	0x8600	Clock divider status for CMU_RIGHTBUS	0x0000_0000
RSVD	0x8604 to 0x87FC	Reserved	Undefined
CLK_GATE_IP_RIGHTBUS	0x8800	Control IP clock gating for RIGHTBUS_BLK	0xFFFF_FFFF
RSVD	0x8804 to	Reserved	Undefined

Register	Offset	Description	Reset Value
	0x895C		
CLK_GATE_IP_PERIR	0x8960	Controls IP clock gating for PERIR_S	0xFFFF_FFFF
RSVD	0x8964 to 0x89FC	Reserved	Undefined
CLKOUT_CMU_RIGHTBUS	0x8A00	CLKOUT control register	0x0001_0000
CLKOUT_CMU_RIGHTBUS _DIV_STAT	0x8A04	Clock divider status for CLKOUT	0x0000_0000
RSVD	0x8A08 to 0xC00F	Reserved	Undefined
EPLL_LOCK	0xC010	Controls PLL locking period for EPLL	0x0000_0FFF
RSVD	0xC014 to 0xC01F	Reserved	Undefined
VPLL_LOCK	0xC020	Controls PLL locking period for VPLL	0x0000_0FFF
RSVD	0xC024 to 0xC10F	Reserved	Undefined
EPLL_CON0	0xC110	Controls PLL output frequency for EPLL	0x0060_0302
EPLL_CON1	0xC114	Controls PLL output frequency for EPLL	0x6601_0000
EPLL_CON2	0xC118	Controls PLL output frequency for EPLL	0x0000_0080
RSVD	0xC11C	Reserved	Undefined
VPLL_CON0	0xC120	Controls PLL output frequency for VPLL	0x006F_0302
VPLL_CON1	0xC124	Controls PLL output frequency for VPLL	0x6601_6000
VPLL_CON2	0xC128	Controls PLL output frequency for VPLL	0x0000_0080
RSVD	0xC12C to 0xC20C	Reserved	Undefined
CLK_SRC_TOP0	0xC210	Selects clock source for CMU_TOP0	0x0000_0000
CLK_SRC_TOP1	0xC214	Selects clock source for CMU_TOP1	0x0000_0000
RSVD	0xC218 to 0xC21F	Reserved	Undefined
CLK_SRC_CAM0	0xC220	Selects clock source for CAM_BLK	0x1111_1111
CLK_SRC_TV	0xC224	Selects clock source for TV_BLK	0x0000_0000
CLK_SRC_MFC	0xC228	Selects clock source for MFC_BLK	0x0000_0000
CLK_SRC_G3D	0xC22C	Selects clock source for G3D_BLK	0x0000_0000
RSVD	0xC230	Reserved	Undefined
CLK_SRC_LCD	0xC234	Selects clock source for LCD_BLK	0x0000_1111
CLK_SRC_ISP	0xC238	Selects clock source for ISP_BLK	0x0000_1111
CLK_SRC_MAUDIO	0xC23C	Selects clock source for AUDIO_BLK	0x0000_0005
CLK_SRC_FSYS	0xC240	Selects clock source for FSYS_BLK	0x0001_1111
RSVD	0xC244 to 0xC24C	Reserved	Undefined

Register	Offset	Description	Reset Value
CLK_SRC_PERIL0	0xC250	Selects clock source for connectivity IPs	0x0001_1111
CLK_SRC_PERIL1	0xC254	Selects clock source for connectivity IPs	0x0111_0055
CLK_SRC_CAM1	0xC258	Selects clock source for CAM_BLK	0x0000_0000
RSVD	0xC25C to 0xC31F	Reserved	Undefined
CLK_SRC_MASK_CAM0	0xC320	Clock source mask for CAM_BLK	0x1111_1111
CLK_SRC_MASK_TV	0xC324	Clock source mask for TV_BLK	0x0000_0111
RSVD	0xC328 to 0xC333	Reserved	Undefined
CLK_SRC_MASK_LCD	0xC334	Clock source mask for LCD_BLK	0x0000_1111
CLK_SRC_MASK_ISP	0xC338	Clock source mask for ISP_BLK	0x0000_1111
CLK_SRC_MASK_MAUDIO	0xC33C	Clock source mask for AUDIO_BLK	0x0000_0001
CLK_SRC_MASK_FSYS	0xC340	Clock source mask for FSYS_BLK	0x0101_1111
RSVD	0xC344 to 0xC34F	Reserved	Undefined
CLK_SRC_MASK_PERIL0	0xC350	Clock source mask for PERIL_BLK	0x0001_1111
CLK_SRC_MASK_PERIL1	0xC354	Clock source mask for PERIL_BLK	0x0111_0111
RSVD	0xC358 to 0xC40F	Reserved	Undefined
CLK_MUX_STAT_TOP0	0xC410	Clock MUX status for CMU_TOP	0x1111_1111
CLK_MUX_STAT_TOP1	0xC414	Clock MUX status for CMU_TOP	0x0111_1110
RSVD	0xC418 to 0xC427	Reserved	Undefined
CLK_MUX_STAT_MFC	0xC428	Clock MUX status for MFC_BLK	0x0000_0111
CLK_MUX_STAT_G3D	0xC42C	Clock MUX status for G3D_BLK	0x0000_0111
RSVD	0xC430 to 0xC454	Reserved	Undefined
CLK_MUX_STAT_CAM1	0xC458	Clock MUX status for CAM_BLK	0x0000_0111
RSVD	0xC45C to 0xC50C	Reserved	Undefined
CLK_DIV_TOP	0xC510	Sets clock divider ratio for CMU_TOP	0x0000_0000
RSVD	0xC514 to 0xC51F	Reserved	Undefined
CLK_DIV_CAM0	0xC520	Sets clock divider ratio for CAM_BLK	0x0000_0000
CLK_DIV_TV	0xC524	Sets clock divider ratio for TV_BLK	0x0000_0000
CLK_DIV_MFC	0xC528	Sets clock divider ratio for MFC_BLK	0x0000_0000
CLK_DIV_G3D	0xC52C	Sets clock divider ratio for G3D_BLK	0x0000_0000
RSVD	0xC530	Reserved	Undefined
CLK_DIV_LCD	0xC534	Sets clock divider ratio for LCD_BLK	0x0070_0000

Register	Offset	Description	Reset Value
CLK_DIV_ISP	0xC538	Sets clock divider ratio for ISP_BLK	0x0000_0000
CLK_DIV_MAUDIO	0xC53C	Sets clock divider ratio for AUDIO_BLK	0x0000_0000
CLK_DIV_FSYS0	0xC540	Sets clock divider ratio for FSYS_BLK	0x00B0_0000
CLK_DIV_FSYS1	0xC544	Sets clock divider ratio for FSYS_BLK	0x0000_0000
CLK_DIV_FSYS2	0xC548	Sets clock divider ratio for FSYS_BLK	0x0000_0000
CLK_DIV_FSYS3	0xC54C	Sets clock divider ratio for FSYS_BLK	0x0000_0000
CLK_DIV_PERIL0	0xC550	Sets clock divider ratio for PERIL_BLK	0x0000_0000
CLK_DIV_PERIL1	0xC554	Sets clock divider ratio for PERIL_BLK	0x0000_0000
CLK_DIV_PERIL2	0xC558	Sets clock divider ratio for PERIL_BLK	0x0000_0000
CLK_DIV_PERIL3	0xC55C	Sets clock divider ratio for PERIL_BLK	0x0000_0000
CLK_DIV_PERIL4	0xC560	Sets clock divider ratio for PERIL_BLK	0x0000_0000
CLK_DIV_PERIL5	0xC564	Sets clock divider ratio for PERIL_BLK	0x0000_0000
CLK_DIV_CAM1	0xC568	Sets clock divider ratio for CAM_BLK	0x0000_0000
RSVD	0xC56C to 0xC57C	Reserved	Undefined
CLKDIV2_RATIO	0xC580	Sets PCLK divider ratio in FSYS, CAM, LCD, TV, and GPS block	0x0111_1111
RSVD	0xC584 to 0xC60F	Reserved	Undefined
CLK_DIV_STAT_TOP	0xC610	Clock divider status for CMU_TOP	0x0000_0000
RSVD	0xC614 to 0xC61F	Reserved	Undefined
CLK_DIV_STAT_CAM0	0xC620	Clock divider status for CAM_BLK	0x0000_0000
CLK_DIV_STAT_TV	0xC624	Clock divider status for TV_BLK	0x0000_0000
CLK_DIV_STAT_MFC	0xC628	Clock divider status for MFC_BLK	0x0000_0000
CLK_DIV_STAT_G3D	0xC62C	Clock divider status for G3D_BLK	0x0000_0000
RSVD	0xC630	Reserved	Undefined
CLK_DIV_STAT_LCD	0xC634	Clock divider status for LCD_BLK	0x0000_0000
CLK_DIV_STAT_ISP	0xC638	Clock divider status for ISP_BLK	0x0000_0000
CLK_DIV_STAT_MAUDIO	0xC63C	Clock divider status for AUDIO_BLK	0x0000_0000
CLK_DIV_STAT_FSYS0	0xC640	Clock divider status for FSYS_BLK	0x0000_0000
CLK_DIV_STAT_FSYS1	0xC644	Clock divider status for FSYS_BLK	0x0000_0000
CLK_DIV_STAT_FSYS2	0xC648	Clock divider status for FSYS_BLK	0x0000_0000
CLK_DIV_STAT_FSYS3	0xC64C	Clock divider status for FSYS_BLK	0x0000_0000
CLK_DIV_STAT_PERIL0	0xC650	Clock divider status for PERIL_BLK	0x0000_0000
CLK_DIV_STAT_PERIL1	0xC654	Clock divider status for PERIL_BLK	0x0000_0000
CLK_DIV_STAT_PERIL2	0xC658	Clock divider status for PERIL_BLK	0x0000_0000
CLK_DIV_STAT_PERIL3	0xC65C	Clock divider status for PERIL_BLK	0x0000_0000

Register	Offset	Description	Reset Value
CLK_DIV_STAT_PERIL4	0xC660	Clock divider status for PERIL_BLK	0x0000_0000
CLK_DIV_STAT_PERIL5	0xC664	Clock divider status for PERIL_BLK	0x0000_0000
CLK_DIV_STAT_CAM1	0xC668	Clock divider status for CAM_BLK	0x0000_0000
RSVD	0xC66C to 0xC67C	Reserved	Undefined
CLKDIV2_STAT	0xC680	PCLK divider status for FSYS, CAM, LCD, and TV block	0x0000_0000
RSVD	0xC684 to 0xC740	Reserved	Undefined
CLK_GATE_BUS_FSYS1	0xC744	Control gating of AXI/AHB/APB clock for FSYS_BLK	0xFFFF_FFFF
RSVD	0xC748 to 0xC91F	Reserved	Undefined
CLK_GATE_IP_CAM	0xC920	Controls IP clock gating for CAM_BLK	0xFFFF_FFFF
CLK_GATE_IP_TV	0xC924	Controls IP clock gating for TV_BLK	0xFFFF_FFFF
CLK_GATE_IP_MFC	0xC928	Controls IP clock gating for MFC_BLK	0xFFFF_FFFF
CLK_GATE_IP_G3D	0xC92C	Controls IP clock gating for G3D_BLK	0xFFFF_FFFF
RSVD	0xC930	Reserved	Undefined
CLK_GATE_IP_LCD	0xC934	Controls IP clock gating for LCD_BLK	0xFFFF_FFFF
CLK_GATE_IP_ISP	0xC938	Controls IP clock gating for ISP_BLK	0xFFFF_FFFF
CLK_GATE_IP_FSYS	0xC940	Controls IP clock gating for FSYS_BLK	0xFFFF_FFFF
RSVD	0xC944 to 0xC948	Reserved	Undefined
CLK_GATE_IP_GPS	0xC94C	Controls IP clock gating for GPS_BLK	0xFFFF_FFFF
CLK_GATE_IP_PERIL	0xC950	Controls IP clock gating for PERIL_BLK	0xFFFF_FFFF
RSVD	0xC954 to 0xC96F	Reserved	Undefined
CLK_GATE_BLOCK	0xC970	Clock gating control block	0xFFFF_FFFF
RSVD	0xC974 to 0xC9FF	Reserved	Undefined
CLKOUT_CMU_TOP	0xCA00	CLKOUT control register	0x0001_0000
CLKOUT_CMU_TOP_ DIV_STAT	0xCA04	Clock divider status for CLKOUT	0x0000_0000
RSVD	0xCA08 to 0x0004	Reserved	Undefined

- Base Address 0x10040000

Register	Offset	Description	Reset Value
MPLL_LOCK	0x0008	Controls PLL locking period for MPLL	0x0000_0FFF
RSVD	0x000C to 0x0104	Reserved	Undefined
MPLL_CON0	0x0108	Controls PLL output frequency for MPLL	0x0064_0300
MPLL_CON1	0x010C	Controls PLL AFC	0x0080_3800
RSVD	0x0110 to 0x01FC	Reserved	Undefined
CLK_SRC_DMC	0x0200	Selects clock source for CMU_DMC	0x0001_0000
RSVD	0x0204 to 0x02FF	Reserved	Undefined
CLK_SRC_MASK_DMC	0x0300	Clock source mask for DMC_BLK	0x0001_0000
RSVD	0x0304 to 0x03FF	Reserved	Undefined
CLK_MUX_STAT_DMC	0x0400	Clock MUX status for CMU_DMC	0x1110_1111
RSVD	0x0404 to 0x04FF	Reserved	Undefined
CLK_DIV_DMC0	0x0500	Sets clock divider ratio for CMU_DMC	0x0000_0000
CLK_DIV_DMC1	0x0504	Sets clock divider ratio for CMU_DMC	0x0000_1000
RSVD	0x0508 to 0x05FF	Reserved	Undefined
CLK_DIV_STAT_DMC0	0x0600	Clock divider status for CMU_DMC	0x0000_0000
CLK_DIV_STAT_DMC1	0x0604	Clock divider status for CMU_DMC	0x0000_0000
RSVD	0x0608 to 0x06FC	Reserved	Undefined
CLK_GATE_BUS_DMC0	0x0700	Control gating of AXI clock for DMC_BLK	0xFFFF_FFFF
CLK_GATE_BUS_DMC1	0x0704	Control gating of APB clock for DMC_BLK	0xFFFF_FFFF
RSVD	0x0708 to 0x08FC	Reserved	Undefined
CLK_GATE_IP_DMC0	0x0900	Control IP clock gating for DMC_BLK	0xFFFF_FFFF
CLK_GATE_IP_DMC1	0x0904	Control IP clock gating for DMC_BLK	0xFFFF_FFFF
RSVD	0x0908 to 0x09FF	Reserved	Undefined
CLKOUT_CMU_DMC	0x0A00	CLKOUT control register	0x0001_0000
CLKOUT_CMU_DMC_ DIV_STAT	0x0A04	Clock divider status for CLKOUT	0x0000_0000
RSVD	0x0A08 to 0x0FFF	Reserved	Undefined
DCGIDX_MAP0	0x1000	DCG index map0	0xFFFF_FFFF
DCGIDX_MAP1	0x1004	DCG index map1	0xFFFF_FFFF

Register	Offset	Description	Reset Value
DCGIDX_MAP2	0x1008	DCG index map2	0xFFFF_FFFF
RSVD	0x100C to 0x101F	Reserved	Undefined
DCGPERF_MAP0	0x1020	DCG performance map0	0xFFFF_FFFF
DCGPERF_MAP1	0x1024	DCG performance map1	0xFFFF_FFFF
RSVD	0x1028 to 0x103F	Reserved	Undefined
DVCIDX_MAP	0x1040	DVC index map	0x00FF_FFFF
RSVD	0x1044 to 0x105F	Reserved	Undefined
FREQ_CPU	0x1060	Maximum frequency of CPU	0x0000_0000
FREQ_DPM	0x1064	Frequency of DPM	0x0000_0000
RSVD	0x1068 to 0x107F	Reserved	Undefined
DVSEMCLK_EN	0x1080	DVS emulation clock enable	0x0000_0000
MAXPERF	0x1084	Maximum performance enable	0x0000_0000
RSVD	0x1088 to 0x1090	Reserved	Undefined
DMC_PAUSE_CTRL	0x1094	Pause function of DREX2 for DVFS	0x0000_0000
DDRPHY_LOCK_CTRL	0x1098	DDRPHY DLL lock control register when C2C is enabled	0x0000_0000
C2C_STATE	0x109C	Current state of C2C SEC FSM	0x0000_0000
RSVD	0x10A0 to 0x3FFC	Reserved	Undefined
APLL_LOCK	0x4000	Control PLL locking period for APLL	0x0000_0FFF
RSVD	0x4004 to 0x40FC	Reserved	Undefined
APLL_CON0	0x4100	Control PLL output frequency for APLL	0x0064_0300
APLL_CON1	0x4104	Control PLL AFC	0x0080_3800
RSVD	0x4108 to 0x41FC	Reserved	Undefined
CLK_SRC_CPU	0x4200	Selects clock source for CMU_CPU	0x0000_0000
RSVD	0x4204 to 0x43FF	Reserved	Undefined
CLK_MUX_STAT_CPU	0x4400	Clock MUX status for CMU_CPU	0x0011_0101
RSVD	0x4404 to 0x44FF	Reserved	Undefined
CLK_DIV_CPU0	0x4500	Sets clock divider ratio for CMU_CPU	0x0000_0000
CLK_DIV_CPU1	0x4504	Sets clock divider ratio for CMU_CPU	0x0000_0000
RSVD	0x4508 to	Reserved	Undefined

Register	Offset	Description	Reset Value
	0x45FF		
CLK_DIV_STAT_CPU0	0x4600	Clock divider status for CMU_CPU	0x0000_0000
CLK_DIV_STAT_CPU1	0x4604	Clock divider status for CMU_CPU	0x0000_0000
RSVD	0x4608 to 0x48FF	Reserved	Undefined
CLK_GATE_IP_CPU	0x4900	Controls IP clock gating for CMU_CPU	0xFFFF_FFFF
RSVD	0x4904 to 0x49FF	Reserved	Undefined
CLKOUT_CMU_CPU	0x4A00	CLKOUT control register	0x0001_0000
CLKOUT_CMU_CPU_ DIV_STAT	0x4A04	Clock divider status for CLKOUT	0x0000_0000
RSVD	0x4A08 to 0x4FFF	Reserved	Undefined
ARMCLK_STOPCTRL	0x5000	ARM clock stop control register SCLK_APLL counts the number of cycles.	0x0404_0404
ATCLK_STOPCTRL	0x5004	ATCLK stop control register SCLK_APLL counts the number of cycles.	0x0000_0404
RSVD	0x500C to 0x501C	Reserved	Undefined
PWR_CTRL	0x5020	Power control register	0x0000_04FF
PWR_CTRL2	0x5024	Power control register	0x0000_0000
RSVD	0x5028 to 0x53FC	Reserved	Undefined
L2_STATUS	0x5400	L2 cache status register	0x0000_0000
RSVD	0x5404 to 0x540C	Reserved	Undefined
CPU_STATUS	0x5410	Cortex-A9 processor status register	0x0000_0000
RSVD	0x5414 to 0x541C	Reserved	Undefined
PTM_STATUS	0x5420	Program trace macrocell (PTM) status register	0x0000_0000
CLK_DIV_ISP0	0x8300	Set clock divider ratio for CMU_ISP0	0x0000_0000
CLK_DIV_ISP1	0x8304	Set clock divider ratio for MPWM in CMU_ISP1	0x0000_0000
CLK_DIV_STAT_ISP0	0x8400	Clock divider status for CMU_ISP0	0x0000_0000
CLK_DIV_STAT_ISP1	0x8404	Clock divider status for MPWM in CMU_ISP1	0x0000_0000
RSVD	0x8408 to 0x87FC	Reserved	Undefined
CLK_GATE_IP_ISP0	0x8800	Control IP clock gating for ISP_BLK register0	0xFFFF_FFFF
CLK_GATE_IP_ISP1	0x8804	Control IP clock gating for ISP_BLK register1	0xFFFF_FFFF

Register	Offset	Description	Reset Value
RSVD	0x8808 to 0x89FC	Reserved	Undefined
CLKOUT_CMU_ISP	0x8A00	CLKOUT control register	0x0001_0000
CLKOUT_CMU_ISP_DIV_STAT	0x8A04	Clock divider status for CLKOUT	0x0000_0000
CMU_ISP_SPARE0	0x8B00	CMU_ISP spare register0	0x0000_0000
CMU_ISP_SPARE1	0x8B04	CMU_ISP spare register1	0x0000_0000
CMU_ISP_SPARE2	0x8B08	CMU_ISP spare register2	0x0000_0000
CMU_ISP_SPARE3	0x8B0C	CMU_ISP spare register3	0x0000_0000

The six address spaces that SFRs fall into are:SFRs with address 0x0_4000 to 0x0_7FFF-These special function registers control clock-related logics for LEFTBUS block. They control clock source selection, clock divider ratio, and clock gating.

SFRs with address 0x0_8000 to 0x0_BFFF-These special function registers control clock-related logics for RIGHTBUS block. They control clock source selection, clock divider ratio, and clock gating.

SFRs with address 0x0_C000 to 0x0_FFFF-These special function registers control clock-related logics for MFC, G3D, TV, LCD, ISP, CAM, FSYS, PERIL, and PERIR blocks. They control EPLL and VPLL, clock source selection, clock divider ratio, and clock gating.

SFRs with address 0x1_0000 to 0x1_3FFF-These special function registers control clock-related logics for DMC block. They control MPLL, clock source selection, clock divider ratio, and clock gating.

SFRs with address 0x1_4000 to 0x1_7FFF-These special function registers control clock-related logics for CPU block. They control APLL, clock source selection, clock divider ratio and CPU-related logics.

SFRs with address 0x1_8000 to 0x1_BFFF-These special function registers control clock-related logics for ISP block. They control clock source selection, clock divider ratio, and clock gating.

7.9.1.1 CLK_SRC_LEFTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x4200, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0
MUX_MPLL_USER_SEL_L	[4]	RW	Controls MUXMPLL 0 = FINPLL 1 = FOUTMPLL	0x0
RSVD	[3:1]	–	Reserved	0x0
MUX_GDL_SEL	[0]	RW	Controls MUXGDL 0 = SCLKMPLL 1 = SCLKAPLL	0x0

7.9.1.2 CLK_MUX_STAT_LEFTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x4400, Reset Value = 0x0000_0011

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0
MPLL_USER_SEL_L	[6:4]	R	Selection Signal Status of MUXMPLL 001 = FINMPLL 010 = FOUTMPL 1xx = Status that the mux is changing.	0x1
RSVD	[3:1]	–	Reserved	0x0
GDL_SEL	[2:0]	R	Selection Signal Status of MUXGDL 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1

7.9.1.3 CLK_DIV_LEFTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x4500, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0
GPL_RATIO	[6:4]	RW	DIVGPL Clock Divider Ratio ACLK_GPL = MOUTGPL/(GPL_RATIO + 1)	0x0
RSVD	[3]	–	Reserved	0x0
GDL_RATIO	[2:0]	RW	DIVGDL Clock Divider Ratio ACLK_GDL = MOUTGDL/(GDL_RATIO + 1)	0x0

7.9.1.4 CLK_DIV_STAT_LEFTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x4600, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0
DIV_GPL	[4]	R	DIVGPL Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_GDL	[0]	R	DIVGDL Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.5 CLK_GATE_IP_LEFTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x4800, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x1FFFFFFF
CLK_ASYNC_G3D	[6]	RW	Gating all clocks for ASYNC_G3D 0 = Mask 1 = Pass	0x1
RSVD	[5]	–	Reserved	0x1
CLK_ASYNC_MFCL	[4]	RW	Gating all clocks for ASYNC_MFCL 0 = Mask 1 = Pass	0x1
CLK_ASYNC_TVX	[3]	RW	Gating all clocks for ASYNC_TVX 0 = Mask 1 = Pass	0x1
RSVD	[2]	–	Reserved	0x1
CLK_PPMULEFT	[1]	RW	Gating all clocks for PPMULEFT 0 = Mask 1 = Pass	0x1
CLK_GPIO_LEFT	[0]	RW	Gating all clocks for GPIO_LEFT 0 = Mask 1 = Pass	0x1

7.9.1.6 CLK_GATE_IP_IMAGE

- Base Address: 0x1003_0000
- Address = Base Address + 0x4930, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	0x3FFFFF
CLK_PPMUIMAGE	[9]	RW	Gating all clocks for PPMUIMAGE 0 = Mask 1 = Pass	0x1
RSVD	[8]	-	Reserved	0x1
RSVD	[7]	–	Reserved	0x1
RSVD	[6]	–	Reserved	0x1
CLK_SMMUMDMA	[5]	RW	Gating all clocks for SMMUMDMA 0 = Mask 1 = Pass	0x1
CLK_SMMUROTATOR	[4]	RW	Gating all clocks for SMMUROTATOR 0 = Mask 1 = Pass	0x1
RSVD	[3]	–	Reserved	0x1
CLK_MDMA	[2]	RW	Gating all clocks for MDMA 0 = Mask 1 = Pass	0x1
CLK_ROTATOR	[1]	RW	Gating all clocks for ROTATOR 0 = Mask 1 = Pass	0x1
RSVD	[0]	–	Reserved	0x1

7.9.1.7 CLKOUT_CMU_LEFTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x4A00, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0x0
ENB_CLKOUT	[16]	RW	Enable CLKOUT 0 = Disables 1 = Enables	0x1
RSVD	[15:14]	–	Reserved	0x0
DIV_RATIO	[13:8]	RW	Divide Ratio Divide ratio = DIV_RATIO + 1	0x0
RSVD	[7:5]	–	Reserved	0x0
MUX_SEL	[4:0]	RW	MUX selection 00000 = SCLK_MPLL/2 00001 = SCLK_APLL/2 00010 = ACLK_GDL 00011 = ACLK_GPL	0x0

7.9.1.8 CLKOUT_CMU_LEFTBUS_DIV_STAT

- Base Address: 0x1003_0000
- Address = Base Address + 0x4A04, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
DIV_STAT	[0]	R	DIVCLKOUT Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.9 CLK_SRC_RIGHTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x8200, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0
MUX_MPLL_USER_SEL_R	[4]	RW	Controls MUXMPLL 0 = FINPLL 1 = FOUTMPLL	0x0
RSVD	[3:1]	–	Reserved	0x0
MUX_GDR_SEL	[0]	RW	Controls MUXGDR 0 = SCLKMPLL 1 = SCLKAPLL	0x0

7.9.1.10 CLK_MUX_STAT_RIGHTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x8400, Reset Value = 0x0000_0011

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0
MPLL_USER_SEL_R	[6:4]	R	Selection Signal Status of MUXMPLL 001 = FINMPLL 010 = FOUTMPLL 1xx = Status that the mux is changing	0x1
RSVD	[3]	–	Reserved	0x0
GDR_SEL	[2:0]	R	Selection Signal Status of MUXGDR 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1

7.9.1.11 CLK_DIV_RIGHTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x8500, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0
GPR_RATIO	[6:4]	RW	DIVGPR Clock Divider Ratio ACLK_GPR = MOUTGPR/(GPR_RATIO + 1)	0x0
RSVD	[3]	–	Reserved	0x0
GDR_RATIO	[2:0]	RW	DIVGDR Clock Divider Ratio ACLK_GDR = MOUTGDR/(GDR_RATIO + 1)	0x0

7.9.1.12 CLK_DIV_STAT_RIGHTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x8600, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0
DIV_GPR	[4]	R	DIVGPR Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_GDR	[0]	R	DIVGDR Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.13 CLK_GATE_IP_RIGHTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x8800, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	0x3FFFFF
CLK_ASYNC_ISPMX	[9]	RW	Gating all clocks for ASYNC_ISPMX 0 = Mask 1 = Pass	0x1
RSVD	[8]	–	Reserved	0x1
CLK_ASYNC_MAUDIOX	[7]	RW	Gating all clocks for ASYNC_MAUDIOX 0 = Mask 1 = Pass	0x1
CLK_ASYNC_MFCR	[6]	RW	Gating all clocks for ASYNC_MFCR 0 = Mask 1 = Pass	0x1
CLK_ASYNC_FSYSD	[5]	RW	Gating all clocks for ASYNC_FSYSD 0 = Mask 1 = Pass	0x1
RSVD	[4]	–	Reserved	0x1
CLK_ASYNC_LCD0X	[3]	RW	Gating all clocks for ASYNC_LCD0X 0 = Mask 1 = Pass	0x1
CLK_ASYNC_CAMX	[2]	RW	Gating all clocks for ASYNC_CAMX 0 = Mask 1 = Pass	0x1
CLK_PPMURIGHT	[1]	RW	Gating all clocks for PPMURIGHT 0 = Mask 1 = Pass	0x1
CLK_GPIO_RIGHT	[0]	RW	Gating all clocks for GPIO_RIGHT 0 = Mask 1 = Pass	0x1

7.9.1.14 CLK_GATE_IP_PERIR

- Base Address: 0x1003_0000
- Address = Base Address + 0x8960, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:19]	–	Reserved	0x1FFF
CLK_CMU_ISPPART	[18]	RW	Gating all clocks for CMU_ISPPART 0 = Mask 1 = Pass	0x1
CLK_TMU_APBIF	[17]	RW	Gating all clocks for TMU_APBIF 0 = Mask 1 = Pass	0x1
CLK_KEYIF	[16]	RW	Gating all clocks for KEYIF 0 = Mask 1 = Pass	0x1
CLK_RTC	[15]	RW	Gating all clocks for RTC 0 = Mask 1 = Pass	0x1
CLK_WDT	[14]	RW	Gating all clocks for WDT 0 = Mask 1 = Pass	0x1
CLK_MCT	[13]	RW	Gating all clocks for System Timer 0 = Mask 1 = Pass	0x1
CLK_SECKEY	[12]	RW	Gating all clocks for SECKEY 0 = Mask 1 = Pass	0x1
CLK_HDMI_CEC	[11]	RW	Gating all clocks for HDMI_CEC 0 = Mask 1 = Pass	0x1
CLK_TZPC5	[10]	RW	Gating all clocks for TZPC5 0 = Mask 1 = Pass	0x1
CLK_TZPC4	[9]	RW	Gating all clocks for TZPC4 0 = Mask 1 = Pass	0x1
CLK_TZPC3	[8]	RW	Gating all clocks for TZPC3 0 = Mask 1 = Pass	0x1
CLK_TZPC2	[7]	RW	Gating all clocks for TZPC2 0 = Mask 1 = Pass	0x1
CLK_TZPC1	[6]	RW	Gating all clocks for TZPC1 0 = Mask	0x1

Name	Bit	Type	Description	Reset Value
			1 = Pass	
CLK_TZPC0	[5]	RW	Gating all clocks for TZPC0 0 = Mask 1 = Pass	0x1
CLK_CMU_COREPART	[4]	RW	Gating all clocks for CMU_COREPART 0 = Mask 1 = Pass	0x1
CLK_CMU_TOPPART	[3]	RW	Gating all clocks for CMU_TOPPART 0 = Mask 1 = Pass	0x1
CLK_PMU_APBIF	[2]	RW	Gating all clocks for PMU_APBIF 0 = Mask 1 = Pass	0x1
CLK_SYSREG	[1]	RW	Gating all clocks for SYSREG 0 = Mask 1 = Pass	0x1
CLK_CHIP_ID	[0]	RW	Gating all clocks for CHIP ID 0 = Mask 1 = Pass	0x1

7.9.1.15 CLKOUT_CMU_RIGHTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x8A00, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0x0
ENB_CLKOUT	[16]	RW	Enable CLKOUT 0 = Disables 1 = Enables	0x1
RSVD	[15:14]	–	Reserved	0x0
DIV_RATIO	[13:8]	RW	Divide Ratio Divide ratio = DIV_RATIO + 1	0x0
RSVD	[7:5]	–	Reserved	0x0
MUX_SEL	[4:0]	RW	MUX Selection 00000 = SCLK_MPLL/2 00001 = SCLK_APLL/2 00010 = ACLK_GDR 00011 = ACLK_GPR	0x0

7.9.1.16 CLKOUT_CMU_RIGHTBUS_DIV_STAT

- Base Address: 0x1003_0000
- Address = Base Address + 0x8A04, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
DIV_STAT	[0]	R	DIVCLKOUT Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.17 EPLL_LOCK

- Base Address: 0x1003_0000
- Address = Base Address + 0xC010, Reset Value = 0x0000_0FFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0
PLL_LOCKTIME	[15:0]	RW	Required period to generate a stable clock output Set (3000cycles x PDIV) to PLL_LOCKTIME for the PLL maximum lock time. 1 cycle = 1/FREF=1/(FIN/PDIV) The maximum PLL lock time is 250usec where FIN is 24MHz, PDIV is 2 and PLL_LOCKTIME is 6000.	0xFFF

The maximum lock time means the waiting time for locking in the worst case. Therefore, the user of this PLL must wait for more than the maximum lock time unconditionally before the PLL is locked. (Waiting time before locking ≥ the maximum locktime)

7.9.1.18 VPLL_LOCK

- Base Address: 0x1003_0000
- Address = Base Address + 0xC020, Reset Value = 0x0000_0FFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0
PLL_LOCKTIME	[15:0]	RW	Required period to generate a stable clock output Set (3000cycles x PDIV) to PLL_LOCKTIME for the PLL maximum lock time. 1 cycle = 1/FREF=1/(FIN/PDIV) The maximum PLL lock time is 250usec where FIN is 24MHz, PDIV is 2 and PLL_LOCKTIME is 6000.	0xFFF

The maximum lock time means the waiting time for locking in the worst case. Therefore, the user of this PLL must wait for more than the maximum lock time unconditionally before the PLL is locked. (Waiting time before locking ≥ the maximum locktime)

7.9.1.19 EPLL_CON0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC110, Reset Value = 0x0060_0302

Name	Bit	Type	Description	Reset Value
ENABLE	[31]	RW	PLL Enable Control 0 = Disables 1 = Enables	0x0
RSVD	[30]	–	Reserved	0x0
LOCKED	[29]	R	PLL Locking Indication 0 = Unlocks 1 = Lockes This field is set after the locking time. EPLL_LOCK SFR register sets the locking time. It is a Read Only register.	0x0
RSVD	[28:25]	–	Reserved	0x0
MDIV	[24:16]	RW	PLL M Divide Value	0x60
RSVD	[15:14]	–	Reserved	0x0
PDIV	[13:8]	RW	PLL P Divide Value	0x3
RSVD	[7:3]	–	Reserved	0x0
SDIV	[2:0]	RW	PLL S Divide Value	0x2

The reset value of EPLL_CON0 generates a 192 MHz output clock for the input clock frequency of 24 MHz.

The equation to calculate the output frequency is: $F_{OUT} = (MDIV + K/65536) \times F_{IN}/(PDIV \times 2^{SDIV})$
The conditions MDIV, PDIV, SDIV, and K
should meet are:

- PDIV: $1 \leq PDIV \leq 63$
- MDIV: $16 \leq MDIV \leq 511$
- SDIV: $0 \leq SDIV \leq 5$
- K: $0 \leq K \leq 65535$
- Fref = F_{IN}/PDIV, where 4 MHz ≤ Fref ≤ 30 MHz
- F_{VCO}= (MDIV + K/65536) × FIN/PDIV)
- F_{OUT} should fall in the range of: 22 MHz ≤ F_{OUT} ≤ 1400 MHz
  Do not set the value PDIV or MDIV to all zeros.

Refer to the section *7.3.2 Recommended PLL PMS Value for EPLL* for recommended PMS values.

7.9.1.20 EPLL_CON1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC114, Reset Value = 0x6601_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
SEL_PF	[30:29]	RW	Modulation Method Control 00 = Down spread 01 = Up spread 1x = Center spread	0x3
MRR	[28:24]	RW	Modulation Rate Control	0x6
MFR	[23:16]	RW	Modulation Frequency Control	0x1
K	[15:0]	RW	PLL 16-bit DSM (Delta-Sigma Modulator)	0x0

Refer to the section *7.3.2 Recommended PLL PMS Value for EPLL* for the recommended value of K.

7.9.1.21 EPLL_CON2

- Base Address: 0x1003_0000
- Address = Base Address + 0xC118, Reset Value = 0x0000_0080

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	0x0
EXTAFC	[12:8]	RW	AFC value	0x0
DCC_ENB	[7]	RW	Decides whether Duty Cycle Corrector (DCC) is enabled or not. 0= Enables DCC 1= Disables DCC It is an active low signal.	0x1
AFC_ENB	[6]	RW	Decides whether Adaptive Frequency Calibrator (AFC) is enabled or not. When AFC is enabled, it calibrates VCO automatically. 0 = Enables AFC 1 = Disables AFC It is an active low signal.	0x0
SSCG_EN	[5]	RW	Specifies if the dithered mode is enabled or not. 0 = Disables dithered mode 1 = Enables dithered mode	0x0
BYPASS	[4]	RW	If BYPASS = 1, then it enables bypass mode (F_{OUT} = F_{IN}) If BYPASS = 0, then PLL3600X operates normally.	0x0
FVCO_EN	[3]	RW	Enable pin for F_{VCO_OUT}	0x0
FSEL	[2]	RW	Pin selection for monitoring purposes. FV_{CO_OUT} =F_{REF}, if FSEL is set to 0 F_{VCO_OUT} =F_{EED}, FSEL is set to 1	0x0
ICP_BOOST	[1:0]	RW	ICP_BOOST	0x0

If AFC_ENB is set to logic LOW, then it enables the AFC. If AFC_ENB is set to logic HIGH, then EXTAFC [4:0] controls the VCO frequency tuning range.

EXTAFC specifies the decimal value of EXTAFC[4:0] as:

- EXTAFC = EXTAFC[4:0]

The hexadecimal values specified for EXTAFC [4:0] registers are:

- 5'b0 0000 ≤ EXTAFC[4:0] ≤ 5'b1 1111

NOTE: The other PLL control inputs should be set as:
DCC_ENB = 1 ICP_BOOST = 0
SSCG_EN = 0 (Disable dithered mode)
AFC_ENB = 0 EXTAFC = 0

7.9.1.22 VPLL_CON0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC120, Reset Value = 0x006F_0302

Name	Bit	Type	Description	Reset Value
ENABLE	[31]	RW	PLL Enable Control 0 = Disables 1 = Enables	0x0
RSVD	[30]	–	Reserved	0x0
LOCKED	[29]	R	PLL Locking Indication 0 = Unlocks 1 = Locks	0x0
RSVD	[28:25]	–	Reserved	0x0
MDIV	[24:16]	RW	PLL M Divide Value	0x6F
RSVD	[15:14]	–	Reserved	0x0
PDIV	[13:8]	RW	PLL P Divide Value	0x3
RSVD	[7:3]	–	Reserved	0x0
SDIV	[2:0]	RW	PLL S Divide Value	0x2

The reset value of VPLL_CON0 generates a 222.75 MHz output clock for an input clock frequency of 24MHz.
Equation to calculate the output frequency is: $F_{OUT} = (MDIV + K/65535) \times F_{IN}/(PDIV \times 2^{SDIV})$

Where, MDIV, PDIV, SDIV, and K should meet the following conditions:

- PDIV: $1 \leq PDIV \leq 63$
- MDIV: $16 \leq MDIV \leq 511$
- SDIV: $0 \leq SDIV \leq 5$
- K: $0 \leq K \leq 65535$
- Fref = F_{IN}/PDIV) Fref should fall in the range of: 4 MHz ≤ Fref ≤ 30 MHz
- $F_{VCO} = (MDIV + K/v) \times F_{IN}/PDIV)$
- FOUT: 22 MHz ≤ F_{OUT} ≤ 1400 MHz

Do not set the value PDIV or MDIV to all zeros.
Refer to the section *7.3.3 Recommended PLL PMS Value for VPLL* for the recommended PMS values.

7.9.1.23 VPLL_CON1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC124, Reset Value = 0x6601_6000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
SEL_PF	[30:29]	RW	Modulation Method Control 00 = Down spread 01 = Up spread 1x = Center spread	0x3
MRR	[28:24]	RW	Modulation Rate Control	0x6
MFR	[23:16]	RW	Modulation Frequency Control	0x1
K	[15:0]	RW	PLL DSM	0x464

The equation to calculate the Modulation Frequency (MF) is: MF = FFIN/PDIV/MFR/32[Hz]

The equation to calculate the Modulation Rate (MR) is:

- MR = MFR × MRR/MDIV/64 × 100[%]

The conditions that MFR and MRR should meet are:

- MFR should fall in the range of: 0 ≤ MFR ≤ 255
- MRR should fall in the range of: 1 ≤ MRR ≤ 31
- 0 ≤ MRR×MFR ≤ 512
- SEL_PF[1:0]: 2'b00 ≤ SEL_PF ≤ 2'b10

7.9.1.24 VPLL_CON2

- Base Address: 0x1003_0000
- Address = Base Address + 0xC128, Reset Value = 0x0000_0080

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	0x0
EXTAFC	[12:8]	RW	AFC value	0x0
DCC_ENB	[7]	RW	Decides whether DCC is enabled or not. 0 = Enables DCC 1 = Disables DCC It is an active low signal.	0x1
AFC_ENB	[6]	RW	Decides whether AFC is enabled or not. When enabled, VCO is calibrated automatically. 0 = Enables AFC 1 = Disables AFC It is an active low signal.	0x0
SSCG_EN	[5]	RW	Specifies if the dithered mode is enabled or not. 0 = Disables dithered mode 1 = Enables dithered mode	0x0
BYPASS	[4]	RW	If BYPASS = 1, then it enables bypass mode (F_{OUT} = F_{IN}) IF BYPASS = 0, then the PLL3600X operates normally.	0x0
FVCO_EN	[3]	RW	Enable pin for F_{VCO_OUT}	0x0
FSEL	[2]	RW	Specifies pin selection for monitoring purposes $F_{VCO\_OUT} = F_{REF}$, if F_{SEL} is set to 0 $F_{VCO\_OUT} = F_{EED}$, if F_{SEL} is set to 1	0x0
ICP_BOOST	[1:0]	RW	ICP_BOOST	0x0

If AFC_ENB is set to logic LOW, then it enables the AFC. If AFC_ENB is set to logic HIGH, then EXTAFC [4:0] controls the VCO frequency tuning range.

EXTAFC specifies the decimal value of EXTAFC[4:0] as:

- EXTAFC = EXTAFC[4:0]

The hexadecimal values specified for EXTAFC [4:0] registers are:

- 5'b0 0000 ≤ EXTAFC[4:0] ≤ 5'b1 1111

NOTE: The other PLL control inputs should be set as:
DCC_ENB = 1
ICP_BOOST = 0
AFC_ENB = 0
EXTAFC = 0

7.9.1.25 CLK_SRC_TOP0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC210, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
MUX_ONENAND_SEL	[28]	RW	Controls MUXONENAND 0 = ACLK_133 1 = ACLK_160	0x0
RSVD	[27:25]	–	Reserved	0x0
MUX_ACLK_133_SEL	[24]	RW	Controls MUXACLK_133 0 = SCLKMPLL 1 = SCLKAPLL	0x0
RSVD	[23:21]	–	Reserved	0x0
MUX_ACLK_160_SEL	[20]	RW	Controls MUXACLK_160 0 = SCLKMPLL 1 = SCLKAPLL	0x0
RSVD	[19:17]	–	Reserved	0x0
MUX_ACLK_100_SEL	[16]	RW	Controls MUXACLK_100 0 = SCLKMPLL 1 = SCLKAPLL	0x0
RSVD	[15:13]	–	Reserved	0x0
MUX_ACLK_200_SEL	[12]	RW	Controls MUXACLK_200 0 = SCLKMPLL 1 = SCLKAPLL	0x0
RSVD	[11:9]	–	Reserved	0x0
MUX_VPLL_SEL	[8]	RW	Controls MUXVPLL 0 = FINPLL 1 = FOUTVPLL	0x0
RSVD	[7:5]	–	Reserved	0x0
MUX_EPLL_SEL	[4]	RW	Controls MUXEPLL 0 = FINPLL 1 = FOUTEPLL	0x0
RSVD	[3:1]	–	Reserved	0x0
MUX_ONENAND_1_SEL	[0]	RW	Controls MUXONENAND_1 0 = MOUTONENAND 1 = SCLKVPLL	0x0

7.9.1.26 CLK_SRC_TOP1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC214, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
MUX_ACLK_400_ MCUISP_SUB_SEL	[24]	RWX	Controls MUXACLK_400_MCUISP_SUB 0 = FINPLL 1 = DIVOUT_ACLK_400_MCUISP	0x0
RSVD	[23:21]	–	Reserved	0x0
MUX_ACLK_200_ SUB_SEL	[20]	RWX	Controls MUXACLK_200_SUB 0 = FINPLL 1 = DIVOUT_ACLK_200	0x0
RSVD	[19:17]	–	Reserved	0x0
MUX_ACLK_266_ GPS_SUB_SEL	[16]	RWX	Controls MUXACLK_266_GPS_SUB 0 = FINPLL 1 = DIVOUT_ACLK_266_GPS	0x0
RSVD	[15:13]	–	Reserved	0x0
MUX_MPLL_USER_ SEL_T	[12]	RW	Controls MUXMPLL 0 = FINPLL 1 = SCLKMPLLL	0x0
RSVD	[11:9]	–	Reserved	0x0
MUX_ACLK_400_ MCUISP_SEL	[8]	RW	Controls MUXACLK_400_MCUISP 0 = SCLKMPLL_USER_T 1 = SCLKAPLL	0x0
RSVD	[7:5]	–	Reserved	0x0
MUX_ACLK_266_ GPS_SEL	[4]	RW	Controls MUXACLK_266_GPS 0 = SCLKMPLL_USER_T 1 = SCLKAPLL	0x0
RSVD	[3:0]	–	Reserved	0x0

7.9.1.27 CLK_SRC_CAM0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC220, Reset Value = 0x1111_1111

Name	Bit	Type	Description	Reset Value
CSIS1_SEL	[31:28]	RW	Controls MUXCSIS1 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXCSIS1 is the source clock of CSIS1.	0x1
CSIS0_SEL	[27:24]	RW	Controls MUXCSIS0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXCSIS0 is the source clock of CSIS0.	0x1
CAM1_SEL	[23:20]	RW	Controls MUXCAM1 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXCAM1 is the source clock of CAM_B_CLKOUT.	0x1
CAM0_SEL	[19:16]	RW	Controls MUXCAM0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL 0111 = SCLKEPLL 1000 = SCLKVPLL	0x1

Name	Bit	Type	Description	Reset Value
			Others = Reserved MUXCAM0 is the source clock of CAM_A_CLKOUT.	
FIMC3_LCLK_SEL	[15:12]	RW	Controls MUXFIMC3_LCLK 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXFIMC3_LCLK is the source clock of FIMC3 local clock.	0x1
FIMC2_LCLK_SEL	[11:8]	RW	Controls MUXFIMC2_LCLK 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXFIMC2_LCLK is the source clock of FIMC2 local clock.	0x1
FIMC1_LCLK_SEL	[7:4]	RW	Controls MUXFIMC1_LCLK 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXFIMC1_LCLK is the source clock of FIMC1 local clock.	0x1
FIMC0_LCLK_SEL	[3:0]	RW	Controls MUXFIMC0_LCLK 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved	0x1

Name	Bit	Type	Description	Reset Value
			MUXFIMC0_LCLK is the source clock of FIMC0 local clock.	

7.9.1.28 CLK_SRC_TV

- Base Address: 0x1003_0000
- Address = Base Address + 0xC224, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
HDMI_SEL	[0]	RW	Controls MUXHDMI 0 = SCLK_PIXEL 1 = SCLK_HDMIPHY MUXHDMI is the source clock of HDMI link.	0x0

7.9.1.29 CLK_SRC_MFC

- Base Address: 0x1003_0000
- Address = Base Address + 0xC228, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	0x0
MFC_SEL	[8]	RW	Controls MUXMFC 0 = MOUTMFC_0 1 = MOUTMFC_1 MUXMFC is the source clock of MFC core.	0x0
RSVD	[7:5]	–	Reserved	0x0
MFC_1_SEL	[4]	RW	Controls MUXMFC_1 0 = SCLKEPLL 1 = SCLKVPLL MUXMFC_1 is the source clock of MFC core.	0x0
RSVD	[3:1]	–	Reserved	0x0
MFC_0_SEL	[0]	RW	Controls MUXMFC_0 0 = SCLKMPLL 1 = SCLKAPLL MUXMFC_0 is the source clock of MFC core.	0x0

7.9.1.30 CLK_SRC_G3D

- Base Address: 0x1003_0000
- Address = Base Address + 0xC22C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	0x0
G3D_SEL	[8]	RW	Controls MUXG3D 0 = MOUTG3D_0 1 = MOUTG3D_1 MUXG3D is the source clock of G3D core.	0x0
RSVD	[7:5]	–	Reserved	0x0
G3D_1_SEL	[4]	RW	Controls MUXG3D_1 0 = SCLKEPLL 1 = SCLKVPLL MUXG3D_1 is the source clock of G3D core.	0x0
RSVD	[3:1]	–	Reserved	0x0
G3D_0_SEL	[0]	RW	Controls MUXG3D_0 0 = SCLKMPLL 1 = SCLKAPLL MUXG3D_0 is the source clock of G3D core.	0x0

7.9.1.31 CLK_SRC_LCD0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC234, Reset Value = 0x0000_1111

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0
MIPI0_SEL	[15:12]	RW	Controls MUXMIPI0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMIPI0 is the source clock of MIPI_DSIM0.	0x1
MDNIE_PWM0_SEL	[11:8]	RW	Controls MUXMDNIE_PWM0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMDNIE_PWM0 is the source clock of MDNIE_PWM0.	0x1
MDNIE0_SEL	[7:4]	RW	Controls MUXMDNIE0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMDNIE0 is the source clock of MDNIE0.	0x1
FIMD0_SEL	[3:0]	RW	Controls MUXFIMD0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL	0x1

Name	Bit	Type	Description	Reset Value
			1000 = SCLKVPLL Others = Reserved MUXFIMD0 is the source clock of FIMD0.	

7.9.1.32 CLK_SRC_ISP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC238, Reset Value = 0x0000_1111

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0
UART_ISP_SEL	[15:12]	RW	Controls MUXUART_ISP 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXUART_ISP is the source clock of MIPI_DSIM1.	0x1
SPI1_ISP_SEL	[11:8]	RW	Controls MUXSPI1_ISP 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXSPI1_ISP is the source clock of SPI1_ISP.	0x1
SPI0_ISP_SEL	[7:4]	RW	Controls MUXSPI0_ISP 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXSPI0_ISP is the source clock of SPI0_ISP.	0x1
PWM_ISP_SEL	[3:0]	RW	Controls MUXPWM_ISP 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL	0x1

Name	Bit	Type	Description	Reset Value
			Others = Reserved MUXPWM_ISP is the source clock of PWM_ISP.	

7.9.1.33 CLK_SRC_MAUDIO

- Base Address: 0x1003_0000
- Address = Base Address + 0xC23C, Reset Value = 0x0000_0005

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
AUDIO0_SEL	[3:0]	RW	Controls MUXAUDIO0 0000 = AUDIOCDCLK0 0001 = Reserved 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0100 = XXTI 0101 = XusbXTI 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXAUDIO0 is the source clock of AUDIO0.	0x5

7.9.1.34 CLK_SRC_FSYS

- Base Address: 0x1003_0000
- Address = Base Address + 0xC240, Reset Value = 0x0001_1111

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
MIPIHSI_SEL	[24]	RW	Control MUXMIPIHSI, which is the source clock of MIPIHSI 0 = SCLKMPLL_USER_T 1 = SCLKAPLL	0x0
RSVD	[23:20]	-	Reserved	0x0
MMC4_SEL	[19:16]	RW	Controls MUXMMC4 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMMC4 is the source clock of MMC4.	0x1
MMC3_SEL	[15:12]	RW	Controls MUXMMC3 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMMC3 is the source clock of MMC3.	0x1
MMC2_SEL	[11:8]	RW	Controls MUXMMC2 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMMC2 is the source clock of MMC2.	0x1
MMC1_SEL	[7:4]	RW	Controls MUXMMC1 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M	0x1

Name	Bit	Type	Description	Reset Value
			0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMMC1 is the source clock of MMC1.	
MMC0_SEL	[3:0]	RW	Controls MUXMMC0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMMC0 is the source clock of MMC0.	0x1

7.9.1.35 CLK_SRC_PERIL0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC250, Reset Value = 0x0001_1111

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved It should be 1'b1.	0x0
UART4_SEL	[19:16]	RW	Controls MUXUART4 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXUART4 is the source clock of UART4.	0x1
UART3_SEL	[15:12]	RW	Controls MUXUART3 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXUART3 is the source clock of UART3.	0x1
UART2_SEL	[11:8]	RW	Controls MUXUART2 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXUART2 is the source clock of UART2.	0x1
UART1_SEL	[7:4]	RW	Controls MUXUART1 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL	0x1

Name	Bit	Type	Description	Reset Value
			1000 = SCLKVPLL Others = Reserved MUXUART1 is the source clock of UART1.	
UART0_SEL	[3:0]	RW	Controls MUXUART0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXUART0 is the source clock of UART0.	0x1

7.9.1.36 CLK_SRC_PERIL1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC254, Reset Value = 0x0111_0055

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
SPI2_SEL	[27:24]	RW	Controls MUXSPI2 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXSPI2 is the source clock of SPI2.	0x1
SPI1_SEL	[23:20]	RW	Controls MUXSPI1 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXSPI1 is the source clock of SPI1.	0x1
SPI0_SEL	[19:16]	RW	Controls MUXSPI0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXSPI0 is the source clock of SPI0.	0x1
RSVD	[15:10]	–	Reserved	0x0
SPDIF_SEL	[9:8]	RW	Controls MUXSPDIF 00 = SCLK_AUDIO0 01 = SCLK_AUDIO1 10 = SCLK_AUDIO2 11 = SPDIF_EXTCLK MUXSPDIF is the source clock of SPDIF.	0x0
AUDIO2	[7:4]	RW	Controls MUXAUDIO2	0x5

Name	Bit	Type	Description	Reset Value
_SEL			0000 = AUDIOCDCLK2 0001 = Reserved 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0100 = XXTI 0101 = XusbXTI 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXAUDIO2 is the source clock of AUDIO2.	
AUDIO1 _SEL	[3:0]	RW	Controls MUXAUDIO1 0000 = AUDIOCDCLK1 0001 = Reserved 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0100 = XXTI 0101 = XusbXTI 0110 = SCLKMPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXAUDIO1 is the source clock of AUDIO1.	0x5

7.9.1.37 CLK_SRC_CAM1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC258, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	0x0
JPEG_SEL	[8]	RW	Controls MUXJPEG 0 = MOUTJPEG_0 1 = MOUTJPEG_1 MUXJPEG is the source clock of JPEG core.	0x0
RSVD	[7:5]	–	Reserved	0x0
JPEG_1_SEL	[4]	RW	Controls MUXJPEG_1 0 = SCLKEPLL 1 = SCLKVPLL MUXJPEG_1 is the source clock of JPEG core.	0x0
RSVD	[3:1]	–	Reserved	0x0
JPEG_0_SEL	[0]	RW	Controls MUXJPEG_0 0 = SCLKMPLL_USER_T 1 = SCLKAPLL MUXJPEG_0 is the source clock of JPEG core.	0x0

7.9.1.38 CLK_SRC_MASK_CAM0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC320, Reset Value = 0x1111_1111

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
CSIS1_MASK	[28]	RW	Mask output clock of MUXCSIS1 0 = Mask 1 = Unmask	0x1
RSVD	[27:25]	–	Reserved	0x0
CSIS0_MASK	[24]	RW	Mask output clock of MUXCSIS0 0 = Mask 1 = Unmask	0x1
RSVD	[23:21]	–	Reserved	0x0
CAM1_MASK	[20]	RW	Mask output clock of MUXCAM1 0 = Mask 1 = Unmask	0x1
RSVD	[19:17]	–	Reserved	0x0
CAM0_MASK	[16]	RW	Mask output clock of MUXCAM0 0 = Mask 1 = Unmask	0x1
RSVD	[15:13]	–	Reserved	0x0
FIMC3_LCLK_MASK	[12]	RW	Mask output clock of MUXFIMC3_LCLK 0 = Mask 1 = Unmask	0x1
RSVD	[11:9]	–	Reserved	0x0
FIMC2_LCLK_MASK	[8]	RW	Mask output clock of MUXFIMC2_LCLK 0 = Mask 1 = Unmask	0x1
RSVD	[7:5]	–	Reserved	0x0
FIMC1_LCLK_MASK	[4]	RW	Mask output clock of MUXFIMC1_LCLK 0 = Mask 1 = Unmask	0x1
RSVD	[3:1]	–	Reserved	0x0
FIMC0_LCLK_MASK	[0]	RW	Mask output clock of MUXFIMC0_LCLK 0 = Mask 1 = Unmask	0x1

7.9.1.39 CLK_SRC_MASK_TV

- Base Address: 0x1003_0000
- Address = Base Address + 0xC324, Reset Value = 0x0000_0111

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
HDMI_MASK	[0]	RW	Mask output clock of MUXHDMI 0 = Mask 1 = Unmask	0x1

7.9.1.40 CLK_SRC_MASK_LCD

- Base Address: 0x1003_0000
- Address = Base Address + 0xC334, Reset Value = 0x0000_1111

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	0x0
MIPI0_MASK	[12]	RW	Mask output clock of MUXMIPI0 0 = Mask 1 = Unmask	0x1
RSVD	[11:9]	–	Reserved	0x0
MDNIE_PWM0_MASK	[8]	RW	Mask output clock of MUXMDNIE_PWM0 0 = Mask 1 = Unmask	0x1
RSVD	[7:5]	–	Reserved	0x0
MDNIE0_MASK	[4]	RW	Mask output clock of MUXMDNIE0 0 = Mask 1 = Unmask	0x1
RSVD	[3:1]	–	Reserved	0x0
FIMD0_MASK	[0]	RW	Mask output clock of MUXFIMD0 0 = Mask 1 = Unmask	0x1

7.9.1.41 CLK_SRC_MASK_ISP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC338, Reset Value = 0x0000_1111

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	0x0
UART_ISP_MASK	[12]	RW	Mask output clock of MUXUART_ISP 0 = Mask 1 = Unmask	0x1
RSVD	[11:9]	–	Reserved	0x0
SPI1_ISP_MASK	[8]	RW	Mask output clock of MUXSPI1_ISP 0 = Mask 1 = Unmask	0x1
RSVD	[7:5]	–	Reserved	0x0
SPI0_ISP_MASK	[4]	RW	Mask output clock of MUXSPI0_ISP 0 = Mask 1 = Unmask	0x1
RSVD	[3:1]	–	Reserved	0x0
PWM_ISP_MASK	[0]	RW	Mask output clock of MUXPWM_ISP 0 = Mask 1 = Unmask	0x1

7.9.1.42 CLK_SRC_MASK_MAUDIO

- Base Address: 0x1003_0000
- Address = Base Address + 0xC33C, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
AUDIO0_MASK	[0]	RW	Mask output clock of MUXAUDIO0 0 = Mask 1 = Unmask	0x1

7.9.1.43 CLK_SRC_MASK_FSYS

- Base Address: 0x1003_0000
- Address = Base Address + 0xC340, Reset Value = 0x0101_1111

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
MIPIHSI_MASK	[24]	RW	Mask output clock of MUXMIPIHSI 0 = Mask 1 = Unmask	0x1
RSVD	[23:17]	–	Reserved	0x0
MMC4_MASK	[16]	RW	Mask output clock of MUXMMC4 0 = Mask 1 = Unmask	0x1
RSVD	[15:13]	–	Reserved	0x0
MMC3_MASK	[12]	RW	Mask output clock of MUXMMC3 0 = Mask 1 = Unmask	0x1
RSVD	[11:9]	–	Reserved	0x0
MMC2_MASK	[8]	RW	Mask output clock of MUXMMC2 0 = Mask 1 = Unmask	0x1
RSVD	[7:5]	–	Reserved	0x0
MMC1_MASK	[4]	RW	Mask output clock of MUXMMC1 0 = Mask 1 = Unmask	0x1
RSVD	[3:1]	–	Reserved	0x0
MMC0_MASK	[0]	RW	Mask output clock of MUXMMC0 0 = Mask 1 = Unmask	0x1

7.9.1.44 CLK_SRC_MASK_PERIL0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC350, Reset Value = 0x0001_1111

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0x0
UART4_MASK	[16]	RW	Mask output clock of MUXUART4 0 = Mask 1 = Unmask	0x1
RSVD	[15:13]	–	Reserved	0x0
UART3_MASK	[12]	RW	Mask output clock of MUXUART3 0 = Mask 1 = Unmask	0x1
RSVD	[11:9]	–	Reserved	0x0
UART2_MASK	[8]	RW	Mask output clock of MUXUART2 0 = Mask 1 = Unmask	0x1
RSVD	[7:5]	–	Reserved	0x0
UART1_MASK	[4]	RW	Mask output clock of MUXUART1 0 = Mask 1 = Unmask	0x1
RSVD	[3:1]	–	Reserved	0x0
UART0_MASK	[0]	RW	Mask output clock of MUXUART0 0 = Mask 1 = Unmask	0x1

7.9.1.45 CLK_SRC_MASK_PERIL1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC354, Reset Value = 0x0111_0111

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
SPI2_MASK	[24]	RW	Mask output clock of MUXSPI2 0 = Mask 1 = Unmask	0x1
RSVD	[23:21]	–	Reserved	0x0
SPI1_MASK	[20]	RW	Mask output clock of MUXSPI1 0 = Mask 1 = Unmask	0x1
RSVD	[19:17]	–	Reserved	0x0
SPI0_MASK	[16]	RW	Mask output clock of MUXSPI0 0 = Mask 1 = Unmask	0x1
RSVD	[15:9]	–	Reserved	0x0
SPDIF_MASK	[8]	RW	Mask output clock of MUXSPDIF 0 = Mask 1 = Unmask	0x1
RSVD	[7:5]	–	Reserved	0x0
AUDIO2_MASK	[4]	RW	Mask output clock of MUXAUDIO2 0 = Mask 1 = Unmask	0x1
RSVD	[3:1]	–	Reserved	0x0
AUDIO1_MASK	[0]	RW	Mask output clock of MUXAUDIO1 0 = Mask 1 = Unmask	0x1

7.9.1.46 CLK_MUX_STAT_TOP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC410, Reset Value = 0x1111_1111

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
ONENAND_SEL	[30:28]	R	Selection signal status of MUXONENAND 001 = DOUT133 010 = DOUT166 1xx = Status that the mux is changing	0x1
RSVD	[27]	–	Reserved	0x0
ACLK_133_SEL	[26:24]	R	Selection signal status of MUXACLK_133 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1
RSVD	[23]	–	Reserved	0x0
ACLK_160_SEL	[22:20]	R	Selection signal status of MUXACLK_160 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1
RSVD	[19]	–	Reserved	0x0
ACLK_100_SEL	[18:16]	R	Selection signal status of MUXACLK_100 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1
RSVD	[15]	–	Reserved	0x0
ACLK_200_SEL	[14:12]	R	Selection signal status of MUXACLK_200 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1
RSVD	[11]	–	Reserved	0x0
VPLL_SEL	[10:8]	R	Selection signal status of MUXVPLL 001 = FINVPLL 010 = FOUTVPLL 1xx = Status that the mux is changing	0x1
RSVD	[7]	–	Reserved	0x0
EPLL_SEL	[6:4]	R	Selection signal status of MUXEPLL 001 = FINPLL 010 = FOUTEPLL 1xx = Status that the mux is changing	0x1
RSVD	[3]	–	Reserved	0x0
ONENAND_1_SEL	[2:0]	R	Selection signal status of MUXONENAND_1 001 = MOUTONENAND 010 = SCLKVPLL	0x1

Name	Bit	Type	Description	Reset Value
			1xx = Status that the mux is changing	

7.9.1.47 CLK_MUX_STAT_TOP1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC414, Reset Value = 0x0111_1110

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0x0
ACLK_400_MCUISP _SUB_SEL	[26:24]	R	Selection signal status of MUXACLK_400_MCUISP 001 = FINPLL 010 = FOUTPOST_ACLK_400_MCUISP 1xx = Status that the mux is changing	0x1
RSVD	[23]	–	Reserved	0x0
ACLK_200_SUB _SEL	[22:20]	R	Selection signal status of MUXACLK_200 001 = FINPLL 010 = FOUTPOST_ACLK_200 1xx = Status that the mux is changing	0x1
RSVD	[19]	–	Reserved	0x0
ACLK_266_GPS _SUB_SEL	[18:16]	R	Selection signal status of MUXACLK_266_GPS 001 = FINPLL 010 = FOUTPOST_ACLK_266_GPS 1xx = Status that the mux is changing	0x1
RSVD	[15]	–	Reserved	0x0
MPLL_USER_SEL_T	[14:12]	R	Selection signal status of MUXMPLL 001 = FINMPLL 010 = FOUTMPLL 1xx = Status that the mux is changing	0x1
RSVD	[11]	–	Reserved	0x0
ACLK_400_MCUISP _SEL	[10:8]	R	Selection signal status of MUXACLK_400_MCUISP 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1
RSVD	[7]	–	Reserved	0x0
ACLK_266_GPS _SEL	[6:4]	R	Selection signal status of MUXACLK_266_GPS 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1
RSVD	[3:0]	–	Reserved	0x0

7.9.1.48 CLK_MUX_STAT_MFC

- Base Address: 0x1003_0000
- Address = Base Address + 0xC428, Reset Value = 0x0000_0111

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved	0x0
MFC_SEL	[10:8]	R	Selection signal status of MUXMFC 001 = MOUTMFC_0 010 = MOUTMFC_1 1xx = Status that the mux is changing	0x1
RSVD	[7]	–	Reserved	0x0
MFC_1_SEL	[6:4]	R	Selection signal status of MUXMFC_1 001 = SCLKEPLL 010 = SCLKVPLL 1xx = Status that the mux is changing	0x1
RSVD	[3]	–	Reserved	0x0
MFC_0_SEL	[2:0]	R	Selection signal status of MUXMFC_0 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1

7.9.1.49 CLK_MUX_STAT_G3D

- Base Address: 0x1003_0000
- Address = Base Address + 0xC42C, Reset Value = 0x0000_0111

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved	0x0
G3D_SEL	[10:8]	R	Selection signal status of MUXG3D 001 = MOUTG3D_0 010 = MOUTG3D_1 1xx = Status that the mux is changing	0x1
RSVD	[7]	–	Reserved	0x0
G3D_1_SEL	[6:4]	R	Selection signal status of MUXG3D_1 001 = SCLKEPLL 010 = SCLKVPLL 1xx = Status that the mux is changing	0x1
RSVD	[3]	–	Reserved	0x0
G3D_0_SEL	[2:0]	R	Selection signal status of MUXG3D_0 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1

7.9.1.50 CLK_MUX_STAT_CAM1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC458, Reset Value = 0x0000_0111

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved	0x0
JPEG_SEL	[10:8]	R	Selection signal status of MUXJPEG 001 = MOUTJPEG_0 010 = MOUTJPEG_1 1xx = Status that the mux is changing	0x1
RSVD	[7]	–	Reserved	0x0
JPEG_1_SEL	[6:4]	R	Selection signal status of MUXJPEG_1 001 = SCLKEPLL 010 = SCLKVPLL 1xx = Status that the mux is changing	0x1
RSVD	[3]	–	Reserved	0x0
JPEG_0_SEL	[2:0]	R	Selection signal status of MUXJPEG_0 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1

7.9.1.51 CLK_DIV_TOP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC510, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0x0
ACLK_400_MCUISP_RATIO	[26:24]	RW	DIVACLK_266 Clock Divider Ratio ACLK_400_MCUISP = [MOUTACLK_400_MCUISP/(ACLK_400_MCUISP_RATIO + 1)]	0x0
RSVD	[23]	–	Reserved	0x0
ACLK_266_GPS_RATIO	[22:20]	RW	DIVACLK_266 Clock Divider Ratio ACLK_266_GPS = [MOUTACLK_266_GPS /(ACLK_266_GPS_RATIO + 1)]	0x0
RSVD	[19]	–	Reserved	0x0
ONENAND_RATIO	[18:16]	RW	DIVONENAND Clock Divider Ratio SCLK_ONENAND = [MOUTONENAND_1 /(ONENAND_RATIO + 1)]	0x0
RSVD	[15]	–	Reserved	0x0
ACLK_133_RATIO	[14:12]	RW	DIVACLK_133 Clock Divider Ratio ACLK_133 = [MOUTACLK_133 /(ACLK_133_RATIO + 1)]	0x0
RSVD	[11]	–	Reserved	0x0
ACLK_160_RATIO	[10:8]	RW	DIVACLK_160 Clock Divider Ratio ACLK_160 = [MOUTACLK_160 /(ACLK_160_RATIO + 1)]	0x0
ACLK_100_RATIO	[7:4]	RW	DIVACLK_100 Clock Divider Ratio ACLK_100 = [MOUTACLK_100/(ACLK_100_RATIO + 1)]	0x0
RSVD	[3]	–	Reserved	0x0
ACLK_200_RATIO	[2:0]	RW	DIVACLK_200 Clock Divider Ratio ACLK_200 = [MOUTACLK_200/(ACLK_200_RATIO + 1)]	0x0

7.9.1.52 CLK_DIV_CAM0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC520, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CSIS1_RATIO	[31:28]	RW	DIVCSIS1 Clock Divider Ratio SCLK_CSIS1 = MOUTCSIS1/(CSIS1_RATIO + 1)	0x0
CSIS0_RATIO	[27:24]	RW	DIVCSIS0 Clock Divider Ratio SCLK_CSIS0 = MOUTCSIS0/(CSIS0_RATIO + 1)	0x0
CAM1_RATIO	[23:20]	RW	DIVCAM1 Clock Divider Ratio SCLK_CAM1 = MOUTCAM1/(CAM1_RATIO + 1)	0x0
CAM0_RATIO	[19:16]	RW	DIVCAM0 Clock Divider Ratio SCLK_CAM0 = MOUTCAM0/(CAM0_RATIO + 1)	0x0
FIMC3_LCLK_RATIO	[15:12]	RW	DIVFIMC3_LCLK Clock Divider Ratio SCLKFIMC3_LCLK = [MOUTFIMC3_LCLK /(FIMC3_LCLK_RATIO + 1)]	0x0
FIMC2_LCLK_RATIO	[11:8]	RW	DIVFIMC2_LCLK Clock Divider Ratio SCLKFIMC2_LCLK = [MOUTFIMC2_LCLK /(FIMC2_LCLK_RATIO + 1)]	0x0
FIMC1_LCLK_RATIO	[7:4]	RW	DIVFIMC1_LCLK Clock Divider Ratio SCLKFIMC1_LCLK =[MOUTFIMC1_LCLK /(FIMC1_LCLK_RATIO + 1)]	0x0
FIMC0_LCLK_RATIO	[3:0]	RW	DIVFIMC0_LCLK Clock Divider Ratio SCLKFIMC0_LCLK = [MOUTFIMC0_LCLK /(FIMC0_LCLK_RATIO + 1)]	0x0

7.9.1.53 CLK_DIV_TV

- Base Address: 0x1003_0000
- Address = Base Address + 0xC524, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
TV_BLK_RATIO	[3:0]	RW	DIVTV_BLK Clock Divider Ratio SCLK_PIXEL = SCLKVPLL/(TV_BLK_RATIO + 1)	0x0

7.9.1.54 CLK_DIV_MFC

- Base Address: 0x1003_0000
- Address = Base Address + 0xC528, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
MFC_RATIO	[3:0]	RW	DIVMFC Clock Divider Ratio SCLK_MFC = MOUTMFC/(MFC_RATIO + 1)	0x0

7.9.1.55 CLK_DIV_G3D

- Base Address: 0x1003_0000
- Address = Base Address + 0xC52C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
G3D_RATIO	[3:0]	RW	DIVG3D Clock Divider Ratio SCLK_G3D= MOUTG3D/(G3D_RATIO + 1)	0x0

7.9.1.56 CLK_DIV_LCD

- Base Address: 0x1003_0000
- Address = Base Address + 0xC534, Reset Value = 0x0070_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x0
MIPI0_PRE_ RATIO	[23:20]	RW	DIVMIPI0_PRE Clock Divider Ratio SCLK_MIPI0 = DOUTMIPI0/(MIPI0_PRE_RATIO + 1)	0x7
MIPI0_RATIO	[19:16]	RW	DIVMIPI0 Clock Divider Ratio SCLK_MIPIDPHY4L = MOUTMIPI0 /(MIPI0_RATIO + 1)	0x0
MDNIE_PWM0 _PRE_RATIO	[15:12]	RW	DIVMDNIE_PWM0_PRE Clock Divider Ratio SCLK_MDNIE_PWM0 = DOUTMDNIE_PWM0 /(MDNIE_PWM0_PRE_RATIO + 1)	0x0
MDNIE_PWM0 _RATIO	[11:8]	RW	DIVMDNIE_PWM0 Clock Divider Ratio DOUTMDNIE_PWM0 = MOUTMDNIE_PWM0 /(MDNIE_PWM0_RATIO + 1)	0x0
MDNIE0_RATIO	[7:4]	RW	DIVMDNIE0 Clock Divider Ratio SCLK_MDNIE0 = MOUTMDNIE0 /(MDNIE0_RATIO + 1)	0x0
FIMD0_RATIO	[3:0]	RW	DIVFIMD0 Clock Divider Ratio SCLK_FIMD0 = MOUTFIMD0/(FIMD0_RATIO + 1)	0x0

7.9.1.57 CLK_DIV_ISP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC538, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
UART_ISP _RATIO	[31:28]	RW	DIVUART_ISP Clock Divider Ratio SCLK_UART_ISP = [DOUTUART_ISP/(UART_ISP_RATIO + 1)]	0x0
SPI1_ISP_PRE _RATIO	[27:20]	RW	DIVSPI1_ISP_PRE Clock Divider Ratio SCLK_SPI1_ISP = [DOUTSPI1_ISP /(SPI1_ISP_PRE_RATIO + 1)]	0x0
SPI1_ISP_RATIO	[19:16]	RW	DIVSPI1_ISP Clock Divider Ratio DOUTSPI1_ISP = [MOUTSPI1_ISP/(SPI1_ISP_RATIO + 1)]	0x0
SPI0_ISP_PRE _RATIO	[15:8]	RW	DIVSPI0_ISP_PRE Clock Divider Ratio SCLK_SPI0_ISP = [DOUTSPI0_ISP/(SPI0_ISP_PRE_RATIO + 1)]	0x0
SPI0_ISP_RATIO	[7:4]	RW	DIVSPI0_ISP Clock Divider Ratio DOUTSPI0_ISP = [MOUTSPI0_ISP/(SPI0_ISP_RATIO + 1)]	0x0
PWM_ISP_RATIO	[3:0]	RW	DIVPWM_ISP Clock Divider Ratio SCLK_PWM_ISP = [MOUTPWM_ISP/(PWM_ISP_RATIO + 1)]	0x0

7.9.1.58 CLK_DIV_MAUDIO

- Base Address: 0x1003_0000
- Address = Base Address + 0xC53C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0x0
PCM0_RATIO	[11:4]	RW	DIVPCM0 Clock Divider Ratio SCLK_PCM0 = SCLK_AUDIO0/(PCM0_RATIO + 1)	0x0
AUDIO0_RATIO	[3:0]	RW	DIVAUDIO0 Clock Divider Ratio SCLK_AUDIO0 = MOUTAUDIO0/(AUDIO0_RATIO + 1)	0x0

7.9.1.59 CLK_DIV_FSYS0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC540, Reset Value = 0x00B0_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x0
MIPIHSI_RATIO	[23:20]	RW	DIVMIPIHSI Clock Divider Ratio SCLK_MIPIHSI = [MOUTMIPIHSI/(MIPIHSI_RATIO + 1)]	0xB
RSVD	[19:0]	–	Reserved	0x0

7.9.1.60 CLK_DIV_FSYS1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC544, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MMC1_PRE_RATIO	[31:24]	RW	DIVMMC1_PRE Clock Divider Ratio SCLK_MMC1=DOUTMMC1/(MMC1_PRE_RATIO + 1)]	0x0
RSVD	[23:20]	–	Reserved	0x0
MMC1_RATIO	[19:16]	RW	DIVMMC1 Clock Divider Ratio DOUTMMC1 = MOUTMMC1/(MMC1_RATIO + 1)	0x0
MMC0_PRE_RATIO	[15:8]	RW	DIVMMC0_PRE Clock Divider Ratio SCLK_MMC0 =[DOUTMMC0/(MMC0_PRE_RATIO + 1)]	0x0
RSVD	[7:4]	–	Reserved	0x0
MMC0_RATIO	[3:0]	RW	DIVMMC0 Clock Divider Ratio DOUTMMC0 = MOUTMMC0/(MMC0_RATIO + 1)	0x0

7.9.1.61 CLK_DIV_FSYS2

- Base Address: 0x1003_0000
- Address = Base Address + 0xC548, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MMC3_PRE_RATIO	[31:24]	RW	DIVMMC3_PRE Clock Divider Ratio SCLK_MMC3 =[DOUTMMC3/(MMC3_PRE_RATIO + 1)]	0x0
RSVD	[23:20]	–	Reserved	0x0
MMC3_RATIO	[19:16]	RW	DIVMMC3 Clock Divider Ratio DOUTMMC3 = MOUTMMC3/(MMC3_RATIO + 1)	0x0
MMC2_PRE_RATIO	[15:8]	RW	DIVMMC2_PRE Clock Divider Ratio SCLK_MMC2 =[DOUTMMC2/(MMC2_PRE_RATIO + 1)]	0x0
RSVD	[7:4]	–	Reserved	0x0
MMC2_RATIO	[3:0]	RW	DIVMMC2 Clock Divider Ratio DOUTMMC2 = MOUTMMC2/(MMC2_RATIO + 1)	0x0

7.9.1.62 CLK_DIV_FSYS3

- Base Address: 0x1003_0000
- Address = Base Address + 0xC54C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0
MMC4_PRE_RATIO	[15:8]	RW	DIVMMC4_PRE Clock Divider Ratio SCLK_MMC4 =[DOUTMMC4/(MMC4_PRE_RATIO + 1)]	0x0
RSVD	[7:4]	–	Reserved	0x0
MMC4_RATIO	[3:0]	RW	DIVMMC4 Clock Divider Ratio DOUTMMC4 = MOUTMMC4/(MMC4_RATIO + 1)	0x0

7.9.1.63 CLK_DIV_PERIL0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC550, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	0x0
UART4_RATIO	[19:16]	RW	DIVUART4 Clock Divider Ratio SCLK_UART4 = MOUTUART4/(UART4_RATIO + 1)	0x0
UART3_RATIO	[15:12]	RW	DIVUART3 Clock Divider Ratio SCLK_UART3 = MOUTUART3/(UART3_RATIO + 1)	0x0
UART2_RATIO	[11:8]	RW	DIVUART2 Clock Divider Ratio SCLK_UART2 = MOUTUART2/(UART2_RATIO + 1)	0x0
UART1_RATIO	[7:4]	RW	DIVUART1 Clock Divider Ratio SCLK_UART1 = MOUTUART1/(UART1_RATIO + 1)	0x0
UART0_RATIO	[3:0]	RW	DIVUART0 Clock Divider Ratio SCLK_UART0 = MOUTUART0/(UART0_RATIO + 1)	0x0

7.9.1.64 CLK_DIV_PERIL1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC554, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SPI1_PRE_RATIO	[31:24]	RW	DIVSPI1_PRE Clock Divider Ratio SCLK_SPI1 = DOUTSPI1/(SPI1_PRE_RATIO + 1)	0x0
RSVD	[23:20]	–	Reserved	0x0
SPI1_RATIO	[19:16]	RW	DIVSPI1 Clock Divider Ratio DOUTSPI1 = MOUTSPI1/(SPI1_RATIO + 1)	0x0
SPI0_PRE_RATIO	[15:8]	RW	DIVSPI0_PRE Clock Divider Ratio SCLK_SPI0 = DOUTSPI0/(SPI0_PRE_RATIO + 1)	0x0
RSVD	[7:4]	–	Reserved	0x0
SPI0_RATIO	[3:0]	RW	DIVSPI0 Clock Divider Ratio DOUTSPI0 = MOUTSPI0/(SPI0_RATIO + 1)	0x0

7.9.1.65 CLK_DIV_PERIL2

- Base Address: 0x1003_0000
- Address = Base Address + 0xC558, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0
SPI2_PRE_RATIO	[15:8]	RW	DIVSPI2_PRE Clock Divider Ratio SCLK_SPI2 = DOUTSPI2/(SPI2_PRE_RATIO + 1)	0x0
RSVD	[7:4]	–	Reserved	0x0
SPI2_RATIO	[3:0]	RW	DIVSPI2 Clock Divider Ratio DOUTSPI2 = MOUTSPI2/(SPI2_RATIO + 1)	0x0

7.9.1.66 CLK_DIV_PERIL3

- Base Address: 0x1003_0000
- Address = Base Address + 0xC55C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x0
SLIMBUS_RATIO	[7:4]	RW	DIVSLIMBUS Clock Divider Ratio SCLK_SLIMBUS = SCLKEPLL/(SLIMBUS_RATIO + 1)	0x0
RSVD	[3:0]	–	Reserved	0x0

7.9.1.67 CLK_DIV_PERIL4

- Base Address: 0x1003_0000
- Address = Base Address + 0xC560, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
PCM2_RATIO	[27:20]	RW	DIVPCM2 Clock Divider Ratio SCLK_PCM2 = SCLK_AUDIO2/(PCM2_RATIO + 1)	0x0
AUDIO2_RATIO	[19:16]	RW	DIVAUDIO2 Clock Divider Ratio SCLK_AUDIO2 = [MOUTAUDIO2/(AUDIO2_RATIO + 1)]	0x0
RSVD	[15:12]	–	Reserved	0x0
PCM1_RATIO	[11:4]	RW	DIVPCM1 Clock Divider Ratio SCLK_PCM1 = SCLK_AUDIO1/(PCM1_RATIO + 1)	0x0
AUDIO1_RATIO	[3:0]	RW	DIVAUDIO1 Clock Divider Ratio SCLK_AUDIO1 = [MOUTAUDIO1/(AUDIO1_RATIO + 1)]	0x0

7.9.1.68 CLK_DIV_PERIL5

- Base Address: 0x1003_0000
- Address = Base Address + 0xC564, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	–	Reserved	0x0
I2S2_RATIO	[13:8]	RW	DIVI2S2 Clock Divider Ratio SCLK_I2S2 = SCLK_AUDIO2/(I2S2_RATIO + 1)	0x0
RSVD	[7:6]	–	Reserved	0x0
I2S1_RATIO	[5:0]	RW	DIVI2S1 Clock Divider Ratio SCLK_I2S1 = SCLK_AUDIO1/(I2S1_RATIO + 1)	0x0

7.9.1.69 CLK_DIV_CAM1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC568, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
JPEG_RATIO	[3:0]	RW	DIVJPEG Clock Divider Ratio ACLK_JPEG = MOUTJPEG/(JPEG_RATIO + 1)	0x0

7.9.1.70 CLKDIV2_RATIO

- Base Address: 0x1003_0000
- Address = Base Address + 0xC580, Reset Value = 0x0110_1011

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	0x0
GPS_BLK	[25:24]	RW	PCLK Divider Ratio in GPS_BLK 0 = Reserved 1 = Divides by 2 2 = Divides by 3 3 = Divides by 4	0x1
RSVD	[23:22]	–	Reserved	0x0
TV_BLK	[21:20]	RW	PCLK Divider Ratio in TV_BLK 0 = Reserved 1 = Divides by 2 2 = Divides by 3 3 = Divides by 4	0x1
RSVD	[19:14]	–	Reserved	0x0
LCD_BLK	[13:12]	RW	PCLK Divider Ratio in LCD_BLK for 160 MHz domain 0 = Reserved 1 = Divides by 2 2 = Divides by 3 3 = Divides by 4	0x1
RSVD	[11:6]	–	Reserved	0x0
CAM_BLK	[5:4]	RW	PCLK Divider Ratio in CAM_BLK 0 = Reserved 1 = Divides by 2 2 = Divides by 3 3 = Divides by 4	0x1
RSVD	[3:2]	–	Reserved	0x0
FSYS_BLK	[1:0]	RW	PCLK Divider Ratio in FSYS_BLK 0 = Reserved 1 = Divides by 2 2 = Divides by 3 3 = Divides by 4	0x1

7.9.1.71 CLK_DIV_STAT_TOP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC610, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
DIV_ACLK_400_MCUISP	[24]	R	DIVACLK_400_MCUISP Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[23:21]	–	Reserved	0x0
DIV_ACLK_266_GPS	[20]	R	DIVACLK_266_GPS Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[19:17]	–	Reserved	0x0
DIV_ONENAND	[16]	R	DIVONENAND Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[15:13]	–	Reserved	0x0
DIV_ACLK_133	[12]	R	DIVACLK_133 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[11:9]	–	Reserved	0x0
DIV_ACLK_160	[8]	R	DIVACLK_160 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:5]	–	Reserved	0x0
DIV_ACLK_100	[4]	R	DIVACLK_100 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_ACLK_200	[0]	R	DIVACLK_200 Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.72 CLK_DIV_STAT_CAM0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC620, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
DIV_CSIS1	[28]	R	DIVCSIS1 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[27:25]	–	Reserved	0x0
DIV_CSIS0	[24]	R	DIVCSIS0 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[23:21]	–	Reserved	0x0
DIV_CAM1	[20]	R	DIVCAM1 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[19:17]	–	Reserved	0x0
DIV_CAM0	[16]	R	DIVCAM0 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[15:13]	–	Reserved	0x0
DIV_FIMC3_LCLK	[12]	R	DIVFIMC3_LCLK Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[11:9]	–	Reserved	0x0
DIV_FIMC2_LCLK	[8]	R	DIVFIMC2_LCLK Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:5]	–	Reserved	0x0
DIV_FIMC1_LCLK	[4]	R	DIVFIMC1_LCLK Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_FIMC0_LCLK	[0]	R	DIVFIMC0_LCLK Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.73 CLK_DIV_STAT_TV

- Base Address: 0x1003_0000
- Address = Base Address + 0xC624, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
DIV_TV_BLK	[0]	R	DIVTV_BLK Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.74 CLK_DIV_STAT_MFC

- Base Address: 0x1003_0000
- Address = Base Address + 0xC628, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
DIV_MFC	[0]	R	DIVMFC Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.75 CLK_DIV_STAT_G3D

- Base Address: 0x1003_0000
- Address = Base Address + 0xC62C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
DIV_G3D	[0]	R	DIVG3D Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.76 CLK_DIV_STAT_LCD

- Base Address: 0x1003_0000
- Address = Base Address + 0xC634, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:21]	–	Reserved	0x0
DIV_MIPI0_PRE	[20]	R	DIVMIPI0_PRE Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[19:17]	–	Reserved	0x0
DIV_MIPI0	[16]	R	DIVMIPI0 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[15:13]	–	Reserved	0x0
DIV_MDNIE_ PWM0_PRE	[12]	R	DIVMDNIE_PWM0_PRE Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[11:9]	–	Reserved	0x0
DIV_MDNIE_PWM0	[8]	R	DIVMDNIE_PWM0 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:5]	–	Reserved	0x0
DIV_MDNIE0	[4]	R	DIVMDNIE0 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_FIMD0	[0]	R	DIVFIMD0 Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.77 CLK_DIV_STAT_ISP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC638, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
DIV_UART_ISP	[28]	R	DIVUART_ISP Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[27:21]	–	Reserved	0x0
DIV_SPI1_ISP_PRE	[20]	R	DIVSPI1_ISP_PRE Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[19:17]	–	Reserved	0x0
DIV_SPI1_ISP	[16]	R	DIVSPI1_ISP Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[15:9]	–	Reserved	0x0
DIV_SPI0_ISP_PRE	[8]	R	DIVSPI0_ISP_PRE Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:5]	–	Reserved	0x0
DIV_SPI0_ISP	[4]	R	DIVSPI0_ISP Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_PWM_ISP	[0]	R	DIVPWM_ISP Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.78 CLK_DIV_STAT_MAUDIO

- Base Address: 0x1003_0000
- Address = Base Address + 0xC63C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0
DIV_PCM0	[4]	R	DIVPCM0 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_AUDIO0	[0]	R	DIVAUDIO0 Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.79 CLK_DIV_STAT_FSYS0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC640, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:21]	–	Reserved	0x0
DIV_MIPIHSI	[20]	R	DIVMIPIHSI Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[19:0]	–	Reserved	0x0

7.9.1.80 CLK_DIV_STAT_FSYS1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC644, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
DIV_MMC1_PRE	[24]	R	DIVMMC1_PRE Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[23:17]	–	Reserved	0x0
DIV_MMC1	[16]	R	DIVMMC1 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[15:9]	–	Reserved	0x0
DIV_MMC0_PRE	[8]	R	DIVMMC0_PR Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:1]	–	Reserved	0x0
DIV_MMC0	[0]	R	DIVMMC0 Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.81 CLK_DIV_STAT_FSYS2

- Base Address: 0x1003_0000
- Address = Base Address + 0xC648, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
DIV_MMC3_PRE	[24]	R	DIVMMC3_PRE Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[23:17]	–	Reserved	0x0
DIV_MMC3	[16]	R	DIVMMC3 Stats 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[15:9]	–	Reserved	0x0
DIV_MMC2_PRE	[8]	R	DIVMMC2_PRE Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:1]	–	Reserved	0x0
DIV_MMC2	[0]	R	DIVMMC2 Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.82 CLK_DIV_STAT_FSYS3

- Base Address: 0x1003_0000
- Address = Base Address + 0xC64C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	0x0
DIV_MMC4_PRE	[8]	R	DIVMMC4_PRE Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:1]	–	Reserved	0x0
DIV_MMC4	[0]	R	DIVMMC4 Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.83 CLK_DIV_STAT_PERIL0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC650, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0x0
DIV_UART4	[16]	R	DIVUART4 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[15:13]	–	Reserved	0x0
DIV_UART3	[12]	R	DIVUART3 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[11:9]	–	Reserved	0x0
DIV_UART2	[8]	R	DIVUART2 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:5]	–	Reserved	0x0
DIV_UART1	[4]	R	DIVUART1 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_UART0	[0]	R	DIVUART0 Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.84 CLK_DIV_STAT_PERIL1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC654, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
DIV_SPI1_PRE	[24]	R	DIVSPI1_PRE Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[23:17]	–	Reserved	0x0
DIV_SPI1	[16]	R	DIVSPI1 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[15:9]	–	Reserved	0x0
DIV_SPI0_PRE	[8]	R	DIVSPI0_PRE Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:1]	–	Reserved	0x0
DIV_SPI0	[0]	R	DIVSPI0 Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.85 CLK_DIV_STAT_PERIL2

- Base Address: 0x1003_0000
- Address = Base Address + 0xC658, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	0x0
DIV_SPI2_PRE	[8]	R	DIVSPI2_PRE Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:1]	–	Reserved	0x0
DIV_SPI2	[0]	R	DIVSPI2 Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.86 CLK_DIV_STAT_PERIL3

- Base Address: 0x1003_0000
- Address = Base Address + 0xC65C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0
DIV_SLIMBUS	[4]	R	DIVSLIMBUS Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:0]	–	Reserved	0x0

7.9.1.87 CLK_DIV_STAT_PERIL4

- Base Address: 0x1003_0000
- Address = Base Address + 0xC660, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:21]	–	Reserved	0x0
DIV_PCM2	[20]	R	DIVPCM2 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[19:17]	–	Reserved	0x0
DIV_AUDIO2	[16]	R	DIVAUDIO2 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[15:5]	–	Reserved	0x0
DIV_PCM1	[4]	R	DIVPCM1 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_AUDIO1	[0]	R	DIVAUDIO1 Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.88 CLK_DIV_STAT_PERIL5

- Base Address: 0x1003_0000
- Address = Base Address + 0xC664, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	0x0
DIV_I2S2	[8]	R	DIVI2S2 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:1]	–	Reserved	0x0
DIV_I2S1	[0]	R	DIVI2S1 Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.89 CLK_DIV_STAT_CAM1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC668, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
DIV_JPEG	[0]	R	DIVJPEG Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.90 CLKDIV2_STAT

- Base Address: 0x1003_0000
- Address = Base Address + 0xC680, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
GPS_BLK	[24]	R	PCLK Divider Status in TV_BLK 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[23:21]	–	Reserved	0x0
TV_BLK	[20]	R	PCLK Divider Status in TV_BLK 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[19:13]	–	Reserved	0x0
LCD_BLK	[12]	R	PCLK Divider Status in LCD_BLK for 160 MHz domain 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[11:5]	–	Reserved	0x0
CAM_BLK	[4]	R	PCLK Divider Status in CAM_BLK 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
FSYS_BLK	[0]	R	PCLK Divider Status in FSYS_BLK 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.91 CLK_GATE_BUS_FSYS1

- Address = 0x1003_C744, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:23]	-	Reserved	0x1FF
PCLK_ASYNCAXIS_GPS_FSYSD	[22]	RW	Gating APB clock for ASYNCAXIS_GPS_FSYSD 0 = Mask 1 = Pass	0x1
PCLK_AXI_FSYSS	[21]	RW	Gating APB clock for AXI_FSYSS 0 = Mask 1 = Pass	0x1
PCLK_AXI_FSYSD	[20]	RW	Gating APB clock for AXI_FSYSD 0 = Mask 1 = Pass	0x1

RSVD	[19:18]	-	Reserved	0x3
PCLK_PPMUFILE	[17]	RW	Gating APB clock for PPMUFILE 0 = Mask 1 = Pass	0x1
PCLK_ADC	[16]	RW	Gating APB clock for FSYS ADC 0 = Mask 1 = Pass	0x1
RSVD	[15:0]	-	Reserved	0xFFFF

7.9.1.92 CLK_GATE_IP_CAM

- Base Address: 0x1003_0000
- Address = Base Address + 0xC920, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:18]	–	Reserved	0xFFF
RSVD	[19]	-	Reserved	0x1
CLK_PIXELASYNCM1	[18]	RW	Gating all clocks for PIXELASYNCM1 0 = Mask 1 = Pass	0x1
CLK_PIXELASYNCM0	[17]	RW	Gating all clocks for PIXELASYNCM0 0 = Mask 1 = Pass	0x1
CLK_PPMUCAMIF	[16]	RW	Gating all clocks for PPMUCAMIF 0 = Mask 1 = Pass	0x1
RSVD	[12:15]	-	Reserved	0xF
CLK_SMMUJPEG	[11]	RW	Gating all clocks for SMMUJPEG 0 = Mask 1 = Pass	0x1
CLK_SMMUFIMC3	[10]	RW	Gating all clocks for SMMUFIMC3 0 = Mask 1 = Pass	0x1
CLK_SMMUFIMC2	[9]	RW	Gating all clocks for SMMUFIMC2 0 = Mask 1 = Pass	0x1
CLK_SMMUFIMC1	[8]	RW	Gating all clocks for SMMUFIMC1 0 = Mask 1 = Pass	0x1
CLK_SMMUFIMC0	[7]	RW	Gating all clocks for SMMUFIMC0 0 = Mask 1 = Pass	0x1
CLK_JPEG	[6]	RW	Gating all clocks for JPEG 0 = Mask 1 = Pass	0x1
CLK_CSIS1	[5]	RW	Gating all clocks for CSIS1 0 = Mask 1 = Pass	0x1
CLK_CSIS0	[4]	RW	Gating all clocks for CSIS0 0 = Mask 1 = Pass	0x1
CLK_FIMC3	[3]	RW	Gating all clocks for FIMC3 0 = Mask 1 = Pass	0x1

Name	Bit	Type	Description	Reset Value
CLK_FIMC2	[2]	RW	Gating all clocks for FIMC2 0 = Mask 1 = Pass	0x1
CLK_FIMC1	[1]	RW	Gating all clocks for FIMC1 0 = Mask 1 = Pass	0x1
CLK_FIMC0	[0]	RW	Gating all clocks for FIMC0 0 = Mask 1 = Pass	0x1

7.9.1.93 CLK_GATE_IP_TV

- Base Address: 0x1003_0000
- Address = Base Address + 0xC924, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	0x3FFFFFF
CLK_PPMUTV	[5]	RW	Gating all clocks for PPMUTV 0 = Mask 1 = Pass	0x1
CLK_SMMUTV	[4]	RW	Gating all clocks for SMMUTV 0 = Mask 1 = Pass	0x1
CLK_HDMI	[3]	RW	Gating all clocks for HDMI link 0 = Mask 1 = Pass	0x1
RSVD	[2]	–	Reserved	0x1
CLK_MIXER	[1]	RW	Gating all clocks for MIXER 0 = Mask 1 = Pass	0x1
CLK_VP	[0]	RW	Gating all clocks for VP 0 = Mask 1 = Pass	0x1

7.9.1.94 CLK_GATE_IP_MFC

- Base Address: 0x1003_0000
- Address = Base Address + 0xC928, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x7FFFFFF
CLK_PPMUMFC_R	[4]	RW	Gating all clocks for PPMUMFC_R 0 = Mask 1 = Pass	0x1
CLK_PPMUMFC_L	[3]	RW	Gating all clocks for PPMUMFC_L 0 = Mask 1 = Pass	0x1
CLK_SMMUMFC_R	[2]	RW	Gating all clocks for SMMUMFC_R 0 = Mask 1 = Pass	0x1
CLK_SMMUMFC_L	[1]	RW	Gating all clocks for SMMUMFC_L 0 = Mask 1 = Pass	0x1
CLK_MFC	[0]	RW	Gating all clocks for MFC 0 = Mask 1 = Pass	0x1

7.9.1.95 CLK_GATE_IP_G3D

- Base Address: 0x1003_0000
- Address = Base Address + 0xC92C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x1FFFFFFF
RSVD	[2]	-	Reserved	0x1
CLK_PPMUG3D	[1]	RW	Gating all clocks for PPMUG3D 0 = Mask 1 = Pass	0x1
CLK_G3D	[0]	RW	Gating all clocks for G3D 0 = Mask 1 = Pass	0x1

7.9.1.96 CLK_GATE_IP_LCD

- Base Address: 0x1003_0000
- Address = Base Address + 0xC934, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	0x3FFFFFF
CLK_PPMULCD0	[5]	RW	Gating all clocks for PPMULCD0 0 = Mask 1 = Pass	0x1
CLK_SMMUFIMD0	[4]	RW	Gating all clocks for SMMUFIMD0 0 = Mask 1 = Pass	0x1
CLK_DSIM0	[3]	RW	Gating all clocks for DSIM0 0 = Mask 1 = Pass	0x1
CLK_MDNIE0	[2]	RW	Gating all clocks for MDNIE0 0 = Mask 1 = Pass	0x1
CLK_MIE0	[1]	RW	Gating all clocks for MIE0 0 = Mask 1 = Pass	0x1
CLK_FIMD0	[0]	RW	Gating all clocks for FIMD0 0 = Mask 1 = Pass	0x1

7.9.1.97 CLK_GATE_IP_ISP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC938, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0xFFFFFFF
CLK_UART_ISP_SCLK	[3]	RW	Gating SCLK clocks for UART_ISP 0 = Mask 1 = Pass	0x1
CLK_SPI1_ISP_SCLK	[2]	RW	Gating SCLK clocks for SPI1_ISP 0 = Mask 1 = Pass	0x1
CLK_SPI0_ISP_SCLK	[1]	RW	Gating SCLK clocks for SPI0_ISP 0 = Mask 1 = Pass	0x1
CLK_PWM_ISP_SCLK	[0]	RW	Gating SCLK clocks for PWM_ISP 0 = Mask 1 = Pass	0x1

7.9.1.98 CLK_GATE_IP_FSYS

- Base Address: 0x1003_0000
- Address = Base Address + 0xC940, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:18]	–	Reserved	0x1FFF
CLK_PPMUFILE	[17]	RW	Gating all clocks for PPMUFILE 0 = Mask 1 = Pass	0x1
CLK_NFCON	[16]	RW	Gating all clocks for NFCON 0 = Mask 1 = Pass	0x1
CLK_ONENAND	[15]	RW	Gating all clocks for ONENAND 0 = Mask 1 = Pass	0x1
RSVD	[14]	–	Reserved	0x1
CLK_USBDEVICE	[13]	RW	Gating all clocks for USB Device 0 = Mask 1 = Pass	0x1
CLK_USBHOST	[12]	RW	Gating all clocks for USB HOST 0 = Mask 1 = Pass	0x1
CLK_SROMC	[11]	RW	Gating all clocks for SROM 0 = Mask 1 = Pass	0x1
CLK_MIPIHSI	[10]	RW	Gating all clocks for MIPIHSI 0 = Mask 1 = Pass	0x1
CLK_SDMMC4	[9]	RW	Gating all clocks for SDMMC4 0 = Mask 1 = Pass	0x1
CLK_SDMMC3	[8]	RW	Gating all clocks for SDMMC3 0 = Mask 1 = Pass	0x1
CLK_SDMMC2	[7]	RW	Gating all clocks for SDMMC2 0 = Mask 1 = Pass	0x1
CLK_SDMMC1	[6]	RW	Gating all clocks for SDMMC1 0 = Mask 1 = Pass	0x1
CLK_SDMMC0	[5]	RW	Gating all clocks for SDMMC0 0 = Mask 1 = Pass	0x1
CLK_TSI	[4]	RW	Gating all clocks for TSI	0x1

Name	Bit	Type	Description	Reset Value
			0 = Mask 1 = Pass	
RSVD	[3:2]	–	Reserved	0x3
CLK_PDMA1	[1]	RW	Gating all clocks for PDMA1 0 = Mask 1 = Pass	0x1
CLK_PDMA0	[0]	RW	Gating all clocks for PDMA0 0 = Mask 1 = Pass	0x1

7.9.1.99 CLK_GATE_IP_GPS

- Base Address: 0x1003_0000
- Address = Base Address + 0xC94C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0xFFFFFFF
RSVD	[3]	-	Reserved	0x1
CLK_PPMUGPS	[2]	RW	Gating all clocks for PPMUGPS 0 = Mask 1 = Pass	0x1
CLK_SMMUGPS	[1]	RW	Gating all clocks for SMMUGPS 0 = Mask 1 = Pass	0x1
CLK_GPS	[0]	RW	Gating all clocks for GPS 0 = Mask 1 = Pass	0x1

7.9.1.100 CLK_GATE_IP_PERIL

- Base Address: 0x1003_0000
- Address = Base Address + 0xC950, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0xF
CLK_AC97	[27]	RW	Gating all clocks for AC97 0 = Mask 1 = Pass	0x1
CLK_SPDIF	[26]	RW	Gating all clocks for SPDIF 0 = Mask 1 = Pass	0x1
CLK_SLIMBUS	[25]	RW	Gating all clocks for Slimbus 0 = Mask 1 = Pass	0x1
CLK_PWM	[24]	RW	Gating all clocks for PWM 0 = Mask 1 = Pass	0x1
CLK_PCM2	[23]	RW	Gating all clocks for PCM2 0 = Mask 1 = Pass	0x1
CLK_PCM1	[22]	RW	Gating all clocks for PCM1 0 = Mask 1 = Pass	0x1
CLK_I2S2	[21]	RW	Gating all clocks for I2S2 0 = Mask 1 = Pass	0x1
CLK_I2S1	[20]	RW	Gating all clocks for I2S1 0 = Mask 1 = Pass	0x1
RSVD	[19]	–	Reserved	0x1
CLK_SPI2	[18]	RW	Gating all clocks for SPI2 0 = Mask 1 = Pass	0x1
CLK_SPI1	[17]	RW	Gating all clocks for SPI1 0 = Mask 1 = Pass	0x1
CLK_SPI0	[16]	RW	Gating all clocks for SPI0 0 = Mask 1 = Pass	0x1
RSVD	[15]	–	Reserved	0x1
CLK_I2CHDMI	[14]	RW	Gating all clocks for I2CHDMI 0 = Mask 1 = Pass	0x1

Name	Bit	Type	Description	Reset Value
CLK_I2C7	[13]	RW	Gating all clocks for I2C7 0 = Mask 1 = Pass	0x1
CLK_I2C6	[12]	RW	Gating all clocks for I2C6 0 = Mask 1 = Pass	0x1
CLK_I2C5	[11]	RW	Gating all clocks for I2C5 0 = Mask 1 = Pass	0x1
CLK_I2C4	[10]	RW	Gating all clocks for I2C4 0 = Mask 1 = Pass	0x1
CLK_I2C3	[9]	RW	Gating all clocks for I2C3 0 = Mask 1 = Pass	0x1
CLK_I2C2	[8]	RW	Gating all clocks for I2C2 0 = Mask 1 = Pass	0x1
CLK_I2C1	[7]	RW	Gating all clocks for I2C1 0 = Mask 1 = Pass	0x1
CLK_I2C0	[6]	RW	Gating all clocks for I2C0 0 = Mask 1 = Pass	0x1
RSVD	[5]	–	Reserved	0x1
CLK_UART4	[4]	RW	Gating all clocks for UART4 0 = Mask 1 = Pass	0x1
CLK_UART3	[3]	RW	Gating all clocks for UART3 0 = Mask 1 = Pass	0x1
CLK_UART2	[2]	RW	Gating all clocks for UART2 0 = Mask 1 = Pass	0x1
CLK_UART1	[1]	RW	Gating all clocks for UART1 0 = Mask 1 = Pass	0x1
CLK_UART0	[0]	RW	Gating all clocks for UART0 0 = Mask 1 = Pass	0x1

7.9.1.101 CLK_GATE_BLOCK

- Base Address: 0x1003_0000
- Address = Base Address + 0xC970, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0xFFFFFF
CLK_GPS	[7]	RW	Gating all clocks for GPS_BLK (GPS) 0 = Mask 1 = Pass	0x1
RSVD	[6:5]	–	Reserved	0x3
CLK_LCD	[4]	RW	Gating all clocks for LCD_BLK (FIMD0, MIE0, and DSIM0) 0 = Mask 1 = Pass	0x1
CLK_G3D	[3]	RW	Gating all clocks for G3D_BLK (G3D) 0 = Mask 1 = Pass	0x1
CLK_MFC	[2]	RW	Gating all clocks for MFC_BLK (MFC) 0 = Mask 1 = Pass	0x1
CLK_TV	[1]	RW	Gating all clocks for TV_BLK (VP, MIXER, TVENC, and HDMI) 0 = Mask 1 = Pass	0x1
CLK_CAM	[0]	RW	Gating all clocks for CAM_BLK (FIMC0, FIMC1, FIMC2, and FIMC3) 0 = Mask 1 = Pass	0x1

7.9.1.102 CLKOUT_CMU_TOP

- Base Address: 0x1003_0000
- Address = Base Address + 0xCA00, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0x0
ENB_CLKOUT	[16]	RW	Enable CLKOUT 0 = Disables 1 = Enables	0x1
RSVD	[15:14]	–	Reserved	0x0
DIV_RATIO	[13:8]	RW	Divide Ratio Divide ratio = DIV_RATIO + 1	0x0
RSVD	[7:5]	–	Reserved	0x0
MUX_SEL	[4:0]	RW	MUX Selection 00000 = EPLL_FOUT 00001 = VPLL_FOUT 00010 = SCLK_HDMI24M 00011 = SCLK_USBPHY0 00101 = SCLK_HDMIPHY 00110 = AUDIOCDCLK0 00111 = AUDIOCDCLK1 01000 = AUDIOCDCLK2 01001 = SPDIF_EXTCLK 01010 = ACLK_160 01011 = ACLK_133 01100 = ACLK_200 01101 = ACLK_100 01110 = SCLK_MFC 01111 = SCLK_G3D 10000 = ACLK_400_MCUISP 10001 = CAM_A_PCLK 10010 = CAM_B_PCLK 10011 = S_RXBYTECLKHS0_2L 10100 = S_RXBYTECLKHS0_4L 10101 = RX_HALF_BYTE_CLK_CSIS0 10110 = RX_HALF_BYTE_CLK_CSIS1 10111 = SCLK_JPEG 11000 = SCLK_PWM_ISP 11001 = SCLK_SPI0_ISP 11010 = SCLK_SPI1_ISP 11011 = SCLK_UART_ISP 11100 = SCLK_MIPIHSI 11101 = SCLK_HDMI 11110 = SCLK_FIMD0 11111 = SCLK_PCM0	0x0

7.9.1.103 CLKOUT_CMU_TOP_DIV_STAT

- Base Address: 0x1003_0000
- Address = Base Address + 0xCA04, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
DIV_STAT	[0]	R	DIVCLKOUT Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.104 MPLL_LOCK

- Base Address: 0x1004_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0FFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0
PLL_LOCKTIME	[15:0]	RW	Required period to generate a stable clock output Set (270cycles x PDIV) to PLL_LOCKTIME for the PLL maximum lock time. 1 cycle = 1/FREF=1/(FIN/PDIV) The maximum PLL lock time is 22.5usec where FIN is 24MHz, PDIV is 2 and PLL_LOCKTIME is 540.	0xFFF

The maximum lock time means the waiting time for locking in the worst case. Therefore, the user of this PLL must wait for more than the maximum lock time unconditionally before the PLL is locked. (Waiting time before locking ≥ the maximum locktime)

7.9.1.105 MPLL_CON0

- Base Address: 0x1004_0000
- Address = Base Address + 0x0108, Reset Value = 0x0064_0300

Name	Bit	Type	Description	Reset Value
ENABLE	[31]	RW	PLL Enable Control 0 = Disables 1 = Enables	0x0
RSVD	[30]	–	Reserved	0x0
LOCKED	[29]	R	PLL Locking Indication 0 = Unlocks 1 = Locks If ENABLE_LOCK_DET = 0, then this field is set to 1 after the locking time. The lock-time is set using the MPLL_LOCK SFR register. If ENABLE_LOCK_DET = 1, then this field is set when the hardware lock detector meets the PLL locking condition. This bit is Read only.	0x0
RSVD	[28]	–	Reserved	0x0
FSEL	[27]	RW	Monitors Frequency Select Pin 0 = $F_{VCO\_OUT} = F_{REF}$ 1 = $F_{VCO\_OUT} = F_{VCO}$	0x0
RSVD	[26]	–	Reserved	0x0
MDIV	[25:16]	RW	PLL M Divide Value	0x64
RSVD	[15:14]	–	Reserved	0x0
PDIV	[13:8]	RW	PLL P Divide Value	0x3
RSVD	[7:3]	–	Reserved	0x0
SDIV	[2:0]	RW	PLL S Divide Value	0x0

The reset value of MPLL_CON0 generates a 800 MHz output clock for an input clock frequency of 24 MHz.

The equation to calculate the output frequency is: FOUT = MDIV × FIN/(PDIV × 2^{SDIV}): 21.9 MHz ≤ FOUT ≤ 1400 MHz

- The conditions MDIV, PDIV, SDIV for APLL and MPLL should meet are:PDIV: 1 ≤ PDIV ≤ 63
- MDIV: 64 ≤ MDIV ≤ 1023
- SDIV: 0 ≤ SDIV ≤ 5
- Fref = FIN/PDIV) Fref should fall in the range of: 2 MHz ≤ Fref ≤ 12 MHzFVCO = MDIV × FIN/PDIV) FVCO should fall in the range of:700 MHz ≤ FVCO ≤ 1400 MHz

Refer to the section *7.3.1 Recommended PLL PMS Value for APLL and MPLL* for recommended PMS values.

7.9.1.106 MPLL_CON1

- Base Address: 0x1004_0000
- Address = Base Address + 0x010C, Reset Value = 0x0080_3800

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
RESV1	[24]	RW	Specifies status of Linear-Region Detector (LDR) when it detects a low signal	0x0
RESV0	[23]	RW	Specifies VCO range boost-up when the signal is high.	0x1
BYPASS	[22]	RW	If BYPASS = 1, then it enables bypass mode (F_{OUT} = FI_N) If BYPASS = 0, then the PLL3500X operates normally.	0x0
DCC_ENB	[21]	RW	Decides whether DCC is enabled or not. 0 = Enables DCC 1 = Disables DCC It is an active low signal.	0x0
AFC_ENB	[20]	RW	Decides whether AFC is enabled or not. 0 = Enables AFC 1 = Disables AFC It is an active low signal.	0x0
RSVD	[19:18]	–	Reserved	0x0
RSVD	[17]	–	Reserved	0x0
FEED_EN	[16]	RW	Enable pin for FEED_OUT	0x0
LOCK_CON_OUT	[15:14]	RW	Specifies lock detector settings of the output margin.	0x0
LOCK_CON_IN	[13:12]	RW	Specifies lock detector settings of the input margin.	0x3
LOCK_CON_DLY	[11:8]	RW	Specifies lock detector settings of the detection resolution.	0x8
RSVD	[7:5]	–	Reserved	0x0
AFC	[4:0]	RW	AFC value	0x0

Refer to the section *7.3.1 Recommended PLL PMS Value for APLL and MPLL* for recommended AFC_ENB and AFC values.

NOTE: The other PLL control inputs should be set as:

RESV1 = 0	RESV0 = 0
DCC_ENB = 1	EXTAFC = 0
LOCK_CON_IN = 3	LOCK_CON_OUT = 0
LOCK_CON_DLY = 8	AFC_ENB = 0

7.9.1.107 CLK_SRC_DMC

- Base Address: 0x1004_0000
- Address = Base Address + 0x0200, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
MUX_G2D_ACP_SEL	[28]	RW	Control MUXG2D_ACP, which is the source clock of G2D_ACP core 0 = MOUTG2D_ACP_0 1 = MOUTG2D_ACP_1	0x0
RSVD	[27:25]	-	Reserved	0x0
MUX_G2D_ACP_1_SEL	[24]	RW	Control MUXG2D_ACP_1, which is the source clock of G2D_ACP core 0 = SCLKEPLL 1 = SCLKVPLL	0x0
RSVD	[23:21]	-	Reserved	0x0
MUX_G2D_ACP_0_SEL	[20]	RW	Control MUXG2D_ACP_0, which is the source clock of G2D_ACP core 0 = SCLKMPLL 1 = SCLKAPLL	0x0
MUX_PWI_SEL	[19:16]	RW	Controls MUXPWI 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0100 = SCLK_USBPHY1 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXPWI is the clock source of PWI.	0x1
RSVD	[15:13]	–	Reserved	0x0
MUX_MPLL_SEL	[12]	RW	Controls MUXMPLL 0 = FINPLL 1 = MOUTMPLLFOUT	0x0
RSVD	[11:9]	–	Reserved	0x0
MUX_DPHY_SEL	[8]	RW	Controls MUXDPHY 0 = SCLKMPLL 1 = SCLKAPLL	0x0
RSVD	[7:5]	–	Reserved	0x0
MUX_DMC_BUS_SEL	[4]	RW	Controls MUXDMC_BUS 0 = SCLKMPLL 1 = SCLKAPLL	0x0
RSVD	[3:1]	–	Reserved	0x0

Name	Bit	Type	Description	Reset Value
MUX_C2C_SEL	[0]	RW	Controls MUXC2C 0 = SCLKMPLL 1 = SCLKAPLL	0x0

7.9.1.108 CLK_SRC_MASK_DMC

- Base Address: 0x1004_0000
- Address = Base Address + 0x0300, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0x0
PWI_MASK	[16]	RW	Mask output clock of MUXPWI 0 = Mask 1 = Unmask	0x1
RSVD	[15:0]	–	Reserved	0x0

7.9.1.109 CLK_MUX_STAT_DMC

- Base Address: 0x1004_0000
- Address = Base Address + 0x0400, Reset Value = 0x1110_1111

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
G2D_ACP_SEL	[30:28]	R	Selection signal status of MUXG2D_ACP 001 = MOUTG2D_ACP_0 010 = MOUTG2D_ACP_1 1xx = On changing	0x1
RSVD	[27]	-	Reserved	0x0
G2D_ACP_1_SEL	[26:24]	R	Selection signal status of MUXG2D_ACP_1 001 = SCLKEPLL 010 = SCLKVPLL 1xx = On changing	0x1
RSVD	[23]	-	Reserved	0x0
G2D_ACP_0_SEL	[22:20]	R	Selection signal status of MUXG2D_ACP_0 001 = SCLKMPLL 010 = SCLKAPLL 1xx = On changing	0x1
RSVD	[19:15]	–	Reserved	0x0
MPLL_SEL	[14:12]	R	Selection signal status of MUXMPLL 001 = FINPLL 010 = MOUTMPLLFOUT 1xx = Status that the mux is changing	0x1
DPHY_SEL	[10:8]	R	Selection signal status of MUXDMC0 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1
RSVD	[7]	–	Reserved	0x0
DMC_BUS_SEL	[6:4]	R	Selection signal status of MUXDMC_BUS 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1
RSVD	[3]	-	Reserved	0x0
C2C_SEL	[2:0]	R	Selection signal status of MUXC2C 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing	0x1

7.9.1.110 CLK_DIV_DMC0

- Base Address: 0x1004_0000
- Address = Base Address + 0x0500, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:23]	–	Reserved	0x0
DMCP_RATIO	[22:20]	RW	DIVCK133 Clock Divider Ratio ACLK_DMCP = ACLK_DMCD/(DMCP_RATIO + 1)	0x0
RSVD	[19]	–	Reserved	0x0
DMCD_RATIO	[18:16]	RW	DIVDMCD Clock Divider Ratio ACLK_DMCD = DOUTDMC/(DMCD_RATIO + 1)	0x0
RSVD	[15]	–	Reserved	0x0
DMC_RATIO	[14:12]	RW	DIVDMC Clock Divider Ratio DOUTDMC = MOUTDMC_BUS/(DMC_RATIO + 1)	0x0
RSVD	[11]	–	Reserved	0x0
DPHY_RATIO	[10:8]	RW	DIVDPHY Clock Divider Ratio SCLK_DPHY = MOUTDPHY/(DPHY_RATIO + 1)	0x0
RSVD	[7]	–	Reserved	0x0
ACP_PCLK_RATIO	[6:4]	RW	DIVACP Clock Divider Ratio PCLK_ACP = ACLK_ACP/(ACP_PCLK_RATIO + 1)	0x0
RSVD	[3]	–	Reserved	0x0
ACP_RATIO	[2:0]	RW	DIVACP Clock Divider Ratio ACLK_ACP = MOUTDMC_BUS/(ACP_RATIO + 1)	0x0

7.9.1.111 CLK_DIV_DMC1

- Base Address: 0x1004_0000
- Address = Base Address + 0x0504, Reset Value = 0x0000_1000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
DPM_RATIO	[30:24]	RW	DIVDPM Clock Divider Ratio It decides frequency of DPM channel clock.	0x0
RSVD	[23]	–	Reserved	0x0
DVSEM_RATIO	[22:16]	RW	DIVDVSEM Clock Divider Ratio It decides frequency for PWM frame time slot in DVS emulation mode.	0x0
RSVD	[15]	–	Reserved	0x0
C2C_ACLK_RATIO	[14:12]	RW	C2C_ACLK Clock Divider Ratio ACLK_C2C = [MOUTC2C_ACLK/(C2C_ACLK_RATIO + 1)]	0x1
PWI_RATIO	[11:8]	RW	DIVPWI Clock Divider Ratio SCLK_PWI = MOUTPWI/(PWI_RATIO + 1)	0x0
RSVD	[7]	–	Reserved	0x0
C2C_RATIO	[6:4]	RW	C2C clock divider ratio SCLK_C2C = MOUTC2C / (C2C_RATIO + 1)	0x0
G2D_ACP_RATIO	[3:0]	RW	DIVG2D_ACP clock divider ratio SCLK_G2D_ACP= MOUTG2D_ACP / (G2D_ACP_RATIO + 1)	0x0

7.9.1.112 CLK_DIV_STAT_DMC0

- Base Address: 0x1004_0000
- Address = Base Address + 0x0600, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:21]	–	Reserved	0x0
DIV_DMCP	[20]	R	DIVDMCP Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[19:17]	–	Reserved	0x0
DIV_DMCD	[16]	R	DIVDMCD Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[15:13]	–	Reserved	0x0
DIV_DMC	[12]	R	DIVDMC Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[11:9]	–	Reserved	0x0
DIV_DPHY	[8]	R	DIVDPHY Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:5]	–	Reserved	0x0
DIV_ACP_PCLK	[4]	R	DIVACP_PCLK Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_ACP	[0]	R	DIVACP Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.113 CLK_DIV_STAT_DMC1

- Base Address: 0x1004_0000
- Address = Base Address + 0x0604, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
DIV_DPM	[24]	R	DIVDPM Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[23:17]	–	Reserved	0x0
DIV_DVSEM	[16]	R	DIVDVSEM Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[15:13]	–	Reserved	0x0
DIV_C2C_ACLK	[12]	R	DIVC2C_ACLK Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[11:9]	–	Reserved	0x0
DIV_PWI	[8]	R	DIVPWI Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:5]	–	Reserved	0x0
DIV_C2C	[4]	R	DIVC2C status 0 = Stable 1 = Divider is changing	0x0
RSVD	[3:1]	-	Reserved	0x0
DIV_G2D_ACP	[0]	R	DIVG2D_ACP status 0 = Stable 1 = Divider is changing	0x0

7.9.1.114 CLK_GATE_IP_DMC

- Base Address: 0x1004_0000
- Address = Base Address + 0x0900, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
CLK_GPIOC2C	[31]	RW	Gating all clocks for GPIOC2C 0 = Mask 1 = Pass	0x1
RSVD	[30:29]	-	Reserved	0x3
CLK_ASYNC_CPU_XIUR	[28]	RW	Gating all clocks for ASYNC_CPU_XIUR 0 = Mask 1 = Pass	0x1
CLK_ASYNC_C2C_XIUL	[27]	RW	Gating all clocks for ASYNC_C2C_XIUL 0 = Mask 1 = Pass	0x1
CLK_C2C	[26]	RW	Gating all clocks for C2C 0 = Mask 1 = Pass	0x1
RSVD	[25]	-	Reserved	0x1
CLK_SMMUG2D_ACP	[24]	RW	Gating all clocks for SMMUG2D_ACP 0 = Mask 1 = Pass	0x1
CLK_G2D_ACP	[23]	RW	Gating all clocks for G2D_ACP 0 = Mask 1 = Pass	0x1
CLK_ASYNC_GDR	[22]	RW	Gating all clocks for ASYNC_GDR 0 = Mask 1 = Pass	0x1
CLK_ASYNC_GDL	[21]	RW	Gating all clocks for ASYNC_GDL 0 = Mask 1 = Pass	0x1
CLK_GIC	[20]	RW	Gating all clocks for GIC 0 = Mask 1 = Pass	0x1
RSVD	[19]	–	Reserved	0x1
CLK_IEM_IEC	[18]	RW	Gating all clocks for IEM IEC 0 = Mask 1 = Pass	0x1
CLK_IEM_APC	[17]	RW	Gating all clocks for IEM APC 0 = Mask 1 = Pass	0x1
CLK_PPMUACP	[16]	RW	Gating all clocks for PPMUCPU 0 = Mask 1 = Pass	0x1

Name	Bit	Type	Description	Reset Value
RSVD	[15:14]	-	Reserved	0x3
CLK_ID_ REMAPPER	[13]	RW	Gating all clocks for ID_REMAPPER 0 = Mask 1 = Pass	0x1
CLK_SMMUSSS	[12]	RW	Gating all clocks for SMMUSSS 0 = Mask 1 = Pass	0x1
RSVD	[11]	–	Reserved	0x1
CLK_PPMUCPU	[10]	RW	Gating all clocks for PPMUCPU 0 = Mask 1 = Pass	0x1
CLK_PPMUDMC1	[9]	RW	Gating all clocks for PPMUDMC1 0 = Mask 1 = Pass	0x1
CLK_PPMUDMC0	[8]	RW	Gating all clocks for PPMUDMC0 0 = Mask 1 = Pass	0x1
RSVD	[7]	–	Reserved	0x1
CLK_FBMDMC1	[6]	RW	Gating all clocks for FBMDMC1 0 = Mask 1 = Pass	0x1
CLK_FBMDMC0	[5]	RW	Gating all clocks for FBMDMC0 0 = Mask 1 = Pass	0x1
CLK_SSS	[4]	RW	Gating all clocks for SSS 0 = Mask 1 = Pass	0x1
RSVD	[3]	–	Reserved	0x1
CLK_INT_COMB	[2]	RW	Gating all clocks for INT_COMB 0 = Mask 1 = Pass	0x1
RSVD	[1]	–	Reserved	0x1
CLK_DREX2	[0]	RW	Gating all clocks for DREX2 0 = Mask 1 = Pass	0x1

7.9.1.115 CLK_GATE_IP_DMC1

- Base Address: 0x1004_0000
- Address = Base Address + 0x0904, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0xFFFFFFF
CLK_TZASC_LR	[3]	RW	Gating all clocks for TZASC_LR 0 = Mask 1 = Pass	0x1
CLK_TZASC_LW	[2]	RW	Gating all clocks for TZASC_LW 0 = Mask 1 = Pass	0x1
CLK_TZASC_RR	[1]	RW	Gating all clocks for TZASC_RR 0 = Mask 1 = Pass	0x1
CLK_TZASC_RW	[0]	RW	Gating all clocks for TZASC_RW 0 = Mask 1 = Pass	0x1

7.9.1.116 CLKOUT_CMU_DMC

- Base Address: 0x1004_0000
- Address = Base Address + 0x0A00, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0x0
ENB_CLKOUT	[16]	RW	Enable CLKOUT 0 = Disables 1 = Enables	0x1
RSVD	[15:14]	–	Reserved	0x0
DIV_RATIO	[13:8]	RW	Divide Ratio Divide ratio = DIV_RATIO + 1	0x0
RSVD	[7:5]	–	Reserved	0x0
MUX_SEL	[4:0]	RW	MUX Selection 00000 = ACLK_DMCD 00001 = ACLK_DMCP 00010 = ACLK_ACP 00011 = PCLK_ACP 00100 = SCLK_DMC 00101 = SCLK_DPHY 00110 = MPLL_FOUT/2 00111 = SCLK_PWI 01000 = Reserved 01001 = SCLK_C2C 01010 = ACLK_C2C	0x0

7.9.1.117 CLKOUT_CMU_DMC_DIV_STAT

- Base Address: 0x1004_0000
- Address = Base Address + 0x0A04, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
DIV_STAT	[0]	R	DIVCLKOUT Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.118 DCGIDX_MAP0

- Base Address: 0x1004_0000
- Address = Base Address + 0x1000, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
DCGIDX_MAP0	[31:0]	RW	IEC Configuration for DCG Index Map[31:0]	0xFFFFFFFF

7.9.1.119 DCGIDX_MAP1

- Base Address: 0x1004_0000
- Address = Base Address + 0x1004, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
DCGIDX_MAP1	[31:0]	RW	IEC Configuration for DCG Index Map[63:32]	0xFFFFFFFF

7.9.1.120 DCGIDX_MAP2

- Base Address: 0x1004_0000
- Address = Base Address + 0x1008, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
DCGIDX_MAP2	[31:0]	RW	IEC Configuration for DCG Index Map[95:64]	0xFFFFFFFF

7.9.1.121 DCGPERF_MAP0

- Base Address: 0x1004_0000
- Address = Base Address + 0x1020, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
DCGPERF_MAP0	[31:0]	RW	DCG Performance Map[31:0]	0xFFFFFFFF

7.9.1.122 DCGPERF_MAP1

- Base Address: 0x1004_0000
- Address = Base Address + 0x1024, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
DCGPERF_MAP1	[31:0]	RW	DCG Performance Map[63:32]	0xFFFFFFFF

7.9.1.123 DVCIDX_MAP

- Base Address: 0x1004_0000
- Address = Base Address + 0x1040, Reset Value = 0x00FF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x0
DCGPERF_MAP0	[23:0]	RW	IEC Configuration for DVC Index Map[23:0]	0xFFFFFF

7.9.1.124 FREQ_CPU

- Base Address: 0x1004_0000
- Address = Base Address + 0x1060, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x0
FREQ_CPU	[23:0]	RW	Maximum Frequency of CPU in KHz	0x0

7.9.1.125 FREQ_DPM

- Base Address: 0x1004_0000
- Address = Base Address + 0x1064, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x0
FREQ_DPM	[23:0]	RW	Maximum Frequency of DPM	0x0

7.9.1.126 DVSEMCLK_EN

- Base Address: 0x1004_0000
- Address = Base Address + 0x1080, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
DVSEMCLK_EN	[0]	RW	DVS Emulation Clock Enable	0x0

7.9.1.127 MAXPERF

- Base Address: 0x1004_0000
- Address = Base Address + 0x1084, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
MAXPERF_EN	[0]	RW	Maximum Performance Enable 0 = Disables 1 = Enables	0x0

7.9.1.128 DMC_PAUSE_CTRL

- Base Address: 0x1004_0000
- Address = Base Address + 0x1094, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:19]	–	Reserved	0x0
STATE	[18:16]	R	Specifies current status for debugging	0x0
RSVD	[15:1]	–	Reserved	0x0
DMC_PAUSE_ENABLE	[0]	RW	Enable pause function for DREX2 DVFS DREX2 pause function works when DMC_RATIO or DMCD_RATIO in CLK_DIV_DMC0 register is changed.	0x0

7.9.1.129 DDRPHY_LOCK_CTRL

- Base Address: 0x1004_0000
- Address = Base Address + 0x1098, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
USE_CTRL_LOCKED	[31]	RW	Use ctrl_locked signal coming from LPDDR_PHY to check DLL lock-time duration 1 = Uses ctrl_locked signal 0 = Uses internal counter to measure DLL lock duration	0x0
CTRL_START_ENABLE	[30]	RW	Enable Clearing of ctrl_start signal	0x0
CTRL_RESYNC_ENABLE	[29]	RW	Enable ctrl_resync pulse generation	0x0
CTRL_RESYNC_MASK	[28]	RW	Mask ctrl_resync pulse form DREX2 during DDRPHY DLL Locking time	0x0
RSVD	[28:18]	–	Reserved	0x0
CURR_STATE	[17:16]	R	Specifies current status for debugging	0x0
DUR_LOCK_WAIT	[15:8]	RW	Sets Duration for DLL Lock Wait of DDR_PHY	0x0
DUR_CTRL_ST_CLR	[7:0]	RW	Sets Duration for clearing ctrl_start signal of DDR_PHY	0x0

7.9.1.130 C2C_STATE

- Base Address: 0x1004_0000
- Address = Base Address + 0x109C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
CURR_STATE	[2:0]	R	Current State ofC2C SEC FSM	0x0

7.9.1.131 APLL_LOCK

- Base Address: 0x1004_0000
- Address = Base Address + 0x4000, Reset Value = 0x0000_0FFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0
PLL_LOCKTIME	[15:0]	RW	Required period to generate a stable clock output Set (270cycles x PDIV) to PLL_LOCKTIME for the PLL maximum lock time. 1 cycle = 1/FREF=1/(FIN/PDIV) The maximum PLL lock time is 22.5usec where FIN is 24MHz, PDIV is 2 and PLL_LOCKTIME is 540.	0xFFF

The maximum lock time means the waiting time for locking in the worst case. Therefore, the user of this PLL must wait for more than the maximum lock time unconditionally before the PLL is locked. (Waiting time before locking ≥ the maximum locktime)

7.9.1.132 APLL_CON0

- Base Address: 0x1004_0000
- Address = Base Address + 0x4100, Reset Value = 0x0064_0300

Name	Bit	Type	Description	Reset Value
ENABLE	[31]	RW	PLL Enable Control 0 = Disables 1 = Enables	0x0
RSVD	[30]	–	Reserved	0x0
LOCKED	[29]	R	PLL Locking Indication 0 = Unlocks 1 = Locks If ENABLE_LOCK_DET = 0, then this field is set to 1 after the locking time. The lock-time is set using the APLL_LOCK SFR register. If ENABLE_LOCK_DET = 1, then this field is set when the hardware lock detector meets the PLL locking condition. This bit is Read only.	0x0
RSVD	[28]	–	Reserved	0x0
FSEL	[27]	RWX	Monitors Frequency Select Pin 0 = $F_{VCO\_OUT} = F_{REF}$ 1 = $F_{VCO\_OUT} = F_{VCO}$	0x0
RSVD	[26]	–	Reserved	0x0
MDIV	[25:16]	RWX	PLL M Divide Value	0xC8
RSVD	[15:14]	–	Reserved	0x0
PDIV	[13:8]	RWX	PLL P Divide Value	0x6
RSVD	[7:3]	–	Reserved	0x0
SDIV	[2:0]	RWX	PLL S Divide Value	0x1

The reset value of APLL_CON0 generates an 800 MHz output clock for an input clock frequency of 24MHz.

The equation to calculate the output frequency is: $FOUT = MDIV \times FIN/(PDIV \times 2^{SDIV})$

FOUT should fall in the range of: 21.9 MHz ≤ FOUT ≤ 1400 MHz

The conditions MDIV, PDIV, SDIV for APLL and MPLL should meet are:

- PDIV: 1 ≤ PDIV ≤ 63
- MDIV: 64 ≤ MDIV ≤ 1023
- SDIV: 0 ≤ SDIV ≤ 5

- Fref = FIN/PDIV) Fref should fall in the range of: 2 MHz ≤ Fref ≤ 12 MHz
- F_{VCO} = MDIV × F_{IN}/PDIV) F_{VCO} should fall in the range of: 700 MHz ≤ F_{VCO} ≤ 1400 MHz

Refer to the section *7.3.1 Recommended PLL PMS Value for APLL and MPLL* for recommended PMS values.

7.9.1.133 APLL_CON1

- Base Address: 0x1004_0000
- Address = Base Address + 0x4104, Reset Value = 0x0080_3800

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
RESV1	[24]	RW	Specifies status of Linear-Region Detector (LDR) when it detects a low signal.	0x0
RESV0	[23]	RW	Specifies VCO range boost-up when the signal is high.	0x1
BYPASS	[22]	RW	If BYPASS = 1, bypass mode is enabled. (F_{OUT} =F_{IN}) If BYPASS = 0, PLL3500X operates normally.	0x0
DCC_ENB	[21]	RW	Decides whether the DCC is enabled or not. 0 = Enables DCC 1 = Disables DCC It is an active low signal.	0x0
AFC_ENB	[20]	RWX	Decides whether AFC is enabled or not. When AFC is enabled, it calibrates VCO automatically. 0 = Enables AFC 1 = Disables AFC It is an active low signal.	0x0
RSVD	[19:18]	–	Reserved	0x0
RSVD	[17]	–	Reserved	0x0
FEED_EN	[16]	RW	Enable signal for FEED_OUT	0x0
LOCK_CON_OUT	[15:14]	RW	Specifies Lock detector settings of the output margin.	0x0
LOCK_CON_IN	[13:12]	RW	Specifies Lock detector settings of the input margin.	0x3
LOCK_CON_DLY	[11:8]	RW	Specifies Lock detector settings of the detection resolution.	0x8
RSVD	[7:5]	–	Reserved	0x0
AFC	[4:0]	RWX	AFC value	0x0

AFC automatically selects adaptive frequency curve of VCO using switched current bank for wide range, high phase noise (or Jitter), and fast lock time.

Refer to the sectin *7.3.1 Recommended PLL PMS Value for APLL and MPLL* for recommended AFC_ENB and AFC values.

NOTE: The other PLL control inputs should be set as:

RESV1 = 0	RESV0 = 0
DCC_ENB = 1	EXTAFC = 0
LOCK_CON_IN = 3	LOCK_CON_OUT = 0
LOCK_CON_DLY = 8	AFC_ENB = 0

7.9.1.134 CLK_SRC_CPU

- Base Address: 0x1004_0000
- Address = Base Address + 0x4200, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
MUX_MPLL_USER_SEL_C	[24]	RW	Controls MUXMPLL 0 = FINPLL 1 = FOUTMPLL	0x0
RSVD	[23:21]	–	Reserved	0x0
MUX_HPM_SEL	[20]	RW	Controls MUXHPM 0 = MOUTAPLL 1 = SCLKMPLL	0x0
RSVD	[19:17]	–	Reserved	0x0
MUX_CORE_SEL	[16]	RW	Controls MUXCORE 0 = MOUTAPLL 1 = SCLKMPLL	0x0
RSVD	[15:1]	–	Reserved	0x0
MUX_APLL_SEL	[0]	RW	Controls MUXAPLL 0 = FINPLL 1 = MOUTAPLLFOUT	0x0

7.9.1.135 CLK_MUX_STAT_CPU

- Address = 0x1004_4400, Reset Value = 0x0111_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0x0
MPLL_USER_SEL_C	[26:24]	R	Selection signal status of MUXMPLL 001 = FINMPLL 010 = FOUTMPLL 1xx = Status that the mux is changing	0x1
HPM_SEL	[22:20]	R	Selection signal status of MUXHPM 001 = MOUTAPLL 010 = SCLKMPLL 1xx = Status that the mux is changing	0x1
RSVD	[19]	–	Reserved	0x0
CORE_SEL	[18:16]	R	Selection signal status of MUXCORE 001 = MOUTAPLL 010 = SCLKMPLL 1xx = Status that the mux is changing	0x1
RSVD	[15:3]	–	Reserved	0x0
RSVD	[7:3]	–	Reserved	0x0
APLL_SEL	[2:0]	R	Selection signal status of MUXAPLL 001 = FINPLL 010 = MOUTAPLLFOUT 1xx = Status that the mux is changing	0x1

7.9.1.136 CLK_DIV_CPU0

- Base Address: 0x1004_0000
- Address = Base Address + 0x4500, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
CORE2_RATIO	[30:28]	RW	DIVCORE2 Clock Divider Ratio ARMCLK = DOUTCORE/(CORE2_RATIO + 1)	0x0
RSVD	[27]	–	Reserved	0x0
APLL_RATIO	[26:24]	RW	DIVAPLL Clock Divider Ratio SCLKAPLL = MOUTAPLL/(APLL_RATIO + 1)	0x0
RSVD	[23]	–	Reserved	0x0
PCLK_DBG_RATIO	[22:20]	RW	DIVPCLK_DBG Clock Divider Ratio PCLK_DBG = ATCLK/(PCLK_DBG_RATIO + 1)	0x0
RSVD	[19]	–	Reserved	0x0
ATB_RATIO	[18:16]	RW	DIVATB Clock Divider Ratio ATCLK = MOUTCORE/(ATB_RATIO + 1)	0x0
RSVD	[15]	–	Reserved	0x0
PERIPH_RATIO	[14:12]	RW	DIVPERIPH Clock Divider Ratio PERIPHCLK = DOUTCORE/(PERIPH_RATIO + 1)	0x0
RSVD	[11]	–	Reserved	0x0
COREM1_RATIO	[10:8]	RW	DIVCOREM1 Clock Divider Ratio ACLK_COREM1 = ARMCLK/(COREM1_RATIO +1)	0x0
RSVD	[7]	–	Reserved	0x0
COREM0_RATIO	[6:4]	RW	DIVCOREM0 Clock Divider Ratio ACLK_COREM0 = ARMCLK/(COREM0_RATIO +1)	0x0
RSVD	[3]	–	Reserved	0x0
CORE_RATIO	[2:0]	RWX	DIVCORE Clock Divider Ratio DIVCORE_OUT = MOUTCORE/(CORE_RATIO + 1)	0x0

7.9.1.137 CLK_DIV_CPU1

- Base Address: 0x1004_0000
- Address = Base Address + 0x4504, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved	0x0
CORES_RATIO	[10:8]	RW	DIVCORES Clock Divider Ratio ACLK_CORES = ARMCLK/(CORES_RATIO + 1)	0x0
RSVD	[7]	–	Reserved	0x0
HPM_RATIO	[6:4]	RWX	DIVHPM Clock Divider Ratio SCLK_HPM = DOUTCOPY/(HPM_RATIO + 1)	0x0
RSVD	[3]	–	Reserved	0x0
COPY_RATIO	[2:0]	RWX	DIVCOPY Clock Divider Ratio DOUTCOPY = MOUTHPM/(COPY_RATIO + 1)	0x0

7.9.1.138 CLK_DIV_STAT_CPU0

- Base Address: 0x1004_0000
- Address = Base Address + 0x4600, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
DIV_CORE2	[28]	R	DIVCORE2 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[27:25]	–	Reserved	0x0
DIV_APLL	[24]	R	DIVAPLL Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[23:21]	–	Reserved	0x0
DIV_PCLK_DBG	[20]	R	DIVPCLK_DBG Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[19:17]	–	Reserved	0x0
DIV_ATB	[16]	R	DIVATB Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[15:13]	–	Reserved	0x0
DIV_PERIPH	[12]	R	DIVPERIPH Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[11:9]	–	Reserved	0x0
DIV_COREM1	[8]	R	DIVCOREM1 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:5]	–	Reserved	0x0
DIV_COREM0	[4]	R	DIVCOREM0 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_CORE	[0]	R	DIVCORE Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.139 CLK_DIV_STAT_CPU1

- Base Address: 0x1004_0000
- Address = Base Address + 0x4604, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	0x0
DIV_CORES	[8]	R	DIVCORES Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:5]	–	Reserved	0x0
DIV_HPM	[4]	R	DIVHPM Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_COPY	[0]	R	DIVCOPY Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.140 CLK_GATE_IP_CPU

- Base Address: 0x1004_0000
- Address = Base Address + 0x4900, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0xFFFF_FFF3
CLK_CSSYS	[1]	RW	Gating all clocks for CoreSight and SecureJTAG 0 = Mask 1 = Pass	0x1
CLK_HPM	[0]	RW	Gating all clocks for HPM 0 = Mask 1 = Pass	0x1

7.9.1.141 CLKOUT_CMU_CPU

- Base Address: 0x1004_0000
- Address = Base Address + 0x4A00, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0x0
ENB_CLKOUT	[16]	RW	Enable CLKOUT 0 = Disables 1 = Enables	0x1
RSVD	[15:14]	–	Reserved	0x0
DIV_RATIO	[13:8]	RW	Divide Ratio Divide ratio = DIV_RATIO + 1	0x0
RSVD	[7:5]	–	Reserved	0x0
MUX_SEL	[4:0]	RW	MUX Selection 00000 = APLL_FOUT/2 00001 = Reserved 00010 = Reserved 00011 = Reserved 00100 = ARMCLK/2 00101 = ACLK_COREM0 00110 = ACLK_COREM1 00111 = ACLK_CORES 01000 = ATCLK 01001 = PERIPHCLK 01010 = PCLK_DBG 01011 = SCLK_HPM ATCLK and PCLK_DBG are the gated clocks. You should not gate ATCLK or PCLK_DBG clocks before changing the DIV_RATIO value on selection of ATCLK or PCLK_DBG.	0x0

7.9.1.142 CLKOUT_CMU_CPU_DIV_STAT

- Base Address: 0x1004_0000
- Address = Base Address + 0x4A04, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
DIV_STAT	[0]	R	DIVCLKOUT Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.143 ARMCLK_STOPCTRL

- Base Address: 0x1004_0000
- Address = Base Address + 0x5000, Reset Value = 0x0404_0404

Name	Bit	Type	Description	Reset Value
L2_PRE_WAIT_CNT	[31:24]	RW	Specifies clock freeze cycle before the CLAMP_L2_0 and CLAMP_L2_1rising transition	0x4
L2_POST_WAIT_CNT	[23:16]	RW	Specifies clock freeze cycle after the L2RET1N_0 and L2RET1N_1 rising transition	0x4
PRE_WAIT_CNT	[15:8]	RW	Specifies clock freeze cycle before the ARM clamp (CLAMPCORE0, CLAMPCORE1, CLAMPCOREOUT, CLAMPL2_0, and CLAMPL2_1) or reset signal (nCPURESET, nDBGRESET, nSCURESET, L2nRESET, nWDRESET, nPERIPHRESET, and nPTMRESET) transition	0x4
POST_WAIT_CNT	[7:0]	RW	Specifies clock freeze cycle after the ARM clamp (CLAMPCORE0, CLAMPCORE1, CLAMPCOREOUT, CLAMPL2_0, and CLAMPL2_1) or reset signal (nCPURESET, nDBGRESET, nSCURESET, L2nRESET, nWDRESET, nPERIPHRESET, and nPTMRESET) transition	0x4

7.9.1.144 ATCLK_STOPCTRL

- Base Address: 0x1004_0000
- Address = Base Address + 0x5004, Reset Value = 0x0000_0404

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0
PRE_WAIT_CNT	[15:8]	RW	Specifies clock freeze cycle before the ATRESETn, nPRESETDBG, and CSSYS_nRESET signal transition	0x4
POST_WAIT_CNT	[7:0]	RW	Specifies clock freeze cycle after the ATRESETn, nPRESETDBG, and CSSYS_nRESET signal transition	0x4

7.9.1.145 PWR_CTRL

- Base Address: 0x1004_0000
- Address = Base Address + 0x5020, Reset Value = 0x0000_04FF

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
CORE2_RATIO	[30:28]	RW	DIVCORE2 on WFI/WFE Set DIVCORE2 clock divider ratio when both ARM cores are in Wait For Interrupt/Event state	0x0
RSVD	[27:21]	–	Reserved	0x0
CSCLK_AUTO_ ENB_IN_DEBUG	[20]	RW	Forces CoreSight clocks to toggle when the debugger is attached 0 = Disables 1 = Enables	0x0
RSVD	[19]	–	Reserved	0x0
CORE_RATIO	[18:16]	RW	DIVCORE on WFI/WFE Set DIVCORE clock divider ratio when both ARM cores are in Wait For Interrupt/Event state	0x0
RSVD	[15:11]	–	Reserved	0x0
F4D_CORESIGHT_E N	[10]	RW	Gating F4D Coresight clocks both ARM cores in IDLE mode 0 = Mask 1 = Pass	0x1
DIVCORE2_ DOWN_ENB	[9]	RW	Enable ARMCLK Down feature when both ARM cores are in IDLE mode for DIVCORE2 0 = Disables 1 = Enables	0x0
DIVCORE_DOWN _ENB	[8]	RW	Enable ARMCLK Down feature when both ARM cores are in IDLE mode for DIVCORE 0 = Disables 1 = Enables	0x0
USE_STANDBYWFE _ARM_CORE3	[7]	RW	Use ARM CORE3 STANDBYWFE to change ARMCLK frequency in ARM IDLE state	0x1
USE_STANDBYWFE _ARM_CORE2	[6]	RW	Use ARM CORE2 STANDBYWFE to change ARMCLK frequency in ARM IDLE state	0x1
USE_STANDBYWFE _ARM_CORE1	[5]	RW	Use ARM CORE1 STANDBYWFE to change ARMCLK frequency in ARM IDLE state	0x1
USE_STANDBYWFE _ARM_CORE0	[4]	RW	Use ARM CORE0 STANDBYWFE to change ARMCLK frequency in ARM IDLE state	0x1
USE_STANDBYWFI _ARM_CORE3	[3]	RW	Use ARM CORE3 STANDBYWFI to change ARMCLK frequency in ARM IDLE state	0x1
USE_STANDBYWFI _ARM_CORE2	[2]	RW	Use ARM CORE2 STANDBYWFI to change ARMCLK frequency in ARM IDLE state	0x1

USE_STANDBYWFI_ARM_CORE1	[1]	RW	Use ARM CORE1 STANDBYWFI to change ARMCLK frequency in ARM IDLE state	0x1
USE_STANDBYWFI_ARM_CORE0	[0]	RW	Use ARM CORE0 STANDBYWFI to change ARMCLK frequency in ARM IDLE state	0x1

7.9.1.146 PWR_CTRL2

- Base Address: 0x1004_0000
- Address = Base Address + 0x5024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	0x0
DIVCORE2_UP_ENB	[25]	RW	Enable ARMCLK Up feature when both ARM cores exit from IDLE mode for DIVCORE2 0 = Disables 1 = Enables	0x0
DIVCORE_UP_ENB	[24]	RW	Enable ARMCLK Up feature when both ARM cores exit from IDLE mode for DIVCORE 0 = Disables 1 = Enables	0x0
DUR_STANDBY2	[23:16]	RW	Sets duration to change to the normal divider value from the middle divider value This bit should be left-shifted by 4bit before comparing it to the counter value.	0x0
DUR_STANDBY1	[15:8]	RW	Sets duration to change to the middle divider value from the divider value in ARM idle state. This bit should be left-shifted by 4-bit before comparing it to counter value.	0x0
RSVD	[7]	–	Reserved	0x0
UP_CORE2_RATIO	[6:4]	RW	Specifies DIVCORE2 clock divider ratio when ARM0 or ARM1 cores are not in a wait state for an interrupt or event to occur.	0x0
RSVD	[3]	–	Reserved	0x0
UP_CORE_RATIO	[2:0]	RW	Specifies DIVCORE clock divider ratio when ARM0 or ARM1 cores are not in a wait state for an interrupt or event to occur.	0x0

7.9.1.147 L2_STATUS

- Base Address: 0x1004_0000
- Address = Base Address + 0x5400, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
L2IDLE	[28]	R	Indicates L2 cache controller is in idle state	0x0
RSVD	[27:25]	–	Reserved	0x0
L2_CLKSTOPPED	[24]	R	Indicates L2 cache controller is in standby-mode	0x0
RSVD	[23]	–	Reserved	0x0
TAGSETUPLAT	[22:20]	R	Setup Latency for Tag RAM	0x0
RSVD	[19]	–	Reserved	0x0
TAGREADLAT	[18:16]	R	Read access Latency for Tag RAM	0x0
RSVD	[15]	–	Reserved	0x0
TAGWRITELAT	[14:12]	R	Write access Latency for Tag RAM	0x0
RSVD	[11]	–	Reserved	0x0
DATASETUPLAT	[10:8]	R	Setup Latency for Data RAM	0x0
RSVD	[7]	–	Reserved	0x0
DATAREADLAT	[6:4]	R	Read access Latency for Data RAM	0x0
RSVD	[3]	–	Reserved	0x0
DATAWRITELAT	[2:0]	R	Write access Latency for Data RAM	0x0

7.9.1.148 CPU_STATUS

- Base Address: 0x1004_0000
- Address = Base Address + 0x5410, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0x0
PMUPRIV	[11:8]	R	Returns status of the Cortex-A9 processor 0 = User mode 1 = Privileged mode	0x0
PMUSECURE	[7:4]	R	Returns security status of the Cortex-A9 processor 0 = Non-secure state 1 = Secure state	0x0
SMPnAMP	[3:0]	R	Specifies signals AMP or SMP mode for each Cortex-A9 processor 0 = Asymmetric signal 1 = Symmetric signal	0x0

7.9.1.149 PTM_STATUS

- Base Address: 0x1004_0000
- Address = Base Address + 0x5420, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
PTMPWRUP0	[3]	R	PTM for CPU0 is active	0x0
PTMPWRUP1	[2]	R	PTM for CPU1 is active	0x0
PTMIDLEnACK0	[1]	R	PTM for CPU0 is an idle state indicator	0x0
PTMIDLEnACK1	[0]	R	PTM for CPU1 is an idle state indicator	0x0

7.9.1.150 CLK_DIV_ISP0

- Base Address: 0x1004_0000
- Address = Base Address + 0x8300, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0
ISPDIV1_RATIO	[6:4]	RW	ISPDIV1 Clock Divider Ratio ISPDIV1_CLK = ACLK_200/(ISPDIV1_RATIO + 1)	0x0
RSVD	[3]	–	Reserved	0x0
ISPDIV0_RATIO	[2:0]	RW	ISPDIV0 Clock Divider Ratio ISPDIV0_CLK = ACLK_200/(ISPDIV0_RATIO + 1)	0x0

7.9.1.151 CLK_DIV_ISP1

- Base Address: 0x1004_0000
- Address = Base Address + 0x8304, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved	0x0
MCUISPDIV1 _RATIO	[10:8]	RW	MCUISPDIV1 Clock Divider Ratio MCUISPDIV1_CLK = [MOUTMCUISPDIV0_CLK/(MCUISPDIV1_RATIO + 1)]	0x0
RSVD	[7]	–	Reserved	0x0
MCUISPDIV0 _RATIO	[6:4]	RW	MCUISPDIV0 Clock Divider Ratio MCUISPDIV0_CLK = [ACLK_400_MCUIPS /(MCUISPDIV0_RATIO + 1)]	0x0
RSVD	[3]	–	Reserved	0x0
MPWMDIV _RATIO	[2:0]	RW	MPWM Clock Divider Ratio MPWMDIV_CLK = [MOUTISPDIV1_CLK /(MPWMDIV_RATIO + 1)]	0x0

7.9.1.152 CLK_DIV_STAT_ISP0

- Base Address: 0x1004_0000
- Address = Base Address + 0x8400, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0x0
DIV_ISPDIV1	[4]	R	ISPDIV1 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_ISPDIV0	[0]	R	ISPDIV0 Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.153 CLK_DIV_STAT_ISP1

- Base Address: 0x1004_0000
- Address = Base Address + 0x8404, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	0x0
DIV_MCUISPDIV1	[8]	R	DIVMCUISP1 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[7:5]	–	Reserved	0x0
DIV_MCUISPDIV0	[4]	R	DIVMCUISP0 Status 0 = Stable 1 = Status that the divider is changing	0x0
RSVD	[3:1]	–	Reserved	0x0
DIV_MPWMDIV	[0]	R	DIVMPWM Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.154 CLK_GATE_IP_ISP0

- Base Address: 0x1004_0000
- Address = Base Address + 0x8800, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
CLK_UART_ISP	[31]	RW	Gating all clocks for UART_ISP except SCLK 0 = Mask 1 = Pass	0x1
CLK_WDT_ISP	[30]	RW	Gating all clocks for WDT_ISP 0 = Mask 1 = Pass	0x1
RSVD	[29]	–	Reserved	0x1
CLK_PWM_ISP	[28]	RW	Gating all clocks for PWM_ISP except SCLK 0 = Mask 1 = Pass	0x1
CLK_MTCADC_ISP	[27]	RW	Gating all clocks for MTCADC_ISP 0 = Mask 1 = Pass	0x1
CLK_I2C1_ISP	[26]	RW	Gating all clocks for I2C1_ISP 0 = Mask 1 = Pass	0x1
CLK_I2C0_ISP	[25]	RW	Gating all clocks for I2C0_ISP 0 = Mask 1 = Pass	0x1
CLK_MPWM_ISP	[24]	RW	Gating all clocks for MPWM_ISP 0 = Mask 1 = Pass	0x1
CLK_MCUCTL_ISP	[23]	RW	Gating all clocks for MCUCTL_ISP 0 = Mask 1 = Pass	0x1
RSVD	[22]	–	Reserved	0x1
CLK_PPMUISPX	[21]	RW	Gating all clocks for PPMUISPX 0 = Mask 1 = Pass	0x1
CLK_PPMUISPMX	[20]	RW	Gating all clocks for PPMUISPMX 0 = Mask 1 = Pass	0x1
RSVD	[13:19]	–	Reserved	0x7F
CLK_SMMU_LITE1	[12]	RW	Gating all clocks for SMMU_LITE1 0 = Mask 1 = Pass	0x1
CLK_SMMU_LITE0	[11]	RW	Gating all clocks for SMMU_LITE0 0 = Mask 1 = Pass	0x1

Name	Bit	Type	Description	Reset Value
CLK_SMMU_FD	[10]	RW	Gating all clocks for SMMU_FD 0 = Mask 1 = Pass	0x1
CLK_SMMU_DRC	[9]	RW	Gating all clocks for SMMU_DRC 0 = Mask 1 = Pass	0x1
CLK_SMMU_ISP	[8]	RW	Gating all clocks for SMMU_ISP 0 = Mask 1 = Pass	0x1
CLK_GICISP	[7]	RW	Gating all clocks for GICISP 0 = Mask 1 = Pass	0x1
RSVD	[6]	–	Reserved	0x1
CLK_MCUISP	[5]	RW	Gating all clocks for MCUISP 0 = Mask 1 = Pass	0x1
CLK_LITE1	[4]	RW	Gating all clocks for LITE1 0 = Mask 1 = Pass	0x1
CLK_LITE0	[3]	RW	Gating all clocks for LITE0 0 = Mask 1 = Pass	0x1
CLK_FD	[2]	RW	Gating all clocks for FD 0 = Mask 1 = Pass	0x1
CLK_DRC	[1]	RW	Gating all clocks for DRC 0 = Mask 1 = Pass	0x1
CLK_ISP	[0]	RW	Gating all clocks for ISP 0 = Mask 1 = Pass	0x1

7.9.1.155 CLK_GATE_IP_ISP1

- Base Address: 0x1004_0000
- Address = Base Address + 0x8804, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	–	Reserved	0x3FFFF
CLK_SPI1_ISP	[13]	RW	Gating all clocks for SPI1_ISP except SCLK 0 = Mask 1 = Pass	0x1
CLK_SPI0_ISP	[12]	RW	Gating all clocks for SPI0_ISP except SCLK 0 = Mask 1 = Pass	0x1
RSVD	[11:5]	–	Reserved	0x7F
CLK_SMMU_ISPCX	[4]	RW	Gating all clocks for CLK_SMMU_ISPCX 0 = Mask 1 = Pass	0x1
RSVD	[3:1]	–	Reserved	0x7
CLK_ASYNCAXIM	[0]	RW	Gating all clocks for CLK_ASYNCAXIM 0 = Mask 1 = Pass	0x1

7.9.1.156 CLKOUT_CMU_ISP

- Base Address: 0x1004_0000
- Address = Base Address + 0x8A00, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0x0
ENB_CLKOUT	[16]	RW	Enable CLKOUT 0 = Disables 1 = Enables	0x1
RSVD	[15:14]	–	Reserved	0x0
DIV_RATIO	[13:8]	RW	Divide Ratio Divide ratio = DIV_RATIO + 1	0x0
RSVD	[7:5]	–	Reserved	0x0
MUX_SEL	[4:0]	RW	MUX Selection 00000 = ACLK_MCUISP 00001 = PCLKDBG_MCUISP 00010 = ACLK_DIV0 00011 = ACLK_DIV1 00100 = SCLK_MPWM_ISP	0x0

7.9.1.157 CLKOUT_CMU_ISP_DIV_STAT

- Base Address: 0x1004_0000
- Address = Base Address + 0x8A04, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
DIV_STAT	[0]	R	DIVCLKOUT Status 0 = Stable 1 = Status that the divider is changing	0x0

7.9.1.158 CMU_ISP_SPARE0

- Base Address: 0x1004_0000
- Address = Base Address + 0x8B00, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SPARE	[31:0]	RW	CMU_ISP Spare Register	0x0

7.9.1.159 CMU_ISP_SPARE1

- Base Address: 0x1004_0000
- Address = Base Address + 0x8B04, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SPARE	[31:0]	RW	CMU_ISP Spare Register	0x0

7.9.1.160 CMU_ISP_SPARE2

- Base Address: 0x1004_0000
- Address = Base Address + 0x8B08, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SPARE	[31:0]	RW	CMU_ISP Spare Register	0x0

7.9.1.161 CMU_ISP_SPARE3

- Base Address: 0x1004_0000
- Address = Base Address + 0x8B0C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SPARE	[31:0]	RW	CMU_ISP Spare Register	0x0

8 Power Management Unit

8.1 Overview

This chapter describes how Power Management Unit (PMU) controls power and operation state of Exynos 4412 SCP in two ways:

- Local power control:
  - In this state, Exynos 4412 SCP has a number of power domains.
  - You can turn on and turn off some of these power domains independently. This functionality is called local power control.
  - You can handle the local power control by accessing registers of PMU.
- System-level power control:
  - In this state, Exynos 4412 SCP has five kinds of system-level power down-modes:
    - NORMAL
    - AFTR
    - LPA
    - DEEP-STOP (D-STOP)
    - SLEEP
  - You can control the power state of each power domain in a uniform and controlled way to make a system level power state. With local power control, you cannot turn AP into system-level power down modes.

NOTE: System-level power control does not support LPA and DEEP-STOP for engineering samples.

8.2 Background

The total power consumption consists of static and dynamic power consumptions.

There is static power consumption during:

- Power supply to a circuit
- No active operation in the circuit

There is dynamic power consumption during:

- Signal changes to the circuit
- Active operations in the circuit

The static power consumption is due to the leakage of current in the process, while dynamic power consumption is due to the transition of gate state. The dynamic power consumption depends on:

- Operating voltage
- Operating frequency
- Toggling ratios of the logic gate

Table 8-1 illustrates comparison of various power-saving techniques.

Table 8-1 Comparison of Power-saving Techniques

Power-saving Techniques	Result	Clock	Power	State Retention	
				Normal F/F	Retention F/F
Frequency scaling	Reduce dynamic power	Enabled	Supplied	Keep state	
Clock gating	Minimize dynamic power	Disabled	Supplied	Keep state	
Power gating	Minimize leakage power	Disabled	External power supplied, while internally gated	Loose state	Keep state
Power off	Nearly zero power	Disabled	Externally turned off	Loose state	

Frequency scaling refers to lowering the frequency of clock to a specific module when you do not require the module to run fast. You can reduce the dynamic power by scaling the frequency

Clock gating refers to disabling the clock to a specific Intellectual Property (IP) module by using gating cells in PMU. To control these clock gating cells, set registers CLK_GATE_IP0-4 and CLK_GATE_BLOCK in PMU.

You can apply the clock-gating technique in synthesis phase of chip development flow. By applying this technique generate gate-level netlist from RTL code by using synthesis tool. The clock gating cells inserted by synthesis tool are controlled not by software, but by hardware automatically.

When you apply clock-gating technique, there is still a power supply to logic gate. Therefore, retain the Normal Flip-Flop (F/F) and Retention F/F. Develop the Retention F/F to keep its state even though there is no power supply due to power gating.

Power gating means that a current path to a specific power domain (a group of IP modules) is internally disconnected using switch cells in that power domain. Therefore, power to that domain is not supplied. The switch cell is located either between real power and virtual power (HEADER), or between real ground and virtual ground (FOOTER). Remember that external power to Exynos 4412 SCP is not "OFF".

When you apply power gating, the power gated block loses the states of normal F/Fs. However, the power gated block retains the retention F/Fs. Therefore, the two power-gating techniques are:

- Power gating without state retention-Uses normal F/F
- Power gating with state retention-uses retention F/F and requires wake-up reset

Power "OFF" refers to the external power OFF state of Exynos 4412 SCP. You apply the power "OFF" by using the regulator or Power Management IC (PMIC).

In Exynos 4412 SCP, PMU generates power control signal to regulator or PMIC. When you apply power "OFF", all blocks which are applied power "OFF" lose state of normal F/Fs and retention F/F. Therefore, if you want to save some important data, you should move the data to external memory and restore it when wake-up event occurs.

To reduce the dynamic power consumption, Exynos 4412 SCP uses clock gating and frequency scaling. You can disable clocks in Exynos 4412 SCP in module-by-module basis. Reduce the clock frequency when the system does not require operating at the maximum frequency.

To further reduce power consumption in the system level, Exynos 4412 SCP enables the Dynamic Random Access Memory (DRAM) to enter into self-refresh mode and deep power-down mode.

To reduce the static current, Exynos 4412 SCP supports block-based power gating. In specific applications, you do not require a certain group of modules to run. Therefore, you do not have to supply power to the block modules. For example, MP3 playback, Multi-Format Codec (MFC), video modules (camera interface, JPEG, video processor, mixer, and so on), and 3D graphics core need not operate. You can power gate these devices for minimum static power consumption.

8.3 Power Domains

Table 8-2 describes the power domains of Exynos 4412 SCP.

Table 8-2 Exynos 4412 SCP Power Domains

Power Domain	Power Source	Internal Power Gating Method	Included Modules
CPU0	VDD_ARM	PMOS power gating	Core 0 of Cortex-A9 including L1 Cache
CPU1	VDD_ARM	PMOS power gating	Core 1 of Cortex-A9 including L1 Cache
CPU2	VDD_ARM	PMOS power gating	Core 2 of Cortex-A9 including L1 Cache
CPU3	VDD_ARM	PMOS power gating	Core 3 of Cortex-A9 including L1 Cache
CPU_L2_0	VDD_ARM	PMOS power gating	L2 Cache block 0
CPU_L2_1	VDD_ARM	PMOS power gating	L2 Cache block 1
CPU_ETC	VDD_ARM	PMOS power gating	L2 Cache controller, SCU, and so on.
MFC	VDD_INT	PMOS power gating	MFC
G3D	VDD_G3D	PMOS power gating	G3D
AudioSS	VDD_INT	PMOS power gating	SRP, iROM, iRAM
LCD0	VDD_INT	PMOS power gating	LCD controller 0, MIE0, DSI0
ISP	VDD_INT	PMOS power gating	ISP_BLK
TV	VDD_INT	PMOS power gating	VP, MIXER, TV Encoder, HDMI
CAM	VDD_INT	PMOS power gating	Camera, CSI, JPEG
GPS	VDD_INT	PMOS power gating	GPS-BB
GPS_ALIVE	VDD_ALIVE	PMOS power gating	GPS_ALIVE inside GPS_BLK
TOP	VDD_INT	PMOS power gating	CMU, GPIO (OFF), Bus components, INTC, NFCON, OneNANDC, SRAMC, PDMA, MDMA, CoreSight, MDMIF, TSI, SDMMC, USB-HOST, USB-DEV, CHIPID, Security key, SPDIF, PCM, SPI, KEYIF, TSADC, I2C, I2S-1/2, AC97, PCM, WDT, UART, Rotator
DMC	VDD_DMC	PMOS power gating	DRAMC, G2D, CryptoEngine
MCT	VDD_INT	None	System Timer
ALIVE	VDD_ALIVE	None	PMU, GPIO (ALIVE), Wakeup logic
RTC	VDD_RTC	None	RTClock
APLL	VDD10_APLL	None	APLL
MPLL	VDD10_MPLL	None	MPLL
EPLL	VDD10_EPLL	None	EPLL
VPLL	VDD10_VPLL	None	VPLL
LPDDR_PHY	VDD_DMC	None	LPDDR_PHY
HDMI_PHY	VDD10_HDMI VDD10_HDMI_PLL	None	HDMI_PHY

Power Domain	Power Source	Internal Power Gating Method	Included Modules
MIPI_PHY	VDD18_MIPI VDD10_MIPI VDD10_MIPI_PLL	None	MIPI_PHY
MIPI2L_PHY	VDD18_MIPI2L VDD10_MIPI2L VDD10_MIPI2L_PLL	None	MIPI2L_PHY
HSIC_PHY	VDD10_HSIC VDD18_HISC	None	HSIC_PHY
USB_PHY	VDD30_UOTG VDD10_UOTG VDD33REG_UHOST	None	USB_picoPHY
ADC_PHY	VDD18_ADC	None	ADC_PHY
TEMP	VDD18_TEMP	None	TEMP SENSOR
ABBG0	VDD18_ABB0	None	ABBG for INT
ABBG1	VDD18_ABB1	None	ABBG for DMC
ABBG2	VDD18_ABB2	None	ABBG for G3D
ABBG3	VDD18_ABB2	None	ABBG for ARM
D-IO	VDDQ_xx	None	Digital I/O pads

8.4 Local Power Control

This section includes:

- Overview
- Sequence

8.4.1 Overview

- You can turn off or turn on the power domains independently.
  Independent power control on those blocks is called Local Power Control. The power domains include:
- MFC
- G3D
- LCD
- ISP
- CAM
- TV
- GPS
- CPU0
- CPU1
- CPU2
- CPU3
- CPU_L2_0
- CPU_L2_1

DMC block can be automatically in retention state according to C2C state when ENABLE_C2C field of C2C_CTRL register is set to "1".

8.4.2 Sequence

In PMU, each power domain has dedicated register field that handles its power state.

8.4.2.1 Power on/off Sequence of MFC

STATUS field of MFC_STATUS register indicates power state of MFC power domain. 0x7 indicates that MFC block is fully powered up, and 0x0 indicates that MFC block is powered off. All the other values indicate power that on/off sequence is in progress.

By default, the MFC block power is turned on. You can set LOCAL_PWR_CFG field of MFC_CONFIGURATION register to 0x0 to power off the MFC power domain. After PMU logic detects the change in LOCAL_PWR_CFG field, PMU starts power-down sequence. STATUS field of MFC_STATUS register indicates the completion of the power-down sequence.

To turn the power on, set LOCAL_PWR_CFG field of MFC_CONFIGURATION register to 0x7. After PMU logic detects the change in LOCAL_PWR_CFG field, PMU starts power up sequence. STATUS field of MFC_STATUS register indicates the power-up sequence.

NOTE: After PMU logic changes the power state requirement, ensure that you do not change it further until the previous requirement changes reflect in PMU.

8.4.2.2 Power on/off Sequence of TV/LCD/CAM

Except for the register name, the power control sequence of TV, LCD, and CAM is similar to that of MFC. .

8.4.2.3 Power on/off Sequence of ISP/GPS

Before accessing power state of ISP and GPS, it is necessary to reduce bus operating clock frequency of ISP and GPS to oscillator clock frequency. For that purpose, you should set CMU_SYSCLK_ISP_SYS_PWR_REG and CMU_SYSCLK_GPS_SYS_PWR_REG registers to "0". After turning the power on, you should first set all dividers in ISP and GPS and then restore bus clock source of ISP and GPS by setting MUX_ACLK_200_SUB_SEL, MUX_ACLK_400_MCUISP_SUB_SEL, and MUX_ACLK_266_GPS_SUB_SEL fields in CLK_SRC_TOP1 register to "1".

Except for the register name, the power control sequence of ISP/GPS is similar to that of MFC.

8.4.2.4 Power on/off Sequence of G3D

Before accessing power state of G3D, it is necessary to power on all Pixel Processors (PPs) inside Mali400.

Except for register name, the power control sequence of G3D is similar to that of MFC.

NOTE: Do not turn off VDD_G3D Regulator except in sleep mode. It may cause leakage of current.

8.4.2.5 Power on/off Sequence of Core 0

STATUS field of ARM_CORE0_STATUS register indicates the power state of Core 0 part of processor core. 0x3 indicates that Core 0 is fully powered up and 0x0 indicates that Core 0 is powered off. All the other values indicate that power on/off sequence is in progress.

By default, Core 0, which is part of processor core, is turned on.

1. Set the USE_DELAYED_RESET_ASSERTION field of ARM_CORE0_OPTION register to 0x1 to retain STANDBYWFI signal state.

2. Set LOCAL_PWR_CFG field of ARM_CORE0_CONFIGURATION register to 0x0 to turn the power off of Core 0 power domain.

After you detect the change in LOCAL_PWR_CFG field, PMU waits for the execution of wait-for-interrupt (WF) I instruction. During the execution of WFI instruction by Core 0, PMU starts power-down sequence. STATUS field of ARM_CORE0_STATUS register indicates the completion of the power-down sequence.

To turn the power on, set LOCAL_PWR_CFG field of ARM_CORE0_CONFIGURATION register to 0x3.

After you have detected the change in LOCAL_PWR_CFG field, PMU starts power up sequence. STATUS field of ARM_CORE_STATUS register indicates the completion of power-up sequence. After power-up sequence, if you set the USE_DELAYED_RESET_ASSERTION feature, set the USE_DELALYED_RESET_ASSERTION field of ARM_CORE0_OPTION register to 0x0.

NOTE:

1. Unlike other power domains, you should execute WFI instruction to turn processor core power down.
2. After PMU logic changes the power state requirement, ensure that you do not change it further until the previous requirement changes reflect in PMU.

8.4.2.6 Power on/off Sequence of Core 1, Core 2 and Core 3

Except, the register name, the power control sequence of Core 1, Core 2 and Core 3 are similar to that of Core 0.

8.4.2.7 Power on/off Sequence of CPU_L2_0

Power domain of L2 cache memory is divided into two areas. You can turn on and turn off each of these two areas independently. Additionally, the power state of L2 has another mode called retention. In retention mode, most of L2 circuit is not power supplied except for those for retaining cache contents.

STATUS field of ARM_L2_0_STATUS register indicates the power state of CPU_L2_0 part of processor core. 0x3 indicates that CPU_L2_0 is fully powered up. 0x0 indicates CPU_L2_0 is in retention mode when the USE_RETENTION field of ARM_L2_0_OPTION register is "1". 0x0 indicates CPU_L2_0 is in fully powered- off mode when the USE_RETENTION field of ARM_L2_0_OPTION register is "0". All the other values indicate that power on/off sequence is in progress.

By default, CPU_L2_0 part of processor core is powered on by default. Set LOCAL_PWR_CFG field of ARM_L2_0_CONFIGURATION register to 0x0 to power off CPU_L2_0 power domain, or set LOCAL_PWER_CFG field of ARM_L2_0_CONFIGURATION to 0x2 to put CPU_L2_0 in retention mode.

After you detect the change in LOCAL_PWR_CFG field, PMU starts the power-down sequence. STATUS field of ARM_L2_0_STATUS register indicates the completion of power down sequence.

To turn the power on, set LOCAL_PWR_CFG field of ARM_L2_0_CONFIGURATION register to 0x3. After you detect the change in LOCAL_PWR_CFG field, PMU starts power-up sequence. STATUS field of ARM_CORE_STATUS register indicates the completion of power-up sequence.

NOTE: After PMU logic changes the power state requirement, ensure that you do not change it further until the previous requirement changes reflect in PMU.

8.4.2.8 Power on/off Sequence of CPU_L2_1

Except for the register, the power control sequence of CPU_L2_1 is similar to that of CPU_L2_0.

8.5 System-level Power Control

This section includes:

- Overview on system-level power control
- Power-down Sequence
- System-level Power-down Configuration Registers
- Wake-up Sources

8.5.1 Overview on System-level Power Control

Exynos 4412 SCP provides five power modes, namely:

- NORMAL
- AFTR
- LPA
- DEEP-STOP
- SLEEP

Table 8-3 describes power modes.

In NORMAL mode, use module-based clock gating, block-based power gating, and frequency scaling to reduce power consumption. To reduce dynamic power consumption, clock gating disables clock input to specific module according to the operating scenario. You can perform clock gating in module-by-module basis.

To reduce static power consumption of a block or power domain (a group of modules), power gating disconnects a current path with a leakage. You can perform power gating in block-by-block basis.

Frequency scaling lowers the operating frequency to reduce dynamic power consumption.

Additionally, when you do not require the CPU, you can put it in power-down mode by executing WFI. This puts Cortex-A9 to freeze its clock internally to reduce dynamic power consumption.

When you require an additional power saving, use one of these system-level power control modes:

- In AFTR mode: AFTR stands for ARM Off and TOP Running. During this mode, PMU power gates Cortex-A9 core. The remaining parts of the chip are running. That is, the TOP modules are still in power-on state.
- In LPA mode: LPA stands for Low Power Audio playback. During this mode, PMU power gates Cortex-A9 core. The contents of L2 cache and TOP modules, except audio-related block, are in retention state, and Phase Look Loop (PLL) circuit blocks except Extra PLL (EPLL) for audio.
- In DEEP-STOP mode: PMU power gates not only the Cortex-A9 Core but also the remaining parts of the chip (except AudioSS, RTC, and ALIVE modules). However, all TOP modules, including the audio-related blocks, are in retention state, and PMU disables all PLLs and unnecessary oscillators.
- The regulator or PMIC turns off external power sources except the source of ALIVE block in this mode. PMU disables all PLLs and unnecessary oscillators. Static power consumption is very less in SLEEP mode. The only leakage power source is because of power supply to ALIVE block.

Change the clock source of all clock dividers inside DMC block to SCLK_MPLL before entering power-down modes.

Table 8-3 describes system-level power mode summary.

Table 8-3 System-Level Power Mode Summary

Domain	NORMAL	AFTR	LPA	D-STOP	SLEEP
CPU0	Pwr-on/off	Pwr-off	Pwr-off	Pwr-off	Externally pwr-off
CPU1	Pwr-on/off	Pwr-off	Pwr-off	Pwr-off	Externally pwr-off
CPU2	Pwr-on/off	Pwr-off	Pwr-off	Pwr-off	Externally pwr-off
CPU3	Pwr-on/off	Pwr-off	Pwr-off	Pwr-off	Externally pwr-off
CPU_L2_0	Pwr-on/off Retention	Retention	Retention	Retention	Externally pwr-off
CPU_L2_1	Pwr-on/off Retention	Retention	Retention	Retention	Externally pwr-off
CPU_ETC	Pwr-on	Pwr-off	Pwr-off	Pwr-off	Externally pwr-off
MFC	Pwr-on/off	Follows as left	Pwr-off	Pwr-off	Externally pwr-off
G3D	Pwr-on/off	Follows as left	Pwr-off	Pwr-off	Externally pwr-off
Audio Sub-system	Pwr-on/off	Follows as left	Pwr-on	Pwr-off	Externally pwr-off
LCD0	Pwr-on/off	Follows as left	Pwr-off	Pwr-off	Externally pwr-off
ISP	Pwr-on/off	Follows as left	Pwr-off	Pwr-off	Externally pwr-off
TV	Pwr-on/off	Follows as left	Pwr-off	Pwr-off	Externally pwr-off
CAM	Pwr-on/off	Follows as left	Pwr-off	Pwr-off	Externally pwr-off
MCT	Pwr-on	Follows as left	Pwr-on	Pwr-on	Externally pwr-off
TOP	Pwr-on	Pwr-on	Retention	Retention	Externally pwr-off
ALIVE	Pwr-on	Pwr-on	Pwr-on	Pwr-on	Pwr-on
RTC	Pwr-on	Pwr-on	Pwr-on	Pwr-on	Pwr-on
APLL	Pwr-on (en/dis)	Follows as left	Pwr-on (dis)	Pwr-on (dis)	Externally pwr-off
MPLL	Pwr-on (en/dis)	Follows as left	Pwr-on (dis)	Pwr-on (dis)	Externally pwr-off
EPLL	Pwr-on (en/dis)	Follows as left	Pwr-on (en)	Pwr-on (dis)	Externally pwr-off
VPLL	Pwr-on (en/dis)	Follows as left	Pwr-on (dis)	Pwr-on (dis)	Externally pwr-off
LPDDR_PHY	Pwr-on	Pwr-on	Pwr-on	Pwr-on	Externally pwr-off
HDMI_PHY	Pwr-on (en/dis)	Follows as left	Pwr-on (dis)	Pwr-on (dis)	Externally pwr-off
MIPI_PHY	Pwr-on	Pwr-on	Pwr-on	Pwr-on	Externally pwr-off
USB_PHY	Pwr-on (en/dis)	Follows as left	Pwr-on - disabled	Pwr-on (dis)	Externally pwr-off
ADC_PHY	Pwr-on (en/dis)	Follows as left	Pwr-on (standby/dis)	Pwr-on (standby/dis)	Pwr-on (dis)/ External pwr-off
I/O	Pwr-on	Pwr-on	Pwr-on	Pwr-on	Pwr-on

Power modes change when you enable C2C.

Table 8-4 describes C2C-enabled power modes Table 8-3.

In LPA, D-STOP and SLEEP modes, PMU either powers on or powers down DMC_BLK according to the C2C connection with modem C2C. The DMC_BLK that includes LPDDR_PHY, C2C and DREX2, and MPLL and OSC pad should be running to provide the Modem with DRAM access even when AP is in power-down mode. Therefore, PMU should either power on or power down DMC_BLK, MPLL, and OSC pad repeatedly according to the C2C connection state LPA, D-STOP, and SLEEP modes.

Table 8-4 describes C2C-enabled system-level power mode summary.

Table 8-4 C2C-enabled System-level Power Mode Summary

Domain	NORMAL	AFTR	LPA	D-STOP	SLEEP
CPU0	Pwr-on/off	Pwr-off	Pwr-off	Pwr-off	Externally pwr-off
CPU1	Pwr-on/off	Pwr-off	Pwr-off	Pwr-off	Externally pwr-off
CPU2	Pwr-on/off	Pwr-off	Pwr-off	Pwr-off	Externally pwr-off
CPU3	Pwr-on/off	Pwr-off	Pwr-off	Pwr-off	Externally pwr-off
CPU_L2_0	Pwr-on/off Retention	Retention	Retention	Retention	Externally pwr-off
CPU_L2_1	Pwr-on/off Retention	Retention	Retention	Retention	Externally pwr-off
CPU_ETC	Pwr-on	Pwr-off	Pwr-off	Pwr-off	Externally pwr-off
MFC	Pwr-on/off	Follows as left	Pwr-off	Pwr-off	Externally pwr-off
G3D	Pwr-on/off	Follows as left	Pwr-off	Pwr-off	Externally pwr-off
Audio Sub-system	Pwr-on/off	Follows as left	Pwr-on	Pwr-off	Externally pwr-off
LCD0	Pwr-on/off	Follows as left	Pwr-off	Pwr-off	Externally pwr-off
ISP	Pwr-on/off	Follows as left	Pwr-off	Pwr-off	Externally pwr-off
TV	Pwr-on/off	Follows as left	Pwr-off	Pwr-off	Externally pwr-off
CAM	Pwr-on/off	Follows as left	Pwr-off	Pwr-off	Externally pwr-off
MCT	Pwr-on	Follows as left	Pwr-on	Pwr-on	Externally pwr-off
TOP except DMC_BLK	Pwr-on	Pwr-on	Retention	Retention	Externally pwr-off
DMC_BLK (C2C, DREX2)	Pwr-on	Pwr-on	Pwr-on/Retention	Pwr-on/Retention	Pwr-on/Retention
ALIVE	Pwr-on	Pwr-on	Pwr-on	Pwr-on	Pwr-on
RTC	Pwr-on	Pwr-on	Pwr-on	Pwr-on	Pwr-on
APLL	Pwr-on (en/dis)	Follows as left	Pwr-on (dis)	Pwr-on (dis)	Externally pwr-off
MPLL	Pwr-on (en/dis)	Follows as left	Pwr-on (en/dis)	Pwr-on (en/dis)	Pwr-on (en/dis)
EPLL	Pwr-on (en/dis)	Follows as left	Pwr-on (en)	Pwr-on (dis)	Externally pwr-off
VPLL	Pwr-on (en/dis)	Follows as left	Pwr-on (dis)	Pwr-on (dis)	Externally pwr-off
LPDDR_PHY	Pwr-on	Pwr-on	Pwr-on	Pwr-on	Pow-on

Domain	NORMAL	AFTR	LPA	D-STOP	SLEEP
HDMI_PHY	Pwr-on (en/dis)	Follows as left	Pwr-on (dis)	Pwr-on (dis)	Externally pwr-off
MIPI_PHY	Pwr-on	Pwr-on	Pwr-on	Pwr-on	Externally pwr-off
USB_PHY	Pwr-on (en/dis)	Follows as left	Pwr-on - disabled	Pwr-on (dis)	Externally pwr-off
ADC_PHY	Pwr-on (en/dis)	Follows as left	Pwr-on (standby/dis)	Pwr-on (standby/dis)	Pwr-on (dis)/ External pwr-off
I/O	Pwr-on	Pwr-on	Pwr-on	Pwr-on	Pwr-on

8.5.2 Power-down Sequence

To enter system-level power-down mode:

1. Set operating frequency of G3D less than 200 MHz. For that purpose, access the CLK_DIV_G3D register at address 0x1003_C52C.

2. Change source clock of all clock dividers in DMC_BLK to SCLK_MPLL if you enable C2C. CLK_SRC_DMC handles this register at address 0x1004_0200.

3. Use the defined power-down mode. The system-level power-down configuration handles registers from offset at address 0x1000. Valid configurations can be found in the System-Level Power-down Configuration Registers section.

4. Enable system power down. Set only CENTRAL_SEQ_CONFIGURATION register if you disable C2C. Set both CENTRAL_SEQ_CONFIGURATION and CENTRAL_SEQ_CONFIGURATION_COREBLK registers if you enable C2C.

5. Disable interrupt-service routine for CPU. The processor core handles the interrupt-service routine. Refer to the Cortex-A9 and Cortex-A5 Technical Reference Manuals. When you use multiple CPU core, all the CPU cores should stop interrupt service.

6. Select wakeup sources to wake up PMU from the system-level power-down mode by setting WAKEUP_MASK and WAKEUP_MASK_COREBLK registers.

7. Execute WFI/WFE for all CPU cores. As soon as all the CPU cores in Exynos 4412 SCP enter STANDBY mode, refer to the Cortex-A9 and Cortex-A5 Technical Reference Manuals. PMU starts to power down.

8. If there are any pending interrupt events, ignore the WFI/WFE instruction (treated them as NOP instruction). Refer to the Cortex-A9 Technical Reference Manual for more information. The other procedure requirements to cancel the power-down sequence are:
   - Disable system power down by clearing CENTRAL_SEQ_CONFIGURATION and CENTRAL_SEQ_CONFIGURATION_COREBLK registers.
   - Clear WAKEUP_STAT and WAKEUP_MASK_COREBLK registers.
   - Enable interrupt-service routine for CPU.

Figure 8-1 illustrates procedures for system-level power mode.

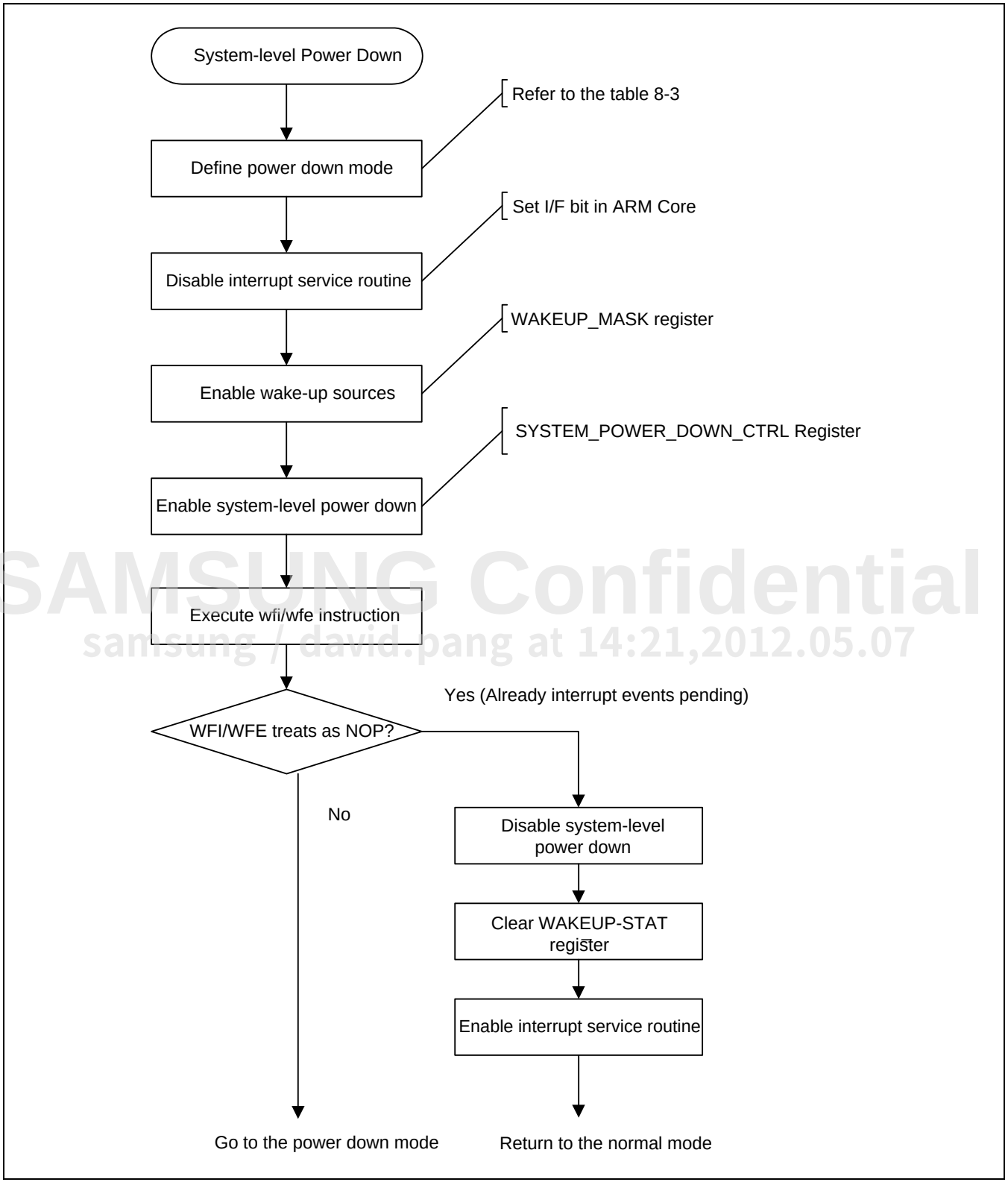


Figure 8-1 Procedures for System-Level Power Mode

8.5.3 System-Level Power-down Configuration Registers

System-level power-down configuration registers determine detailed actions during system-level power-down sequence. According to their configurations, PMU can put the system in AFTR, LPA, or SLEEP mode.

Table 8-5 describes the system-level power-down configuration registers.

Table 8-5 System-level Power-down Configuration Registers

Register	Bit	AFTR	STOP	LPA	D-STOP	SLEEP
ARM_CORE0_SYS_PWR_REG	[1:0]	0x0	0x3	0x0	0x0	0x2
ARM_CORE1_SYS_PWR_REG	[1:0]	0x0	0x3	0x0	0x0	0x2
ARM_CORE2_SYS_PWR_REG	[1:0]	0x0	0x3	0x0	0x0	0x2
ARM_CORE3_SYS_PWR_REG	[1:0]	0x0	0x3	0x0	0x0	0x2
ISP_ARM_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
ARM_COMMON_SYS_PWR_REG	[1:0]	0x0	0x3	0x0	0x0	0x2
ARM_L2_0_SYS_PWR_REG	[1:0]	0x0	0x3	0x0	0x0	0x3
ARM_L2_0_OPTION	[4]	0x1	0x1	0x1	0x1	0x0
ARM_L2_1_SYS_PWR_REG	[1:0]	0x0	0x3	0x0	0x0	0x3
ARM_L2_1_OPTION	[4]	0x1	0x1	0x1	0x1	0x0
CMU_ACLKSTOP_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_SCLKSTOP_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_RESET_SYS_PWR_REG	[0]	0x1	0x1	0x1	0x1	0x0
CMU_ACLKSTOP_COREBLK_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_SCLKSTOP_COREBLK_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_RESET_COREBLK_SYS_PWR_REG (C2C dis)	[0]	0x1	0x1	0x1	0x1	0x0
CMU_RESET_COREBLK_SYS_PWR_REG (C2C en)	[0]	0x1	0x1	0x1	0x1	0x1
APLL_SYSCLK_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
MPLL_SYSCLK_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
VPLL_SYSCLK_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
EPLL_SYSCLK_SYS_PWR_REG	[0]	0x1	0x0	0x1	0x0	0x0
MPLLUSER_SYSCLK_SYS_PWR_REG (C2C dis)	[0]	0x1	0x0	0x0	0x0	0x0
MPLLUSER_SYSCLK_SYS_PWR_REG (C2C en)	[0]	0x1	0x0	0x0	0x0	0x0
CMU_CLKSTOP_GPS_ALIVE_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_RESET_GPS_ALIVE_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_CLKSTOP_CAM_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_CLKSTOP_TV_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_CLKSTOP_MFC_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_CLKSTOP_G3D_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_CLKSTOP_LCD0_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_CLKSTOP_ISP_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0

Register	Bit	AFTR	STOP	LPA	D-STOP	SLEEP
CMU_CLKSTOP_MAUDIO_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_CLKSTOP_GPS_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x0
CMU_RESET_CAM_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
CMU_RESET_TV_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
CMU_RESET_MFC_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
CMU_RESET_G3D_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
CMU_RESET_LCD0_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
CMU_RESET_ISP_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
CMU_RESET_MAUDIO_SYS_PWR_REG	[0]	0x1	0x1	0x1	0x0	0x0
CMU_RESET_GPS_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
TOP_BUS_SYS_PWR_REG	[1:0]	0x3	0x0	0x0	0x0	0x0
TOP_RETENTION_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x1
TOP_PWR_SYS_PWR_REG	[1:0]	0x3	0x3	0x0	0x0	0x3
TOP_BUS_COREBLK_SYS_PWR_REG	[1:0]	0x3	0x0	0x0	0x0	0x0
TOP_RETENTION_COREBLK_SYS_PWR_REG (C2C dis)	[0]	0x1	0x1	0x0	0x0	0x1
TOP_RETENTION_COREBLK_SYS_PWR_REG (C2C en)	[0]	0x1	0x1	0x0	0x0	0x0
TOP_PWR_COREBLK_SYS_PWR_REG (C2C dis)	[1:0]	0x3	0x3	0x0	0x0	0x3
TOP_PWR_COREBLK_SYS_PWR_REG (C2C en)	[1:0]	0x3	0x3	0x0	0x0	0x0
LOGIC_RESET_SYS_PWR_REG	[0]	0x1	0x1	0x1	0x1	0x0
OSCCLK_GATE_SYS_PWR_REG	[0]	0x1	0x0	0x0	0x0	0x1
LOGIC_RESET_COREBLK_SYS_PWR_REG (C2C dis)	[0]	0x1	0x1	0x1	0x1	0x0
LOGIC_RESET_COREBLK_SYS_PWR_REG (C2C en)	[0]	0x1	0x1	0x1	0x1	0x1
OSCCLK_GATE_COREBLK_SYS_PWR_REG (C2C dis)	[0]	0x1	0x0	0x0	0x0	0x1
OSCCLK_GATE_COREBLK_SYS_PWR_REG (C2C en)	[0]	0x1	0x0	0x0	0x0	0x0
HSI_MEM_SYS_PWR_REG	[1:0]	0x3	0x3	0x0	0x0	0x0
HSI_MEM_OPTION	[4]	0x1	0x1	0x1	0x1	0x0
G2D_ACP_MEM_SYS_PWR_REG	[1:0]	0x3	0x3	0x0	0x0	0x0
G2D_ACP_MEM_OPTION	[4]	0x1	0x1	0x1	0x1	0x0
USBOTG_MEM_SYS_PWR_REG	[1:0]	0x3	0x3	0x0	0x0	0x0
USBOTG_MEM_OPTION	[4]	0x1	0x1	0x1	0x1	0x0
SDMMC_MEM_SYS_PWR_REG	[1:0]	0x3	0x3	0x0	0x0	0x0
SDMMC_MEM_OPTION	[4]	0x1	0x1	0x1	0x1	0x0

Register	Bit	AFTR	STOP	LPA	D-STOP	SLEEP
CSSYS_MEM_SYS_PWR_REG	[1:0]	0x3	0x3	0x0	0x0	0x0
CSSYS_MEM_OPTION	[4]	0x1	0x1	0x1	0x1	0x0
SECSS_MEM_SYS_PWR_REG	[1:0]	0x3	0x3	0x0	0x0	0x0
SECSS_MEM_OPTION	[4]	0x1	0x1	0x1	0x1	0x0
ROTATOR_MEM_SYS_PWR_REG	[1:0]	0x3	0x3	0x0	0x0	0x0
ROTATOR_MEM_OPTION	[4]	0x1	0x1	0x1	0x1	0x0
PAD_RETENTION_DRAM_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
PAD_RETENTION_MAUDIO_SYS_PWR_REG	[0]	0x1	0x1	0x1	0x0	0x0
PAD_RETENTION_GPIO_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
PAD_RETENTION_UART_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
PAD_RETENTION_MMCA_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
PAD_RETENTION_MMCB_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
PAD_RETENTION_EBIA_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
PAD_RETENTION_EBIB_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
PAD_RETENTION_GPIO_COREBLK_SYS_PWR_REG(C2C dis)	[0]	0x1	0x1	0x0	0x0	0x0
PAD_RETENTION_GPIO_COREBLK_SYS_PWR_REG(C2C en)	[0]	0x1	0x1	0x1	0x1	0x1
PAD_ISOLATION_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
PAD_ISOLATION_COREBLK_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
PAD_ALV_SEL_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
XUSBXTI_SYS_PWR_REG (NOTE)	[0]	0x1	0x0	0x1	0x0	0x0
XXTI_SYS_PWR_REG (NOTE)	[0]	0x1	0x0	0x1	0x0	0x0
EXT_REGULATOR_SYS_PWR_REG	[0]	0x1	0x1	0x1	0x1	0x0
GPIO_MODE_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
GPIO_MODE_COREBLK_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
GPIO_MODE_MAUDIO_SYS_PWR_REG	[0]	0x1	0x1	0x1	0x0	0x0
TOP_ASB_RESET_SYS_PWR_REG (C2C dis)	[0]	0x1	0x1	0x1	0x1	0x1
TOP_ASB_RESET_SYS_PWR_REG (C2C en)	[0]	0x1	0x1	0x1	0x1	0x0
TOP_ASB_ISOLATION_SYS_PWR_REG (C2C dis)	[0]	0x1	0x1	0x0	0x0	0x1
TOP_ASB_ISOLATION_SYS_PWR_REG (C2C en)	[0]	0x1	0x1	0x0	0x0	0x0
CAM_SYS_PWR_REG	[2:0]	0x7	0x7	0x0	0x0	0x0
TV_SYS_PWR_REG	[2:0]	0x7	0x7	0x0	0x0	0x0
MFC_SYS_PWR_REG	[2:0]	0x7	0x7	0x0	0x0	0x0
G3D_SYS_PWR_REG	[2:0]	0x7	0x7	0x0	0x0	0x0
LCD0_SYS_PWR_REG	[2:0]	0x7	0x7	0x0	0x0	0x0
ISP_SYS_PWR_REG	[2:0]	0x7	0x7	0x0	0x0	0x0

Register	Bit	AFTR	STOP	LPA	D-STOP	SLEEP
MAUDIO_SYS_PWR_REG	[2:0]	0x7	0x7	0x7	0x7	0x0
GPS_SYS_PWR_REG	[2:0]	0x7	0x7	0x0	0x0	0x0
GPS_ALIVE_SYS_PWR_REG	[2:0]	0x7	0x7	0x0	0x0	0x0
CMU_SYSCLK_ISP_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0
CMU_SYSCLK_GPS_SYS_PWR_REG	[0]	0x1	0x1	0x0	0x0	0x0

NOTE: If you use wake-up source-required oscillator clock, that is, HDMI CEC System Timer, you should set value to 0x1 in DEEP-STOP and SLEEP mode.

8.5.4 Wake-up Sources

Table 8-6 describes wake-up sources.

Table 8-6 Wake-up Sources

Wake-up Source	AFTR	LPA	DEEP-STOP	SLEEP
AudioSS	O	O	X	X
MMC0	O	O	O	X
MMC1	O	O	O	X
MMC2	O	O	O	X
MMC3	O	O	O	X
MCT (1)	O	O	O	X
External interrupt sources (EINT)	O	O	O	O
RTC Alarm	O	O	O	O
RTC TICK	O	O	O	O
KEYIF	O	O	O	O
HDMI CEC (2)	O	O	O	O
EGIC_nIRQ[0]	O	X	X	X
EGIC_nFIQ[0]	O	X	X	X
EGIC_nIRQ[1]	O	X	X	X
EGIC_nFIQ[1]	O	X	X	X
C2C_RESET_REQ	O	O	O	O
GPS	O	O	O	O

NOTE:

1. System Timer wakeup works only when clock source is alive.
2. HDMI CEC and MCT wakeups work only when you enable the main oscillator clock in system-level power-down mode.
3. You can use C2C_RESET_REQ as a wakeup source only when you set the ENABLE_C2C field of C2C_CTRL register to "1".

8.5.4.1 External Interrupts

External interrupts are the common wake-up source of LPA, DEEP-STOP, and SLEEP modes. The logic for external interrupt configuration such as polarity, edge/level sensitivity, and masking resides in the GPIO. You can modify the external interrupts through GPIO register setting before entering power-down modes. The external interrupt handling logic holds the information until user clears the information.

8.5.4.2 RTC Alarm

The Real Time Clock (RTC) has 32-bit counter to wake up the system after specified time. If the timer alarm triggers, the PMU wakes up the system and sets the RTL_ALARM field of WAKEUP_STAT register to 1. After the wake-up, you can find the cause of wake-up source through the WAKEUP_STAT register.

8.5.4.3 System Timer

System Timer is a new module in Exynos 4412 SCP. It supplements PWM timer, which suffers from accumulation of time deviation when you operate in variable tick mode. On the contrary, System Timer is free from such deviation and is a preferable choice for variable tick generation.

In LPA and DEEP-STOP mode, there can be no system clock when you power gate the TOP block.

Therefore, use RTC to generate timing tick instead of PWM timer. However, by using this clock, you cannot control the timing to meet exact 1 ms OS time tick as RTC clock does not have high resolution. On the other hand, System Timer has the function to generate interrupts at various intervals, and do not require manual setting. Thus, it does not wake up the chip too often, and provides accurate 1 ms timing ticks. It uses an external crystal clock, RTC clock, and the PMU-generated clock as clock input.

System Timer is not gated in LPA and DEEP-STOP mode. The wake up event from System Timer wakes up Exynos 4412 SCP from LPA and DEEP-STOP mode.

8.6 Reset

Exynos 4412 SCP has five types of resets and reset generator can place the system into one of five reset states.

The five reset states are:

- Hardware reset-You can generate the hardware reset when you drive XnRESET to low. Hardware reset is an uncompromised, ungated, and total reset that is used to drive Exynos 4412 SCP to a known initial state.
- Watchdog reset-It is the reset signal by watchdog timer
- Warm reset-The warm reset is generated when XnWRESET is driven to low.
- Software reset-It is the reset signal by setting special control register
- Wakeup reset-You can generate wakeup reset signal when you power down a module that has normal F/Fs. You can power up the module again by wakeup events. However, in sleep mode, you can generate wakeup reset to all modules that were powered off regardless of normal F/F or retention F/F.

The priorities of the five resets are:

1. Hardware reset
2. Watchdog reset
3. Warm reset
4. Software reset
5. Wakeup reset

Table 8-7 describes reset types and their coverage/effect.

Table 8-7 Reset Types and their Coverage/Effect

Reset Type	CPU	CPU Debugger	DRAM	Wait for Power Stabilization	Wait for Oscillator Clock Stabilization	All the Remaining Circuits
Hardware reset	Reset	Reset	Contents lost	Yes	Yes	Reset
Watchdog timer reset	Reset	Reset	Contents lost	No	No	Reset
Software reset	Reset	Contents kept	Contents kept with self-refresh mode	No	No	Reset
Warm reset	Reset	Contents kept	Contents kept with self-refresh mode	Yes	Yes	Reset
Wakeup reset	Reset	Reset	Contents kept with self-refresh mode	Yes (1)	Yes (2)	Reset

NOTE:

1. These resets are only for SLEEP wakeup.
2. You can use these resets when you disable the oscillator clock during system-level power-down mode

Table 8-8 describes the reset types and their effect on PMU.

Table 8-8 Reset Types and their Effect on PMU

Reset Type	INFORM0 INFORM1 INFORM2 INFORM3	INFORM4 INFORM5 INFORM6 INFORM7	PS_HOLD_ CONTROL	System-level Power Controller	All the Remaining
Hardware reset	Reset	Keep its value	Reset	Reset	Reset
Watchdog timer reset	Reset	Keep its value	Keep its value	Reset	Reset
Software reset	Keep its value	Keep its value	Keep its value	Reset	Reset
Warm reset	Keep its value	Keep its value	Keep its value	Reset	Reset
Wakeup reset	Keep its value	Keep its value	Keep its value	Keep its value	Reset

8.6.1 Hardware Reset

You can assert the hardware reset when you drives XnRESET pin to low. All units in the system, except RTC function modules are reset to known states.

The actions that take place during the hardware reset are:

- All internal registers and Cortex-A9 enter into their pre-defined reset state.
- All pins find their reset state.
- PMU asserts the XnRSTOUT when you drive XnRESET.

You can assert the hardware reset when an external source drives the XnRESET input pin low. XnRESET is non-maskable, and therefore is always applicable. Upon assertion of XnRESET, Exynos 4412 SCP enters into reset state regardless of the previous state. To assert the hardware, you should hold XnRESET long enough to allow internal stabilization and propagation of the reset state.

Caution: Power regulator for system should be stable prior to the de-assertion of XnRESET. If power regulator for system is not stable, it damages the Exynos 4412 SCP and its operation.

Figure 8-2 illustrates the clock behavior during the power-on reset sequence.

The crystal oscillator begins to oscillate within several milliseconds after the power supplies enough power-level to the Exynos 4412 SCP. CMU disables the internal PLLs after you assert the power-on reset. You should release XnRESET signal after power supply-level settles down fully. For the proper system operation, the Exynos 4412 SCP requires a hazard-free system clock (SYSCLK, ARMCLK, HCLK and PCLK) when you release the system reset (XnRESET). However, since CMU disables the PLLs, CMU feeds Fin (the direct external oscillator clock) directly to SYSCLK instead of the MPLL_CLK (PLL output) before the S/W configures the MPLLCON register to enable the operation of PLLs. If you require new P/M/S values, the S/W configures P/M/S field first, and then the PLL_EN field later.

The PLL begins the lockup sequence toward the new frequency only after the S/W configures the PLL with a new frequency-value. The CMU configures SYSCLK to be PLL output (MPLL_CLK) immediately after lock time.

You should be aware that the hardware does not explicitly add crystal oscillator settle-down time during the power-up sequence. The Exynos 4412 SCP assumes that the crystal oscillation settles during the power-supply settle-down period. However, to ensure the proper operation during wake-up from the STOP mode, the Exynos 4412 SCP explicitly adds the crystal oscillator settle-down time (you can program the wait-time by using the OSC_STABLE registers.) after wake-up from the STOP mode.

Exynos 4412 SCP has four PLLs, namely:

- APLL-Used to generate ARM clock
- MPLL-Used to generate system bus clock and several special clocks
- EPLL-Used to generate several special clocks
- VPLL-Used to generate video clocks

Figure 8-2 illustrates power on/off reset sequence.

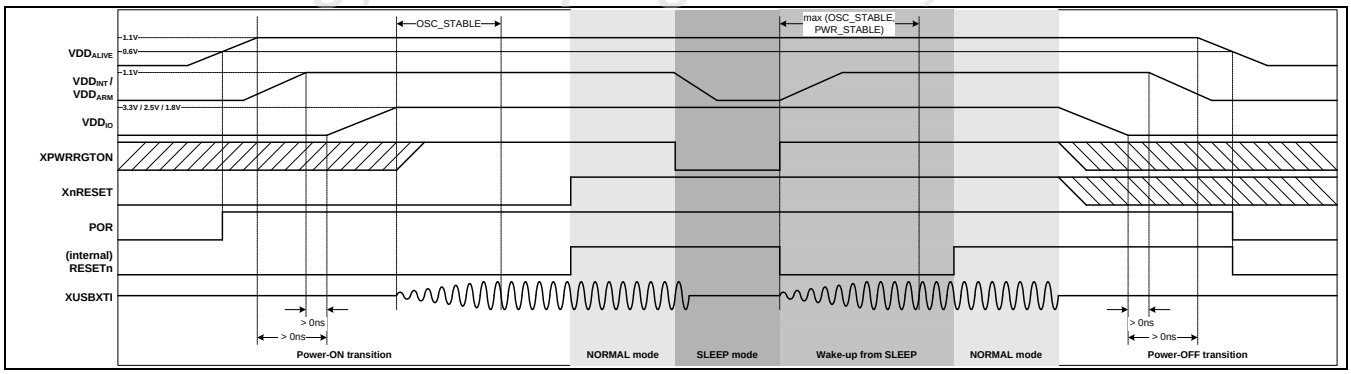


Figure 8-2 Power-on/off Reset Sequence

8.6.2 Watchdog Reset

The watchdog timer asserts watchdog reset when software fails to prevent the watchdog timer from timing out. In watchdog reset, PMU resets all units in Exynos 4412 SCP to their predefined reset states. The behavior is similar as hardware reset after watchdog timer asserts the watchdog reset.

The actions that occur during the watchdog reset, the following actions occur:

- All units enter into their pre-defined reset state.
- All pins find their reset state.
- You can assert the XnRSTOUT pin during watchdog reset.

You can activate watchdog reset in NORMAL and LPA mode because watchdog timer can expire with clock.

You can assert watchdog reset when you enable watchdog timer and reset. (WTCON[5] = 1, WTCON[0] = 1). Also, you can assert watchdog reset after expiring watchdog timer.

When you assert the watchdog reset:

1. Watchdog timer generates time-out signal.
2. PMU invokes reset signals and initializes internal IPs.
3. The PMU asserts reset including nRSTOUT until the reset counter, RST_STABLE, expires.

8.6.3 Software Reset

You can assert software reset when CPU Writes "1" to SWRESET register is in NORMAL mode.

During the software reset:

- All units enter into their pre-defined reset state.
- All pins find their reset state.
- The PMU asserts the XnRSTOUT pin during software reset.

When you assert the software reset:

1. PMU requests bus controller to finish current transactions.
2. Bus controller acknowledges to PMU after completing bus transactions.
3. PMU requests memory controller to enter into self refresh mode.
4. PMU waits for self refresh acknowledgment from the memory controller.
5. PMU asserts internal reset signals and XnRSTOUT and activates reset counter.
6. PMU expires the reset counter and then internal reset signals. PMU de-asserts XnRSTOUT.

8.6.4 Wakeup Reset

You can assert the wakeup reset when you power down a module that has normal F/Fs. Again, you can power up the module by wakeup events. If the module has only retention F/Fs, wakeup reset is not asserted. However, in sleep mode, you can generate wakeup reset to all modules that were powered off regardless of normal F/F or retention F/F.

Therefore, you can assert wakeup reset in NORMAL, AFTR, LPA, DEEP-STOP, and SLEEP mode.

In NORMAL mode, when you power down a sub-domain, and when you power up the sub-domain again, PMU asserts the wakeup reset in the sub-domain.

In AFTR, LPA, and DEEP-STOP mode, PMU asserts wakeup reset to Cortext-A9, since it powers up Cortex-A9 again when wakeup event occurs in these power modes. PMU also asserts wakeup reset to a sub-block that becomes power on after exiting from AFTR, LPA, and DEEP-STOP mode.

Finally, the PMU asserts the wakeup reset when the system wakes up form sleep mode by wakeup event.

8.7 IO Description

Table 8-9 describes Input/Output descriptions for signals.

Table 8-9 I/O Description

Signal	In/Out	Description	Pad
nRESET	In	–	XnRESET
nRSTOUT	Out	–	XnRSTOUT
PSHOLD	Out	–	XPSHOLD
PWRRGTON	Out	–	PWRRGTON

8.8 Register Description

8.8.1 Register Map Summary

- Base Address: 0x1002_0000

Register	Offset	Description	Reset Value
OM_STAT	0x0000	OM status register	0x0000_0000
LPI_MASK0	0x0004	Masks register for low-power interface handshaking. Modifies the fields of this register.	0x0000_0008
LPI_MASK1	0x0008	Masks register for low-power interface handshaking. Modifies the fields of this register.	0x0800_1E00
LPI_MASK2	0x000C	Masks register for low-power Interface handshaking. Modifies the fields of this register.	0x0000_0000
RTC_CLKO_SEL	0x0010	Controls RTCCLKOUT	0x0000_0000
GNSS_RTC_OUT_CTRL	0x0014	Controls GNSS_RTC_OUT	0x0000_0001
LPI_DENIAL_MASK0	0x0018	Masks register for LPI denial.	0x0000_47C0
LPI_DENIAL_MASK1	0x001C	Masks register for LPI denial.	0x19DF_9E00
LPI_DENIAL_MASK2	0x0020	Disables denial function of LPI.	0x0000_0007
C2C_CTRL	0x0024	0 = Disables C2C. 1 = Enables C2C	0x0000_0000
INTR_SSPREAD_ENABLE	0x0100	Control Interrupt spreading function.	0x0000_0000
INTR_SSPREAD_USE_STANDBYWFI	0x0104	Decide whether to use STANDBYWFI for deciding CPU's idle state.	0x0000_0003
INTR_SSPREAD_BLOCKING_DURATION	0x0108	Decides duration of time for blocking second interrupt.	0x0000_0080
CENTRAL_SEQ_CONFIGURATION	0x0200	Decides whether system-level low-power mode is used.	0x0001_0000
CENTRAL_SEQ_OPTION	0x0208	Sets control options for CENTRAL_SEQ	0x07C7_C000
CENTRAL_SEQ_CONFIGURATION_COREBLK	0x0240	Decides whether system-level low-power mode is used.	0x0001_0000
SWRESET	0x0400	Generates software reset	0x0000_0000
RST_STAT	0x0404	Resets status register	0x0000_0000
AUTOMATIC_WDT_RESET_DISABLE	0x0408	WDT reset disable register	0xFFFF_FFFF
MASK_WDT_RESET_REQUEST	0x040C	WDT reset request mask register	0xFFFF_FFFF
WAKEUP_STAT	0x0600	Wakes up status register	0x0000_0000
EINT_WAKEUP_MASK	0x0604	Configures EINT (external interrupt) mask	0x0000_0000
WAKEUP_MASK	0x0608	Configures wakeup source mask	0x0000_0000
WAKEUP_STAT_COREBLK	0x0620	Wakes up status register	0x0000_0000

Register	Offset	Description	Reset Value
WAKEUP_MASK_ COREBLK	0x0628	Configures wakeup source mask	0x0010_0000
HDMI_PHY_CONTROL	0x0700	HDMI PHY control register	0x0096_0000
USB_PHY_CONTROL	0x0704	USB PHY control register	0x0000_0000
HSIC_1_PHY_CONTROL	0x0708	HSIC_1 PHY control register	0x0000_0000
HSIC_2_PHY_CONTROL	0x070C	HSIC_2_PHY_CONTROL	0x0000_0000
MIPI_PHY0_CONTROL	0x0710	MIPI PHY control register	0x0000_0000
MIPI_PHY1_CONTROL	0x0714	MIPI PHY control register	0x0000_0000
ADC_PHY_CONTROL	0x0718	TS-ADC control register	0x0000_0001
BODY_BIAS_CON0	0x0780	INT ABB control register	0x0000_0000
BODY_BIAS_CON1	0x0784	MIF ABB control register	0x0000_0000
BODY_BIAS_CON2	0x0788	G3D ABB control register	0x0000_0000
BODY_BIAS_CON3	0x78C	ARM ABB control register	0x0000_0000
INFORM0	0x0800	Information register 0	0x0000_0000
INFORM1	0x0804	Information register 1	0x0000_0000
INFORM2	0x0808	Information register 2	0x0000_0000
INFORM3	0x080C	Information register 3	0x0000_0000
INFORM4	0x0810	Information register 4	0x0000_0000
INFORM5	0x0814	Information register 5	0x0000_0000
INFORM6	0x0818	Information register 6	0x0000_0000
INFORM7	0x081C	Information register 7	0x0000_0000
IROM_DATA_REG0	0x0980	Data register for IROM code.	0x0000_0000
IROM_DATA_REG1	0x0984	Data register for IROM code.	0x0000_0000
IROM_DATA_REG2	0x0988	Data register for IROM code.	0x0000_0000
IROM_DATA_REG3	0x098C	Data register for IROM code.	0x0000_0000
PMU_DEBUG	0x0A00	PMU debug register	0x0000_0000
ARM_CORE0_SYS_ PWR_REG	0x1000	Sets system-level low-power option	0xFFFF_FFFF
DIS_IRQ_ARM_CORE0_ LOCAL_SYS_PWR_REG	0x1004	Sets system-level low-power option	0xFFFF_FFFE
DIS_IRQ_ARM_CORE0_ CENTRAL_SYS_PWR_ REG	0x1008	Sets system-level low-power option	0xFFFF_FFFE
ARM_CORE1_SYS_ PWR_REG	0x1010	Sets system-level low-power option	0xFFFF_FFFF
DIS_IRQ_ARM_CORE1_ LOCAL_SYS_PWR_REG	0x1014	Sets system-level low-power option	0xFFFF_FFFE
DIS_IRQ_ARM_CORE1_ CENTRAL_SYS_PWR_	0x1018	Sets system-level low-power option	0xFFFF_FFFE

Register	Offset	Description	Reset Value
REG			
ISP_ARM_SYS_PWR_REG	0x1050	Sets system-level low-power option	0xFFFF_FFFF
DIS_IRQ_ISP_ARM_ LOCAL_SYS_PWR_REG	0x1054	Sets system-level low-power option	0xFFFF_FFFE
DIS_IRQ_ISP_ARM_ CENTRAL_SYS_PWR_ REG	0x1058	Sets system-level low-power option	0xFFFF_FFFE
ARM_COMMON_SYS_ PWR_REG	0x1080	Sets system-level low-power option	0xFFFF_FFFF
ARM_L2_0_SYS_PWR_ REG	0x10C0	Sets system-level low-power option	0xFFFF_FFFF
ARM_L2_1_SYS_PWR_ REG	0x10C4	Sets system-level low-power option	0xFFFF_FFFF
CMU_ACLKSTOP_SYS_ PWR_REG	0x1100	Sets system-level low-power option	0xFFFF_FFFF
CMU_SCLKSTOP_SYS_ PWR_REG	0x1104	Sets system-level low-power option	0xFFFF_FFFF
CMU_RESET_SYS_ PWR_REG	0x110C	Sets system-level low-power option	0xFFFF_FFFF
CMU_ACLKSTOP_ COREBLK_SYS_PWR_ REG	0x1110	Sets system-level low-power option	0xFFFF_FFFF
CMU_SCLKSTOP_ COREBLK_SYS_PWR_ REG	0x1114	Sets system-level low-power option	0xFFFF_FFFF
CMU_RESET_COREBLK_ SYS_PWR_REG	0x111C	Sets system-level low-power option	0xFFFF_FFFF
APLL_SYSCLK_SYS_ PWR_REG	0x1120	Sets system-level low-power option	0xFFFF_FFFF
MPLL_SYSCLK_SYS_ PWR_REG	0x1124	Sets system-level low-power option	0xFFFF_FFFF
VPLL_SYSCLK_SYS_ PWR_REG	0x1128	Sets system-level low-power option	0xFFFF_FFFF
EPLL_SYSCLK_SYS_ PWR_REG	0x112C	Sets system-level low-power option	0xFFFF_FFFF
MPLLUSER_SYSCLK_ SYS_PWR_REG	0x1130	Sets system-level low-power option	0xFFFF_FFFF
CMU_CLKSTOP_GPS_ ALIVE_SYS_PWR_REG	0x1138	Sets system-level low-power option	0xFFFF_FFFF
CMU_RESET_GPS_ ALIVE_SYS_PWR_REG	0x113C	Sets system-level low-power option	0xFFFF_FFFF
CMU_CLKSTOP_CAM_ SYS_PWR_REG	0x1140	Sets system-level low-power option	0xFFFF_FFFF

Register	Offset	Description	Reset Value
CMU_CLKSTOP_TV_SYS_PWR_REG	0x1144	Sets system-level low-power option	0xFFFF_FFFF
CMU_CLKSTOP_MFC_SYS_PWR_REG	0x1148	Sets system-level low-power option	0xFFFF_FFFF
CMU_CLKSTOP_G3D_SYS_PWR_REG	0x114C	Sets system-level low-power option	0xFFFF_FFFF
CMU_CLKSTOP_LCD0_SYS_PWR_REG	0x1150	Sets system-level low-power option	0xFFFF_FFFF
CMU_CLKSTOP_ISP_SYS_PWR_REG	0x1154	Sets system-level low-power option	0xFFFF_FFFF
CMU_CLKSTOP_MAUDIO_SYS_PWR_REG	0x1158	Sets system-level low-power option	0xFFFF_FFFF
CMU_CLKSTOP_GPS_SYS_PWR_REG	0x115C	Sets system-level low-power option	0xFFFF_FFFF
CMU_RESET_CAM_SYS_PWR_REG	0x1160	Sets system-level low-power option	0xFFFF_FFFF
CMU_RESET_TV_SYS_PWR_REG	0x1164	Sets system-level low-power option	0xFFFF_FFFF
CMU_RESET_MFC_SYS_PWR_REG	0x1168	Sets system-level low-power option	0xFFFF_FFFF
CMU_RESET_G3D_SYS_PWR_REG	0x116C	Sets system-level low-power option	0xFFFF_FFFF
CMU_RESET_LCD0_SYS_PWR_REG	0x1170	Sets system-level low-power option	0xFFFF_FFFF
CMU_RESET_ISP_SYS_PWR_REG	0x1174	Sets system-level low-power option	0xFFFF_FFFF
CMU_RESET_MAUDIO_SYS_PWR_REG	0x1178	Sets system-level low-power option	0xFFFF_FFFF
CMU_RESET_GPS_SYS_PWR_REG	0x117C	Sets system-level low-power option	0xFFFF_FFFF
TOP_BUS_SYS_PWR_REG	0x1180	Sets system-level low-power option	0xFFFF_FFFF
TOP_RETENTION_SYS_PWR_REG	0x1184	Sets system-level low-power option	0xFFFF_FFFF
TOP_PWR_SYS_PWR_REG	0x1188	Sets system-level low-power option	0xFFFF_FFFF
TOP_BUS_COREBLK_SYS_PWR_REG	0x1190	Sets system-level low-power option	0xFFFF_FFFF
TOP_RETENTION_COREBLK_SYS_PWR_REG	0x1194	Sets system-level low-power option	0xFFFF_FFFF
TOP_PWR_COREBLK_SYS_PWR_REG	0x1198	Sets system-level low-power option	0xFFFF_FFFF

Register	Offset	Description	Reset Value
LOGIC_RESET_SYS_ PWR_REG	0x11A0	Sets system-level low-power option	0xFFFF_FFFF
OSCCLK_GATE_SYS_ PWR_REG	0x11A4	Sets system-level low-power option	0xFFFF_FFFF
LOGIC_RESET_COREBLK _SYS_PWR_REG	0x11B0	Sets system-level low-power option	0xFFFF_FFFF
OSCCLK_GATE_ COREBLK_SYS_PWR_ REG	0x11B4	Sets system-level low-power option	0xFFFF_FFFF
OneNANDXL_MEM_SYS_ PWR_REG	0x11C0	Sets system-level low-power option	0xFFFF_FFFF
HSI_MEM_SYS_PWR_ REG	0x11C4	Sets system-level low-power option	0xFFFF_FFFF
G2D_ACP_MEM_SYS_ PWR_REG	0x11C8	Sets system-level low-power option	0xFFFF_FFFF
USBOTG_MEM_SYS_ PWR_REG	0x11CC	Sets system-level low-power option	0xFFFF_FFFF
SDMMC_MEM_SYS_ PWR_REG	0x11D0	Sets system-level low-power option	0xFFFF_FFFF
CSSYS_MEM_SYS_ PWR_REG	0x11D4	Sets system-level low-power option	0xFFFF_FFFF
SECSS_MEM_SYS_ PWR_REG	0x11D8	Sets system-level low-power option	0xFFFF_FFFF
ROTATOR_MEM_SYS_ PWR_REG	0x11DC	Sets system-level low-power option	0xFFFF_FFFF
PAD_RETENTION_DRAM _SYS_PWR_REG	0x1200	Sets system-level low-power option	0xFFFF_FFFF
PAD_RETENTION_ MAUDIO_SYS_PWR_REG	0x1204	Sets system-level low-power option	0xFFFF_FFFF
PAD_RETENTION_GPIO_ SYS_PWR_REG	0x1220	Sets system-level low-power option	0xFFFF_FFFF
PAD_RETENTION_UART_ SYS_PWR_REG	0x1224	Sets system-level low-power option	0xFFFF_FFFF
PAD_RETENTION_MMCA_ SYS_PWR_REG	0x1228	Sets system-level low-power option	0xFFFF_FFFF
PAD_RETENTION_MMCB_ SYS_PWR_REG	0x122C	Sets system-level low-power option	0xFFFF_FFFF
PAD_RETENTION_EBIA_ SYS_PWR_REG	0x1230	Sets system-level low-power option	0xFFFF_FFFF
PAD_RETENTION_EBIB_ SYS_PWR_REG	0x1234	Sets system-level low-power option	0xFFFF_FFFF
PAD_RETENTION_GPIO_ COREBLK_SYS_PWR_	0x123C	Sets system-level low-power option	0xFFFF_FFFF

Register	Offset	Description	Reset Value
REG			
PAD_ISOLATION_SYS_PWR_REG	0x1240	Sets system-level low-power option	0xFFFF_FFFF
PAD_ISOLATION_COREBLK_SYS_PWR_REG	0x1250	Sets system-level low-power option	0xFFFF_FFFF
PAD_ALV_SEL_SYS_PWR_REG	0x1260	Sets system-level low-power option	0xFFFF_FFFF
XUSBXTI_SYS_PWR_REG	0x1280	Sets system-level low-power option	0xFFFF_FFFF
XXTI_SYS_PWR_REG	0x1284	Sets system-level low-power option	0xFFFF_FFFF
EXT_REGULATOR_SYS_PWR_REG	0x12C0	Sets system-level low-power option	0xFFFF_FFFF
GPIO_MODE_SYS_PWR_REG	0x1300	Sets system-level low-power option	0xFFFF_FFFF
GPIO_MODE_COREBLK_SYS_PWR_REG	0x1320	Sets system-level low-power option	0xFFFF_FFFF
GPIO_MODE_MAUDIO_SYS_PWR_REG	0x1340	Sets system-level low-power option	0xFFFF_FFFF
TOP_ASB_RESET_SYS_PWR_REG	0x1344	Sets system-level low-power option	0xFFFF_FFFF
TOP_ASB_ISOLATION_SYS_PWR_REG	0x1348	Sets system-level low-power option	0xFFFF_FFFF
CAM_SYS_PWR_REG	0x1380	Sets system-level low-power option	0xFFFF_FFFF
TV_SYS_PWR_REG	0x1384	Sets system-level low-power option	0xFFFF_FFFF
MFC_SYS_PWR_REG	0x1388	Sets system-level low-power option	0xFFFF_FFFF
G3D_SYS_PWR_REG	0x138C	Sets system-level low-power option	0xFFFF_FFFF
LCD0_SYS_PWR_REG	0x1390	Sets system-level low-power option	0xFFFF_FFFF
ISP_SYS_PWR_REG	0x1394	Sets system-level low-power option	0xFFFF_FFFF
MAUDIO_SYS_PWR_REG	0x1398	Sets system-level low-power option	0xFFFF_FFFF
GPS_SYS_PWR_REG	0x139C	Sets system-level low-power option	0xFFFF_FFFF
GPS_ALIVE_SYS_PWR_REG	0x13A0	Sets system-level low-power option	0xFFFF_FFFF
DRAM_FREQ_DOWN_SYS_PWR_REG	0x13B0	Sets system-level low-power option	0xFFFF_FFFF
DDRPHY_DLLOFF_SYS_PWR_REG	0x13B4	Sets system-level low-power option	0xFFFF_FFFF
CMU_SYSCLK_ISP_SYS_PWR_REG	0x13B8	Sets system-level low-power option	0xFFFF_FFFF
CMU_SYSCLK_GPS_SYS_PWR_REG	0x13BC	Sets system-level low-power option	0xFFFF_FFFF

Register	Offset	Description	Reset Value
LPDDR_PHY_DLL_LOCK_SYS_PWR_REG	0x13C0	Sets system-level low-power option	0xFFFF_FFFF
ARM_CORE0_CONFIGURATION	0x2000	Configures power mode of ARM_CORE0	0x0000_0003
ARM_CORE0_STATUS	0x2004	Verifies power mode of ARM_CORE0	0x0003_0003
ARM_CORE0_OPTION	0x2008	Sets control options for ARM_CORE0	0x0101_0001
ARM_CORE1_CONFIGURATION	0x2080	Configures power mode of ARM_CORE1	0x0000_0003
ARM_CORE1_STATUS	0x2084	Verifies power mode of ARM_CORE1	0x0003_0003
ARM_CORE1_OPTION	0x2088	Sets control options for ARM_CORE0	0x0101_0001
ISP_ARM_OPTION	0x2288	Sets control options for ISP_ARM	0x0101_0000
ISP_ARM_CONFIGURATION	0x2280		0x0000_0001
ISP_ARM_STATUS	0x2284		0x0000_0001
ARM_COMMON_OPTION	0x2408	Sets control options for ARM_COMMON	0x0000_0001
ARM_L2_0_CONFIGURATION	0x2600	Configures power mode of ARM_L2_0	0x0000_0003
ARM_L2_0_STATUS	0x2604	Verifies power mode of ARM_L2_0	0x0000_0003
ARM_L2_0_OPTION	0x2608	Sets control options for ARM_L2_0	0x0000_0010
ARM_L2_1_CONFIGURATION	0x2620	Configures power mode of ARM_L2_1	0x0000_0003
ARM_L2_1_STATUS	0x2624	Verifies power mode of ARM_L2_1	0x0000_0003
ARM_L2_1_OPTION	0x2628	Sets control options for ARM_L2_1	0x0000_0010
DRAM_FREQ_DOWN_OPTION	0x29A8	Sets control options for DRAM_FREQ_DOWN	0x0000_0000
DDRPHY_DLLOFF_OPTION	0x2DC8	Sets control options for DDRPHY_DLLOFF	0x0000_0000
OneNANDXL_MEM_OPTION	0x2E08	Sets control options for OneNANDXL_MEM	0x0000_0010
HSI_MEM_OPTION	0x2E28	Sets control options for HSI_MEM	0x0000_0010
G2D_ACP_MEM_OPTION	0x2E48	Sets control options for G2D_ACP_MEM	0x0000_0010
USBOTG_MEM_OPTION	0x2E68	Sets control options for USBOTG_MEM	0x0000_0010
SDMMC_MEM_OPTION	0x2E88	Sets control options for SDMMC_MEM	0x0000_0010
CSSYS_MEM_OPTION	0x2EA8	Sets control options for CSSYS_MEM	0x0000_0010
SECSS_MEM_OPTION	0x2EC8	Sets control options for SECSS_MEM	0x0000_0010
ROTATOR_MEM_OPTION	0x2F48	Sets control options for ROTATOR_MEM	0x0000_0010
PAD_RETENTION_MAUDIO_OPTION	0x3028	Sets control options for PAD_RETENTION_MAUDIO	0x0000_0000
PAD_RETENTION_GPIO_OPTION	0x3108	Sets control options for PAD_RETENTION_GPIO	0x0000_0000

Register	Offset	Description	Reset Value
PAD_RETENTION_ UART_OPTION	0x3128	Sets control options for PAD_RETENTION_UART	0x0000_0000
PAD_RETENTION_ MMCA_OPTION	0x3148	Sets control options for PAD_RETENTION_MMCA	0x0000_0000
PAD_RETENTION_ MMCB_OPTION	0x3168	Sets control options for PAD_RETENTION_MMCB	0x0000_0000
PAD_RETENTION_EBIA_ OPTION	0x3188	Sets control options for PAD_RETENTION_EBIA	0x0000_0000
PAD_RETENTION_EBIB_ OPTION	0x31A8	Sets control options for PAD_RETENTION_EBIB	0x0000_0000
PAD_RETENTION_GPIO_ COREBLK_OPTION	0x31E8	Sets control options for PAD_RETENTION_GPIO_COREBLK	0x0000_0000
PS_HOLD_CONTROL	0x330C	PS_HOLD control register	0x0000_5200
XUSBXTI_ CONFIGURATION	0x3400	Configures the pad of XUSBXTI	0x0000_0001
XUSBXTI_STATUS	0x3404	Verifies the pad of XUSBXTI	0x0000_0001
XUSBXTI_DURATION	0x341C	Sets required time to stabilize XUSBXTI	0xFFF0_0000
XXTI_CONFIGURATION	0x3420	Configures the pad of XXTI	0x0000_0001
XXTI_STATUS	0x3424	Verifies the pad of XXTI	0x0000_0001
XXTI_DURATION	0x343C	Sets required time to stabilize XXTI	0xFFF0_0000
EXT_REGULATOR_ DURATION	0x361C	Sets required time to stabilize EXT_REGULATOR	0xFFF0_3FFF
CAM_CONFIGURATION	0x3C00	Configures power mode of CAM	0x0000_0007
CAM_STATUS	0x3C04	Verifies power mode of CAM	0x0006_0007
CAM_OPTION	0x3C08	Sets control options for CAM	0x0000_0001
TV_CONFIGURATION	0x3C20	Configures power mode of TV	0x0000_0007
TV_STATUS	0x3C24	Verifies power mode of TV	0x0006_0007
TV_OPTION	0x3C28	Sets control options for TV	0x0000_0001
MFC_CONFIGURATION	0x3C40	Configures power mode of MFC	0x0000_0007
MFC_STATUS	0x3C44	Verifies power mode of MFC	0x0006_0007
MFC_OPTION	0x3C48	Sets control options for MFC	0x0000_0001
G3D_CONFIGURATION	0x3C60	Configures power mode of G3D	0x0000_0007
G3D_STATUS	0x3C64	Verifies power mode of G3D	0x0006_0007
G3D_OPTION	0x3C68	Sets control options for G3D	0x0000_0001
LCD0_CONFIGURATION	0x3C80	Configures power mode of LCD0	0x0000_0007
LCD0_STATUS	0x3C84	Verifies power mode of LCD0	0x0006_0007
LCD0_OPTION	0x3C88	Sets control options for LCD0	0x0000_0001
ISP_CONFIGURATION	0x3CA0	Configures power mode of ISP	0x0000_0007
ISP_STATUS	0x3CA4	Verifies power mode of ISP	0x0006_0007

Register	Offset	Description	Reset Value
ISP_OPTION	0x3CA8	Sets control options for ISP	0x0000_0001
ISP_DURATION0	0x3CB8	Sets duration time for RESET, SCPRE, and SCALL	0xFFFF_FFFF
ISP_DURATION2	0x3CB8	Sets duration time for ISO	0xFFFF_FFFF
MAUDIO_ CONFIGURATION	0x3CC0	Configures power mode of MAUDIO	0x0000_0007
MAUDIO_STATUS	0x3CC4	Verifies power mode of MAUDIO	0x0006_0007
MAUDIO_OPTION	0x3CC8	Sets control options for MAUDIO	0x0000_0001
GPS_CONFIGURATION	0x3CE0	Configures power mode of GPS	0x0000_0007
GPS_STATUS	0x3CE4	Verifies power mode of GPS	0x0006_0007
GPS_OPTION	0x3CE8	Sets control options for GPS	0x0000_0001
GPS_ALIVE_ CONFIGURATION	0x3D00	Configures power mode of GPS	0x0000_0007
GPS_ALIVE_STATUS	0x3D04	Verifies power mode of GPS	0x0006_0007
GPS_ALIVE_OPTION	0x3D08	Sets control options for GPS	0x0000_0001

8.8.1.1 OM_STAT

- Base Address: 0x1002_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0x0
OM	[6:0]	R	Operation mode value	0x0

8.8.1.2 LPI_MASK0

- Base Address: 0x1002_0000
- Address = Base Address +0x0004, Reset Value = 0x0000_0008

Name	Bit	Type	Description	Reset Value
RSVD	[31:15]	–	Reserved	0x0
ATB_ISP_ARM	[14]	RW	Disables LPI of ATB asynchronous bridge for ISP_ARM.	0x0
RSVD	[13:11]	–	Reserved	0x0
FIMC_DRC	[10]	RW	Disables LPI of FIMC_DRC.	0x0
FIMC_LITE0	[9]	RW	Disables LPI of FIMC_LITE0.	0x0
FIMC_LITE1	[8]	RW	Disables LPI of FIMC_LITE1.	0x0
FIMC_ISP	[7]	RW	Disables LPI of FIMC_ISP.	0x0
FIMC_FD	[6]	RW	Disables LPI of FIMC_FD.	0x0
I2S0	[5]	RW	Even though I2S0 has AXI master interface. It has AHB bus master. I2S0 resides in MAUDIO block.	0x0
RSVD	[3]	–	Reserved	0x1
AHB_USBHS	[2]	RW	Disables LPI of AHB_USBHS	0x0
AHB_MMCHS	[1]	RW	Disables LPI of AHB_MMCHS	0x0
AHB_EXMHS	[0]	RW	Disables LPI of AHB_EXMHS	0x0

8.8.1.3 LPI_MASK1

- Base Address: 0x1002_0000
- Address = Base Address + 0x0008, Reset Value = 0x0800_1E00

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
MFC	[28]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x0
G3D	[27]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x1
RSVD	[26:25]	–	Reserved	0x0
MIXER	[24]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x0
VP	[23]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x0
ROTATOR	[22]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x0
RSVD	[21]	–	Reserved	0x0
FIMD0	[20]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x0
FIMC3	[19]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x0
FIMC2	[18]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x0
FIMC1	[17]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x0
FIMC0	[16]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x0
RP	[15]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x0
RSVD	[14:13]	–	Reserved	0x0
GPS	[12]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x1
PDMA1	[11]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x1
PDMA0	[10]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x1
MDMA	[9]	RW	When set to HIGH, PMU ignores bus down acknowledge from selected IPs.	0x1
RSVD	[8:0]	–	Reserved	0x0

8.8.1.4 LPI_MASK2

- Base Address: 0x1002_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
SSS	[2]	RW	Disables LPI of SSS.	0x0
G2D_ACP	[1]	RW	Disables LPI of G2D_ACP.	0x0
DREX2	[0]	RW	Disables LPI of DREX2.	0x0

8.8.1.5 RTC_CLKO_SEL

- Base Address: 0x1002_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
EN_OSC_EN	[0]	RW	When you activate EN_OSC_EN, RTCCLKOUT drives internal oscillator clock-enable value.	0x0

8.8.1.6 GNSS_RTC_OUT_CTRL

- Base Address: 0x1002_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
GNSS_RTC_ OUT_EN	[0]	RW	When you activate GNSS_RTC_OUT_EN, GNSS_RTC_OUT outputs oscillator clock.	0x1

8.8.1.7 LPI_DENIAL_MASK0

- Base Address: 0x1002_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_47C0

Name	Bit	Type	Description	Reset Value
RSVD	[31:15]	–	Reserved	0x0
ATB_ISP_ARM	[14]	RW	Disables LPI denial of ATB asynchronous bridge for ISP_ARM.	0x1
RSVD	[13:11]	–	Reserved	0x0
FIMC_DRC	[10]	RW	Disables LPI denial of FIMC_DRC.	0x1
FIMC_LITE0	[9]	RW	Disables LPI denial of FIMC_LITE0.	0x1
FIMC_LITE1	[8]	RW	Disables LPI denial of FIMC_LITE1.	0x1
FIMC_ISP	[7]	RW	Disables LPI denial of FIMC_ISP.	0x1
FIMC_FD	[6]	RW	Disables LPI denial of FIMC_FD.	0x1
RSVD	[5:0]	–	Reserved	0x0

8.8.1.8 LPI_DENIAL_MASK1

- Base Address: 0x1002_0000
- Address = Base Address + 0x001C, Reset Value = 0x19DF_9E00

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
MFC	[28]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
G3D	[27]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
RSVD	[26:25]	–	Reserved	0x0
MIXER	[24]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
VP	[23]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
ROTATOR	[22]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
RSVD	[21]	–	Reserved	0x0
FIMD0	[20]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
FIMC3	[19]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
FIMC2	[18]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
FIMC1	[17]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
FIMC0	[16]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
RP	[15]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
RSVD	[14:13]	–	Reserved	0x0
GPS	[12]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
PDMA1	[11]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
PDMA0	[10]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
MDMA	[9]	RW	When set to HIGH, PMU ignores bus down denial from selected IPs.	0x1
RSVD	[8:0]	–	Reserved	0x0

8.8.1.9 LPI_DENIAL_MASK2

- Base Address: 0x1002_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
SSS	[2]	RW	Disables denial LPI of SSS.	0x1
G2D_ACP	[1]	RW	Disables denial LPI of G2D_ACP.	0x1
DREX2	[0]	RW	Disables denial LPI of DREX2.	0x1

8.8.1.10 C2C_CTRL

- Base Address: 0x1002_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
ENABLE_C2C	[0]	RW	0 = Disables C2C 1 = Enables C2C	0x0

8.8.1.11 INTR_SPREAD_ENABLE

- Base Address: 0x1002_0000
- Address = Base Address + 0x0100, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
ENABLE	[3:0]	RW	Enable bit for corresponding CPU	0x0

8.8.1.12 INTR_SPREAD_USE_STANDBYWFI

- Base Address: 0x1002_0000
- Address = Base Address + 0x0104, Reset Value = 0x0000_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
SPREAD_USE_ STANDBYWFI	[3:0]	RW	STANDBYWFI for corresponding CPU is used to decide idle state.	0x3

8.8.1.13 INTR_SPREAD_BLOCKING_DURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x0108, Reset Value = 0x0000_0080

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0x0
DURATION	[11:0]	RW	Number of clock cycles for preventing additional interrupt is given. Default value guarantee 5.3us	0x80

8.8.1.14 CENTRAL_SEQ_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x0200, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0x0
SYS_PWR_CFG	[16]	RWX	Decides whether you can use the system-level power mode. HIGH: Disables system-level low-power mode. LOW: Enables system-level low-power mode. When system enters low-power mode, PMU clears this field automatically to HIGH.	0x1
RSVD	[15:0]	–	Reserved	0x0

8.8.1.15 CENTRAL_SEQ_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x0208, Reset Value = 0x07C7_C000

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0x0
USE_STANDBYWFE_ISP_ARM	[26]	RWX	Uses STANDBYWFE for judging whether ISP_ARM is ready for entering low-power mode. When you set USE_STANDBYWFI and USE_STANDBYWFE to HIGH, either STANDBYWFI or STANDBYWFE can indicate idle ARM state. When you set USE_STANDBYWFI and USE_STANDBYWFE to LOW, PMU assumes that ARM is ready without regarding to STANDBYWFI and STANDBYWFE value. Updates changes on USE_STANDBYWFI field of FSYS_ARM_OPTION register automatically in this field.	0x1
USE_STANDBYWFE_ARM_CORE1	[25]	RWX	Uses STANDBYWFE for judging whether ARM CORE 1 is ready for entering low-power mode. When you set USE_STANDBYWFI and USE_STANDBYWFE to HIGH, either STANDBYWFI or STANDBYWFE can indicate idle ARM state. When you set USE_STANDBYWFI and USE_STANDBYWFE to LOW, PMU assumes that ARM is ready without regarding to STANDBYWFI and STANDBYWFE value. Updates changes on USE_STANDBYWFI field of ARM_CORE0_OPTION register automatically in this field.	0x1
USE_STANDBYWFE_ARM_CORE0	[24]	RWX	Uses STANDBYWFE for judging whether ARM CORE 0 is ready for entering low-power mode. When you set USE_STANDBYWFI and USE_STANDBYWFE to HIGH, either STANDBYWFI or STANDBYWFE can indicate idle ARM state. When you set USE_STANDBYWFI and USE_STANDBYWFE to LOW, PMU assumes that ARM is ready without regarding to STANDBYWFI and STANDBYWFE value. Updates changes on USE_STANDBYWFI field of ARM_CORE0_OPTION register automatically in this field.	0x1
USE_STANDBYWFE_ARM_CORE3	[23]	RWX	Uses STANDBYWFE for judging whether ARM CORE 3 is ready for entering low-power mode. When you set USE_STANDBYWFI and USE_STANDBYWFE to HIGH, either STANDBYWFI or STANDBYWFE can indicate idle ARM state. When you set USE_STANDBYWFI and USE_STANDBYWFE to LOW, PMU assumes that ARM is ready without regarding to STANDBYWFI	0x1

Name	Bit	Type	Description	Reset Value
			and STANDBYWFE value. Updates changes on USE_STANDBYWFI field of ARM_CORE0_OPTION register automatically in this field.	
USE_STANDBYWFE_ARM_CORE2	[22]	RWX	Uses STANDBYWFE for judging whether ARM CORE 2 is ready for entering low-power mode. When you set USE_STANDBYWFI and USE_STANDBYWFE to HIGH, either STANDBYWFI or STANDBYWFE can indicate idle ARM state. When you set USE_STANDBYWFI and USE_STANDBYWFE to LOW, PMU assumes that ARM is ready without regarding to STANDBYWFI and STANDBYWFE value. Updates changes on USE_STANDBYWFI field of ARM_CORE0_OPTION register automatically in this field.	0x1
RSVD	[21:19]	–	Reserved	0x0
USE_STANDBYWFI_ISP_ARM	[18]	RWX	Uses STANDBYWFI for judging whether ISP_ARM is ready for entering low-power mode. You should activate either USE_STANDBYWFI or USE_STANDBYWFE at a time. Updates changes on USE_STANDBYWFI field of ISP_ARM_OPTION register automatically in this field.	0x1
USE_STANDBYWFI_ARM_CORE1	[17]	RWX	Uses STANDBYWFI for judging whether ARM CORE 1 is ready for entering low-power mode. When you set USE_STANDBYWFI and USE_STANDBYWFE to HIGH, either STANDBYWFI or STANDBYWFE can indicate idle ARM state. When you set USE_STANDBYWFI and USE_STANDBYWFE to LOW, PMU assumes that ARM is ready without regarding to STANDBYWFI and STANDBYWFE value. Updates changes on USE_STANDBYWFI field of ARM_CORE0_OPTION register automatically in this field.	0x1
USE_STANDBYWFI_ARM_CORE0	[16]	RWX	Uses STANDBYWFI for judging whether ARM CORE 0 is ready for entering low-power mode. When you set USE_STANDBYWFI and USE_STANDBYWFE to HIGH, either STANDBYWFI or STANDBYWFE can indicate idle ARM state. When you set USE_STANDBYWFI and USE_STANDBYWFE to LOW, PMU assumes that ARM is ready without regarding to STANDBYWFI and STANDBYWFE value. Updates changes on USE_STANDBYWFI field of ARM_CORE0_OPTION register automatically in this field.	0x1
USE_STANDBYWFI	[15]	RWX	Uses STANDBYWFI for judging whether ARM	0x1

Name	Bit	Type	Description	Reset Value
_ARM_CORE3			CORE 3 is ready for entering low-power mode. When you set USE_STANDBYWFI and USE_STANDBYWFE to HIGH, either STANDBYWFI or STANDBYWFE can indicate idle ARM state. When you set USE_STANDBYWFI and USE_STANDBYWFE to LOW, PMU assumes that ARM is ready without regarding to STANDBYWFI and STANDBYWFE value. Updates changes on USE_STANDBYWFI field of ARM_CORE0_OPTION register automatically in this field.	
USE_STANDBYWFI _ARM_CORE2	[14]	RWX	Uses STANDBYWFI for judging whether ARM CORE 2 is ready for entering low-power mode. When you set USE_STANDBYWFI and USE_STANDBYWFE to HIGH, either STANDBYWFI or STANDBYWFE can indicate idle ARM state. When you set USE_STANDBYWFI and USE_STANDBYWFE to LOW, PMU assumes that ARM is ready without regarding to STANDBYWFI and STANDBYWFE value. Updates changes on USE_STANDBYWFI field of ARM_CORE0_OPTION register automatically in this field.	0x1
RSVD	[13:10]	–	Reserved	0x0
FAST_PWUP	[9]	RW	When you set to HIGH, all sub-power domains are turned on at the same time. When you set to LOW, each sub-block is tuned on one by one.	0x0
FAST_PWDN	[8]	RW	When you set to HIGH, all sub-power domains are turned off at the same time. When you set to LOW, each sub-block is tuned off one by one.	0x0
RSVD	[7:0]	–	Reserved	0x0

8.8.1.16 CENTRAL_SEQ_CONFIGURATION_COREBLK

- Base Address: 0x1002_0000
- Address = Base Address + 0x0240, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0x0
SYS_PWR_CFG	[16]	RWX	Decides whether PMU uses the system-level power mode. HIGH: Disables system-level low-power mode. LOW: Enables system-level low-power mode. When system enters low-power mode, PMU clears this field automatically to HIGH.	0x1
RSVD	[15:0]	–	Reserved	0x0

8.8.1.17 SWRESET

- Base Address: 0x1002_0000
- Address = Base Address + 0x0400, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved (should be zero)	0x0
SYSTEM	[0]	RWX	Software reset for whole system. After the software reset process is over, PMU clears this field automatically. 0 = No effect 1 = Reset	0x0

8.8.1.18 RST_STAT

- Base Address: 0x1002_0000
- Address = Base Address + 0x0404, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0x0
SWRESET	[29]	R	Sets this field automatically when you give software reset. Clears this field when you activate other reset source.	0x0
WRESET	[28]	R	Sets this field automatically when you give warm reset. Clears this field when you activate other reset source.	0x0
RSVD	[27:26]	–	Reserved	0x0
ISP_ARM_ WDTRESET	[25]	R	Sets this field automatically when you give ISP_ARM watchdog timer reset. Clears this field when you activate other reset source.	0x0
F4D_WDTRESET1	[24]	R	Sets this field automatically when you give F4D watchdog timer reset1. Clears this field when you activate other reset source.	0x0
F4D_WDTRESET0	[23]	R	Sets this field automatically when you give F4D watchdog timer reset0. Clears this field when you activate other reset source.	0x0
SYS_WDTRESET	[20]	R	Sets this field automatically when you give timer reset. Clears this field when you activate other reset source.	0x0
RSVD	[19:17]	–	Reserved	0x0
PINRESET	[16]	R	Sets this field automatically when you give XnRESET reset. Clears this field when you activate other reset source.	0x0
F4D_WDTRESET3	[15]	R	Sets this field automatically when you give F4D watchdog timer reset3. Clears this field when you activate other reset source.	0x0

F4D_WDTRESET2	[14]	R	Sets this field automatically when you give F4D watchdog timer reset2. Clears this field when you activate other reset source.	0x0
RSVD	[13:0]	–	Reserved	0x0

8.8.1.19 AUTOMATIC_WDT_RESET_DISABLE

- Base Address: 0x1002_0000
- Address = Base Address + 0x0408, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0x1F
ISP_ARM_ WDTRESET	[26]	RW	When you set this field, the PMU disables corresponding ISP_ARM_WDTRESET when the corresponding processor is in reset state.	0x1
RSVD	[25]	–	Reserved	0x1
F4D_WDTRESET1	[24]	RW	When you set this field, the PMU disables corresponding F4D_WDTRESET1 when the corresponding processor is in reset state.	0x1
F4D_WDTRESET0	[23]	RW	When you set this field, the PMU disables corresponding F4D_WDTRESET0 when the corresponding processor is in reset state.	0x1
CORETIMER_ WDTRESET1	[22]	RW	When you set this field, the PMU disables corresponding CORETIMER_WDTRESET1 when the corresponding processor is in reset state.	0x1
CORETIMER_ WDTRESET0	[21]	RW	When you set this field, the PMU disables corresponding CORETIMER_WDTRESET0 when the corresponding processor is in reset state.	0x1
SYS_WDTRESET	[20]	RW	When you set this field, the PMU disables corresponding SYS_WDTRESET when the corresponding processor is in reset state.	0x1
F4D_WDTRESET3	[19]	RW	When you set this field, the PMU disables corresponding F4D_WDTRESET3 when the corresponding processor is in reset state.	0x1
F4D_WDTRESET2	[18]	RW	When you set this field, the PMU disables corresponding F4D_WDTRESET2 when the corresponding processor is in reset state.	0x1
CORETIMER_ WDTRESET3	[17]	RW	When you set this field, the PMU disables corresponding CORETIMER_WDTRESET3 when the corresponding processor is in reset state.	0x1
CORETIMER_ WDTRESET2	[16]	RW	When you set this field, the PMU disables corresponding CORETIMER_WDTRESET2 when the corresponding processor is in reset state.	0x1
RSVD	[15:0]	–	Reserved	0xFFFF

8.8.1.20 MASK_WDT_RESET_REQUEST

- Base Address: 0x1002_0000
- Address = Base Address + 0x040C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0x1F
ISP_ARM_ WDTRESET	[26]	RW	When you set ISP_ARM_WDTRESET, PMU ignores the watchdog timer reset.	0x1
RSVD	[25]	–	Reserved	0x1
F4D_WDTRESET1	[24]	RW	When you set F4D_WDTRESET1, PMU ignores the watchdog timer reset.	0x1
F4D_WDTRESET0	[23]	RW	When you set F4D_WDTRESET0, PMU ignores the watchdog timer reset.	0x1
CORETIMER_ WDTRESET1	[22]	RW	When you set CORETIMER_WDTRESET1, PMU ignores the watchdog timer reset.	0x1
CORETIMER_ WDTRESET0	[21]	RW	When you set CORETIMER_WDTRESET0, PMU ignores the watchdog timer reset.	0x1
SYS_WDTRESET	[20]	RW	When you set SYS_WDTRESET, PMU ignores the watchdog timer reset.	0x1
F4D_WDTRESET3	[19]	RW	When you set F4D_WDTRESET3, PMU ignores the watchdog timer reset.	0x1
F4D_WDTRESET2	[18]	RW	When you set F4D_WDTRESET2, PMU ignores the watchdog timer reset.	0x1
CORETIMER_ WDTRESET3	[17]	RW	When you set CORETIMER_WDTRESET3, PMU ignores the watchdog timer reset.	0x1
CORETIMER_ WDTRESET2	[16]	RW	When you set CORETIMER_WDTRESET2, PMU ignores the watchdog timer reset.	0x1
RSVD	[19:0]	–	Reserved	0xFFFF

8.8.1.21 WAKEUP_STAT

- Base Address: 0x1002_0000
- Address Base Address + 0x0600, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
WAKEUP	[31]	R	This field indicates wakeup source that caused system to start working again. Automatically sets when you assert any reset source.	0x0
RSVD	[30:26]	–	Reserved	0x0
EXT_GIC_FIQ3 (NOTE)	[25]	RWX	Wake-up by FIQ3 of External GIC Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
EXT_GIC_IRQ3 (NOTE)	[24]	RWX	Wake-up by IRQ3 of External GIC Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
EXT_GIC_FIQ2 (NOTE)	[23]	RWX	Wake-up by FIQ2 of External GIC Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
EXT_GIC_IRQ2 (NOTE)	[22]	RWX	Wake-up by IRQ2 of External GIC Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
GPS_ALIVE	[21]	RWX	Wake-up by GPS_ALIVE	0x0
C2C_RESET_REQ	[20]	RWX	Wake-up by C2C_RESET_REQ of C2C	0x0
EXT_GIC_FIQ1 (NOTE)	[19]	RWX	Wake-up by FIQ1 of External GIC Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
EXT_GIC_IRQ1 (NOTE)	[18]	RWX	Wake-up by IRQ1 of External GIC Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
EXT_GIC_FIQ0 (NOTE)	[17]	RWX	Wake-up by FIQ0 of External GIC Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
EXT_GIC_IRQ0 (NOTE)	[16]	RWX	Wake-up by IRQ0 of External GIC Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
CEC	[15]	RWX	Wake-up by HDMI-CEC Clears this field automatically when system starts to enter system-level power-down mode operation. Also,	0x0

Name	Bit	Type	Description	Reset Value
			clears this field by writing 0 to this field.	
MCT	[14]	RWX	Wake-up by MCT Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
AUDIOSS	[13]	RWX	Wake-up by Audio sub-system Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
MMC3	[12]	RWX	Wake-up by MMC3 Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
MMC2	[11]	RWX	Wake-up by MMC2 Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
MMC1	[10]	RWX	Wake-up by MMC1 Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
MMC0	[9]	RWX	Wake-up by MMC0 Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
RSVD	[8:6]	–	Reserved	0x0
KEY	[5]	RWX	Wake-up by KEY I/F Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
TS1	[4]	RWX	Wake-up by TSADC1 Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
TS0	[3]	RWX	Wake-up by TSADC0 Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
RTC_TICK	[2]	RWX	Wake-up by RTC-TICK Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0
RTC_ALARM	[1]	RWX	Wake-up by RTC-Alarm Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0

Name	Bit	Type	Description	Reset Value
EINT	[0]	RWX	Wake-up by EINT Clears this field automatically when system starts to enter system-level power-down mode operation. Also, clears this field by writing 0 to this field.	0x0

NOTE: AP does not support the external GIC wake-up. That is, the field is meaningless.

8.8.1.22 EINT_WAKEUP_MASK

- Base Address: 0x1002_0000
- Address = Base Address + 0x0604, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
EINT_WAKEUP_ MASK	[31:0]	RW	External interrupt wake-up mask EINT[31:0] 0 = Uses as a wake-up source 1 = The PMU disables the external interrupt.	0x0

8.8.1.23 WAKEUP_MASK

- Base Address: 0x1002_0000
- Address = Base Address + 0x0608, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
USE_LEVEL_TRIGGER	[30]	RW	Uses level-trigger mode for wakeup request signal. By default, PMU uses edge-trigger.	0x0
RSVD	[29:26]	–	Reserved	0x0
EXT_GIC_FIQ3 (NOTE)	[25]	RW	Wake-up mask for FIQ3 of External GIC 0 = Pass 1 = Mask	0x0
EXT_GIC_IRQ3 (NOTE)	[24]	RW	Wake-up mask for IRQ3 of External GIC 0 = Pass 1 = Mask	0x0
EXT_GIC_FIQ2 (NOTE)	[23]	RW	Wake-up mask for FIQ2 of External GIC 0 = Pass 1 = Mask	0x0
EXT_GIC_IRQ2 (NOTE)	[22]	RW	Wake-up mask for IRQ2 of External GIC 0 = Pass 1 = Mask	0x0
GPS_ALIVE	[21]	RW	Wake-up mask for GPS_ALIVE wakeup source 0 = Pass 1 = mask	0x1
C2C_RESET_REQ	[20]	RW	Wake-up mask for C2C_RESET_REQ wakeup source 0 = Pass 1 = Mask	0x1
EXT_GIC_FIQ1 (NOTE)	[19]	RW	Wake-up mask for FIQ1 of External GIC 0 = Pass 1 = Mask	0x0
EXT_GIC_IRQ1 (NOTE)	[18]	RW	Wake-up mask for IRQ1 of External GIC 0 = Pass 1 = Mask	0x0
EXT_GIC_FIQ0 (NOTE)	[17]	RW	Wake-up mask for FIQ0 of External GIC 0 = Pass 1 = Mask	0x0
EXT_GIC_IRQ0 (NOTE)	[16]	RW	Wake-up mask for IRQ0 of External GIC 0 = Pass 1 = Mask	0x0
CEC	[15]	RW	Wake-up mask for HDMI-CEC 0 = Pass 1 = Mask	0x0
ST	[14]	RW	Wake-up mask for system timer	0x0

Name	Bit	Type	Description	Reset Value
			0 = Pass 1 = Mask	
MAUDIO	[13]	RW	Wake-up mask for audio sub-system 0 = Pass 1 = Mask	0x0
MMC3	[12]	RW	Wake-up mask for MMC3 0 = Pass 1 = Mask	0x0
MMC2	[11]	RW	Wake-up mask for MMC2 0 = Pass 1 = Mask	0x0
MMC1	[10]	RW	Wake-up mask for MMC1 0 = Pass 1 = Mask	0x0
MMC0	[9]	RW	Wake-up mask for MMC0 0 = Pass 1 = Mask	0x0
RSVD	[8:6]	–	Reserved	0x0
KEY	[5]	RW	Wake-up mask for KEY I/F 0 = Pass 1 = Mask	0x0
TS1	[4]	RW	Wake-up mask for TSADC1 0 = Pass 1 = Mask	0x0
TS0	[3]	RW	Wake-up mask for TSADC0 0 = Pass 1 = Mask	0x0
RTC_TICK	[2]	RW	Wake-up mask for RTC-TICK 0 = Pass 1 = Mask	0x0
RTC_ALARM	[1]	RW	Wake-up mask for RTC-Alarm 0 = Pass 1 = Mask	0x0
RSVD	[0]	–	Reserved	0x0

NOTE: AP does not support the external GIC wake-up. That is, the field is meaningless.

8.8.1.24 WAKEUP_STAT_COREBLK

- Base Address: 0x1002_0000
- Address = Base Address + 0x0620, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
WAKEUP	[31]	R	This field indicates wakeup source that causes system to start working again. Automatically sets when any reset source is asserted.	0x0
RSVD	[30:21]	–	Reserved	0x0
C2C	[20]	RWX	Wake-up by C2C_WAKEUP The C2C_WAKEUP wakes up only DMC_BLK from retention state.	0x0
RSVD	[19:0]	–	Reserved	0x0

8.8.1.25 WAKEUP_MASK_COREBLK

- Base Address: 0x1002_0000
- Address = Base Address + 0x0628, Reset Value = 0x0010_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
USE_LEVEL_TRIGGER	[30]	RW	Uses level trigger mode for wakeup request signal. By default, PMU uses edge-trigger.	0x0
RSVD	[29:21]	–	Reserved	0x0
C2C	[20]	RW	Wake-up mask for C2C_WAKEUP The PMU disables C2C_WAKEUP in default setting.	0x1
RSVD	[19:0]	–	Reserved	0x0

8.8.1.26 HDMI_PHY_CONTROL

- Base Address: 0x1002_0000
- Address = Base Address + 0x0700, Reset Value = 0x0096_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	0x0
DIV_RATIO	[25:16]	RW	Clock divider ratio for HDMI	0x96
RSVD	[15:1]	–	Reserved	0x0
ENABLE	[0]	RW	HDMI PHY output isolation enables control 0 = Enables isolation 1 = Bypasses isolation By default, isolation is enabled because power supply for the corresponding hard macro may not be present.	0x0

8.8.1.27 USB_PHY_CONTROL

- Base Address: 0x1002_0000
- Address = Base Address + 0x0704, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
ENABLE	[0]	RW	USB PHY output isolation control 0 = Enables isolation 1 = Bypasses isolation By default, isolation is enabled because power supply for the corresponding hard macro may not be present.	0x0

8.8.1.28 HSIC_1_PHY_CONTROL

- Base Address: 0x1002_0000
- Address = Base Address + 0x0708, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
ENABLE	[0]	RW	HSIC_1 PHY output isolation control 0 = Enables isolation 1 = Bypasses isolation By default, isolation is enabled because power supply for the corresponding hard macro may not be present.	0x0

8.8.1.29 HSIC_2_PHY_CONTROL

- Base Address: 0x1002_0000
- Address = Base Address + 0x070C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
ENABLE	[0]	RW	HSIC_2 PHY output isolation control 0 = Enables isolation 1 = Bypasses isolation By default, isolation is enabled because power supply for the corresponding hard macro may not be present.	0x0

8.8.1.30 MIPI_PHY0_CONTROL

- Base Address: 0x1002_0000
- Address = Base Address + 0x0710, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
M_RESETN	[2]	RW	Resets DSIM part of MIPI_PHY0. This bit should be HIGH before enabling DSIM.	0x0
S_RESETN	[1]	RW	Resets CSIS part of MIPI_PHY0. This bit should be HIGH before enabling CSIS.	0x0
ENABLE	[0]	RW	MIPI_PHY0 enables selection. Set this bit to 1 at the system initialization step before data access from/to MIPI_PHY0 begins. Caution: If you have not used MIPI_PHY0 in your system, do not touch this bit. 0 = Disables 1 = Enables	0x0

8.8.1.31 MIPI_PHY1_CONTROL

- Base Address: 0x1002_0000
- Address = Base Address + 0x0714, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
M_RESETN	[2]	RW	Resets DSIM part of MIPI_PHY1. This bit should be HIGH before enabling DSIM.	0x0
S_RESETN	[1]	RW	Resets CSIS part of MIPI_PHY1. This bit should be HIGH before enabling CSIS.	0x0
ENABLE	[0]	RW	MIPI_PHY1 enables selection. Set this bit to 1 at the system initialization step before data access from/to MIPI_PHY1 begins. Caution: If you have not used MIPI_PHY1 in your system, do not touch this bit. 0 = Disables 1 = Enables	0x0

8.8.1.32 ADC_PHY_CONTROL

- Base Address: 0x1002_0000
- Address = Base Address + 0x0718, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
ENABLE	[0]	RW	TS-ADC enables control 0 = Disables 1 = Enables	0x1

8.8.1.33 BODY_BIAS_CON0

- Base Address: 0x1002_0000
- Address = Base Address + 0x0780, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SEL	[31]	RW	Selection for INT block body bias generator control signal 0 = Using internal eFuse 1 = Using BODY_BIAS_CON0	0x0
RSVD	[30:8]	-	Reserved	0x0
ENABLE_PMOS	[7]	RW	Enable INT block PMOS ABB control 0 = Bypass. Body bias voltage is the voltage of VDD_INT. 1 = Enable PMOS ABB control. Body bias voltage is selected by CODE_PMOS.	0x0
RSVD	[6:5]	-	Reserved.	0x0
CODE_PMOS	[4:0]	RW	INT block ABB control value for PMOS ABB control 00000 = 0.60 * voltage of VDD_INT 00001 = 0.65 * voltage of VDD_INT 00010 = 0.70 * voltage of VDD_INT 00011 = 0.75 * voltage of VDD_INT 00100 = 0.80 * voltage of VDD_INT 00101 = 0.85 * voltage of VDD_INT 00110 = 0.90 * voltage of VDD_INT 00111 = 0.95 * voltage of VDD_INT 01000 = 1.00 * voltage of VDD_INT 01001 = 1.05 * voltage of VDD_INT 01010 = 1.10 * voltage of VDD_INT 01011 = 1.15 * voltage of VDD_INT 01100 = 1.20 * voltage of VDD_INT 01101 = 1.25 * voltage of VDD_INT 01110 = 1.30 * voltage of VDD_INT 01111 = 1.35 * voltage of VDD_INT 10000 = 1.40 * voltage of VDD_INT 10001 = 1.45 * voltage of VDD_INT 10010 = 1.50 * voltage of VDD_INT 10011 = 1.55 * voltage of VDD_INT 10100 = 1.60 * voltage of VDD_INT 10101 ~ 11111 = Not available The voltage of VDD_INT shall be lower than 1.30V	0x0

8.8.1.34 BODY_BIAS_CON1

- Base Address: 0x1002_0000
- Address = Base Address + 0x0784, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SEL	[31]	RW	Selection for MIF block body bias generator control signal 0 = Using internal eFuse 1 = Using BODY_BIAS_CON0	0x0
RSVD	[30:8]	-	Reserved	0x0
ENABLE_PMOS	[7]	RW	Enable MIF block PMOS ABB control 0 = Bypass. Body bias voltage is the voltage of VDD_MIF. 1 = Enable PMOS ABB control. Body bias voltage is selected by CODE_PMOS.	0x0
RSVD	[6:5]	-	Reserved.	0x0
CODE_PMOS	[4:0]	RW	MIF block ABB control value for PMOS ABB control 00000 = 0.60 * voltage of VDD_MIF 00001 = 0.65 * voltage of VDD_MIF 00010 = 0.70 * voltage of VDD_MIF 00011 = 0.75 * voltage of VDD_MIF 00100 = 0.80 * voltage of VDD_MIF 00101 = 0.85 * voltage of VDD_MIF 00110 = 0.90 * voltage of VDD_MIF 00111 = 0.95 * voltage of VDD_MIF 01000 = 1.00 * voltage of VDD_MIF 01001 = 1.05 * voltage of VDD_MIF 01010 = 1.10 * voltage of VDD_MIF 01011 = 1.15 * voltage of VDD_MIF 01100 = 1.20 * voltage of VDD_MIF 01101 = 1.25 * voltage of VDD_MIF 01110 = 1.30 * voltage of VDD_MIF 01111 = 1.35 * voltage of VDD_MIF 10000 = 1.40 * voltage of VDD_MIF 10001 = 1.45 * voltage of VDD_MIF 10010 = 1.50 * voltage of VDD_MIF 10011 = 1.55 * voltage of VDD_MIF 10100 = 1.60 * voltage of VDD_MIF 10101 ~ 11111 = Not available The voltage of VDD_MIF shall be lower than 1.30V	0x0

8.8.1.35 BODY_BIAS_CON2

- Base Address: 0x1002_0000
- Address = Base Address + 0x0788, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SEL	[31]	RW	Selection for G3D block body bias generator control signal 0 = Using internal eFuse 1 = Using BODY_BIAS_CON0	0x0
RSVD	[30:8]	-	Reserved	0x0
ENABLE_PMOS	[7]	RW	Enable G3D block PMOS ABB control 0 = Bypass. Body bias voltage is the voltage of VDD_G3D. 1 = Enable PMOS ABB control. Body bias voltage is selected by CODE_PMOS.	0x0
RSVD	[6:5]	-	Reserved.	0x0
CODE_PMOS	[4:0]	RW	G3D block ABB control value for PMOS ABB control 00000 = 0.60 * voltage of VDD_G3D 00001 = 0.65 * voltage of VDD_G3D 00010 = 0.70 * voltage of VDD_G3D 00011 = 0.75 * voltage of VDD_G3D 00100 = 0.80 * voltage of VDD_G3D 00101 = 0.85 * voltage of VDD_G3D 00110 = 0.90 * voltage of VDD_G3D 00111 = 0.95 * voltage of VDD_G3D 01000 = 1.00 * voltage of VDD_G3D 01001 = 1.05 * voltage of VDD_G3D 01010 = 1.10 * voltage of VDD_G3D 01011 = 1.15 * voltage of VDD_G3D 01100 = 1.20 * voltage of VDD_G3D 01101 = 1.25 * voltage of VDD_G3D 01110 = 1.30 * voltage of VDD_G3D 01111 = 1.35 * voltage of VDD_G3D 10000 = 1.40 * voltage of VDD_G3D 10001 = 1.45 * voltage of VDD_G3D 10010 = 1.50 * voltage of VDD_G3D 10011 = 1.55 * voltage of VDD_G3D 10100 = 1.60 * voltage of VDD_G3D 10101 ~ 11111 = Not available The voltage of VDD_G3D shall be lower than 1.30V	0x0

8.8.1.36 BODY_BIAS_CON3

- Base Address: 0x1002_0000
- Address = Base Address + 0x078C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SEL	[31]	RW	Selection for ARM block body bias generator control signal 0 = Using internal eFuse 1 = Using BODY_BIAS_CON0	0x0
RSVD	[30:8]	-	Reserved	0x0
ENABLE_PMOS	[7]	RW	Enable ARM block PMOS ABB control 0 = Bypass. Body bias voltage is the voltage of VDD_ARM. 1 = Enable PMOS ABB control. Body bias voltage is selected by CODE_PMOS.	0x0
RSVD	[6:5]	-	Reserved.	0x0
CODE_PMOS	[4:0]	RW	ARM block ABB control value for PMOS ABB control 00000 = 0.60 * voltage of VDD_ARM 00001 = 0.65 * voltage of VDD_ARM 00010 = 0.70 * voltage of VDD_ARM 00011 = 0.75 * voltage of VDD_ARM 00100 = 0.80 * voltage of VDD_ARM 00101 = 0.85 * voltage of VDD_ARM 00110 = 0.90 * voltage of VDD_ARM 00111 = 0.95 * voltage of VDD_ARM 01000 = 1.00 * voltage of VDD_ARM 01001 = 1.05 * voltage of VDD_ARM 01010 = 1.10 * voltage of VDD_ARM 01011 = 1.15 * voltage of VDD_ARM 01100 = 1.20 * voltage of VDD_ARM 01101 = 1.25 * voltage of VDD_ARM 01110 = 1.30 * voltage of VDD_ARM 01111 = 1.35 * voltage of VDD_ARM 10000 = 1.40 * voltage of VDD_ARM 10001 = 1.45 * voltage of VDD_ARM 10010 = 1.50 * voltage of VDD_ARM 10011 = 1.55 * voltage of VDD_ARM 10100 = 1.60 * voltage of VDD_ARM 10101 ~ 11111 = Not available The voltage of VDD_ARM shall be lower than 1.30V	0x0

8.8.1.37 INFORM0

- Base Address: 0x1002_0000
- Address = Base Address + 0x0800, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
INFORM	[31:0]	RW	User-defined information register. By asserting XnRESET pin, PMU clears INFORM0 to 3 registers.	0x0

8.8.1.38 INFORM1

- Base Address: 0x1002_0000
- Address = Base Address + 0x0804, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
INFORM	[31:0]	RW	User-defined information register. By asserting XnRESET pin, PMU clears INFORM0 to 3 registers.	0x0

8.8.1.39 INFORM2

- Base Address: 0x1002_0000
- Address = Base Address + 0x0808, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
INFORM	[31:0]	RW	User-defined information register. By asserting XnRESET pin, PMU clears INFORM0 to 3 registers.	0x0

8.8.1.40 INFORM3

- Base Address: 0x1002_0000
- Address = Base Address + 0x080C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
INFORM	[31:0]	RW	User-defined information register. By asserting XnRESET pin, PMU clears INFORM0 to 3 registers.	0x0

8.8.1.41 INFORM4

- Base Address: 0x1002_0000
- Address = Base Address + 0x0810, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
INFORM	[31:0]	RW	User-defined information register. Only power-up reset clears INFORM4 to 7.	0x0

8.8.1.42 INFORM5

- Base Address: 0x1002_0000
- Address = Base Address + 0x0814, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
INFORM	[31:0]	RW	User-defined information register. Only power-up reset clears INFORM4 to 7.	0x0

8.8.1.43 INFORM6

- Base Address: 0x1002_0000
- Address = Base Address + 0x0818, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
INFORM	[31:0]	RW	User-defined information register. Only power-up reset clears INFORM4 to 7.	0x0

8.8.1.44 INFORM7

- Base Address: 0x1002_0000
- Address = Base Address + 0x081C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
INFORM	[31:0]	RW	User-defined information register. Only power-up reset clears INFORM4 to 7.	0x0

8.8.1.45 IROM_DATA_REG0

- Base Address: 0x1002_0000
- Address = Base Address + 0x0980, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
IROM_DATA	[31:0]	RW	Data field for IROM code	0x0

8.8.1.46 IROM_DATA_REG1

- Base Address: 0x1002_0000
- Address = Base Address + 0x0984, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
IROM_DATA	[31:0]	RW	Data field for IROM code	0x0

8.8.1.47 IROM_DATA_REG2

- Base Address: 0x1002_0000
- Address = Base Address + 0x0988, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
IROM_DATA	[31:0]	RW	Data field for IROM code	0x0

8.8.1.48 IROM_DATA_REG3

- Base Address: 0x1002_0000
- Address = Base Address + 0x098C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
IROM_DATA	[31:0]	RW	Data field for IROM code	0x0

8.8.1.49 PMU_DEBUG

- Base Address: 0x1002_0000
- Address = Base Address + 0x0A00, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ENABLE_SERIAL_DEBUG	[31]	RW	When you enable ENABLE_SERIAL_DEBUG, PMU generates serial debug output data to external interrupt port.	0x0
SELECT_POWER_DOWN_STATE_OUTPUT_PORT	[30:26]	RW	Decides the debug port that is used for giving power-down state information.	0x0
RSVD	[25]	–	Reserved	0x0
SELECT_SERIAL_DEBUG_OUTPUT_PORT	[24:20]	RW	Decides the debug port for serial debug output data. This field is effective only when you turn on ENABLE_SERIAL_DEBUG.	0x0
RSVD	[19]	–	Reserved	0x0
DBG_SEL	[18:16]	RW	Debugger signals set selection Debugging information is available to XEINT port that is configured in GPIO_ALIVE.	0x0
RSVD	[15:12]	–	Reserved	0x0
CLKOUT_SEL	[11:8]	RW	Controls clock out 0000 = Clock output signal from CMU_DMC 0001 = Clock output signal from CMU_TOP 0010 = Clock output signal from CMU_LEFTBUS 0011 = Clock output signal from CMU_RIGHTBUS 0100 = Clock output signal from CMU_CPU 0101 = Clock output signal from CMU_ISP 1000 = XXTI 1001 = XUSBXTI 1100 = RTC TICCLK 1101 = RTCCLK 1110 = CLKOUT__DEBUG	0x0
RSVD	[7:1]	–	Reserved	0x0
CLKOUT_DISABLE	[0]	RW	When CLKOUT_DISABLE is set to HIGH, PMU disables CLKOUT.	0x0

8.8.1.50 ARM_CORE0_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1000, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x2 = Reset 0x0 = Power down	0x3

8.8.1.51 DIS_IRQ_ARM_CORE0_LOCAL_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1004, Reset Value = 0xFFFF_FFFE

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down By default, field [0] is set to LOW. It sets this register to HIGH only for debugging purpose.	0x0

8.8.1.52 DIS_IRQ_ARM_CORE0_CENTRAL_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1008, Reset Value = 0xFFFF_FFFE

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power down level HIGH: Power on LOW: Power down By default, field [0] is set to LOW. It sets this register to HIGH only for debugging purpose.	0x0

8.8.1.53 ARM_CORE1_SYS_PWR_REG

- Base Address = 0x1002_0000 Base Address: 0x1002_0000
- Address = Base Address + 0x1010, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x2 = Reset 0x0 = Power down	0x3

8.8.1.54 DIS_IRQ_ARM_CORE1_LOCAL_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1014, Reset Value = 0xFFFF_FFFE

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down By default, field[0] is set to LOW. It sets this register to HIGH only for debugging purpose.	0x0

8.8.1.55 DIS_IRQ_ARM_CORE1_CENTRAL_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1018, Reset Value = 0xFFFF_FFFE

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down By default, field[0] is set to LOW. It sets this register to HIGH only for debugging purpose.	0x0

8.8.1.56 ARM_CORE2_SYS_PWR_REG

- Base Address = 0x1002_0000 Base Address: 0x1002_0000
- Address = Base Address + 0x1020, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x2 = Reset 0x0 = Power down	0x3

8.8.1.57 DIS_IRQ_ARM_CORE2_LOCAL_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1024, Reset Value = 0xFFFF_FFFE

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down By default, field[0] is set to LOW. It sets this register to HIGH only for debugging purpose.	0x0

8.8.1.58 DIS_IRQ_ARM_CORE2_CENTRAL_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1028, Reset Value = 0xFFFF_FFFE

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down By default, field[0] is set to LOW. It sets this register to HIGH only for debugging purpose.	0x0

8.8.1.59 ARM_CORE3_SYS_PWR_REG

- Base Address = 0x1002_0000 Base Address: 0x1002_0000
- Address = Base Address + 0x1030, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x2 = Reset 0x0 = Power down	0x3

8.8.1.60 DIS_IRQ_ARM_CORE3_LOCAL_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1034, Reset Value = 0xFFFF_FFFE

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down By default, field[0] is set to LOW. It sets this register to HIGH only for debugging purpose.	0x0

8.8.1.61 DIS_IRQ_ARM_CORE3_CENTRAL_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1038, Reset Value = 0xFFFF_FFFE

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down By default, field[0] is set to LOW. It sets this register to HIGH only for debugging purpose.	0x0

8.8.1.62 ISP_ARM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1050, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level low-power mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.63 DIS_IRQ_ISP_ARM_LOCAL_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1054, Reset Value = 0xFFFF_FFFE

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level low-power mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down By default, field[0] is set to LOW. It sets this register to HIGH only for debugging purpose.	0x0

8.8.1.64 DIS_IRQ_ISP_ARM_CENTRAL_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1058, Reset Value = 0xFFFF_FFFE

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level low-power mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down By default, field[0] is set to LOW. It sets this register to HIGH only for debugging purpose.	0x0

8.8.1.65 ARM_COMMON_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1080, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x2 = Reset 0x0 = Power down	0x3

8.8.1.66 ARM_L2_0_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x10C0, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x2 = Retention 0x0 = Power off	0x3

8.8.1.67 ARM_L2_1_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x10C4, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x2 = Retention 0x0 = Power off	0x3

8.8.1.68 CMU_ACLKSTOP_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1100, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.69 CMU_SCLKSTOP_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1104, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.70 CMU_RESET_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x110C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.71 CMU_ACLKSTOP_COREBLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1110, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	-	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.72 CMU_SCLKSTOP_COREBLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1114, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.73 CMU_RESET_COREBLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x111C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.74 APLL_SYSCLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1120, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.75 MPLL_SYSCLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1124, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.76 VPLL_SYSCLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1128, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.77 EPLL_SYSCLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x112C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.78 MPLLUSER_SYSCLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1130, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.79 CMU_CLKSTOP_GPS_ALIVE_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1138, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.80 CMU_RESET_GPS_ALIVE_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x113C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.81 CMU_CLKSTOP_CAM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1140, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.82 CMU_CLKSTOP_TV_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1144, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.83 CMU_CLKSTOP_MFC_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1148, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.84 CMU_CLKSTOP_G3D_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x114C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.85 CMU_CLKSTOP_LCD0_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1150, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.86 CMU_CLKSTOP_ISP_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1154, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.87 CMU_CLKSTOP_MAUDIO_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1158, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.88 CMU_CLKSTOP_GPS_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x115C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.89 CMU_RESET_CAM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1160, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.90 CMU_RESET_TV_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1164, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.91 CMU_RESET_MFC_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1168, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.92 CMU_RESET_G3D_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x116C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.93 CMU_RESET_LCD0_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1170, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.94 CMU_RESET_ISP_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1174, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.95 CMU_RESET_MAUDIO_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1178, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.96 CMU_RESET_GPS_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x117C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.97 TOP_BUS_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1180, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.98 TOP_RETENTION_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1184, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.99 TOP_PWR_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1188, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.100 TOP_BUS_COREBLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1190, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x3

8.8.1.101 TOP_RETENTION_COREBLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1194, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.102 TOP_PWR_COREBLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1198, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x3

8.8.1.103 LOGIC_RESET_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x11A0, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.104 OSCCLK_GATE_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x11A4, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.105 LOGIC_RESET_COREBLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x11B0, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.106 OSCCLK_GATE_COREBLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x11B4, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.107 OneNANDXL_MEM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x11C0, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.108 HSI_MEM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x11C4, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x3

8.8.1.109 G2D_ACP_MEM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x11C8, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.110 USBOG_MEM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x11CC, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.111 SDMMC_MEM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x11D0, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.112 CSSYS_MEM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x11D4, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.113 SECSS_MEM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x11D8, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in system-level power-down mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.114 ROTATOR_MEM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x11DC, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x3FFFFFFF
SYS_PWR_CFG	[1:0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x3

8.8.1.115 PAD_RETENTION_DRAM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1200, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.116 PAD_RETENTION_MAUDIO_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1204, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.117 PAD_RETENTION_GPIO_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1220, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.118 PAD_RETENTION_UART_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1224, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.119 PAD_RETENTION_MMCA_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1228, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.120 PAD_RETENTION_MMCB_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x122C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.121 PAD_RETENTION_EBIA_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1230, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.122 PAD_RETENTION_EBIB_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1234, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.123 PAD_RETENTION_GPIO_COREBLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x123C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.124 PAD_ISOLATION_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1240, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.125 PAD_ISOLATION_COREBLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1250, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.126 PAD_ALV_SEL_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1260, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.127 XUSBXTI_SYS_PWR_REG

- Base Address: 0x1002_0000
- Base Address = Base Address + 0x1280, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Enables 0x0 = Disables	0x1

8.8.1.128 XXTI_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1284, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Enables 0x0 = Disables	0x1

8.8.1.129 EXT_REGULATOR_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x12C0, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.130 GPIO_MODE_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1300, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.131 GPIO_MODE_COREBLK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1320, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.132 GPIO_MODE_MAUDIO_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1340, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in system-level power-down mode 0x1 = Power on 0x0 = Power down	0x1

8.8.1.133 TOP_ASB_RESET_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1344, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.134 TOP_ASB_ISOLATION_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1348, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.135 CAM_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1380, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x1FFFFFFF
SYS_PWR_CFG	[2:0]	RW	Controls power state in system-level power-down mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.136 TV_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1384, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x1FFFFFFF
SYS_PWR_CFG	[2:0]	RW	Controls power state in system-level power-down mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.137 MFC_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1388, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x1FFFFFFF
SYS_PWR_CFG	[2:0]	RW	Controls power state in system-level power-down mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.138 G3D_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x138C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x1FFFFFFF
SYS_PWR_CFG	[2:0]	RW	Controls power state in system-level power-down mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.139 LCD0_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1390, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x1FFFFFFF
SYS_PWR_CFG	[2:0]	RW	Controls power state in system-level power-down mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.140 ISP_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1394, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x1FFFFFFF
SYS_PWR_CFG	[2:0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x7

8.8.1.141 MAUDIO_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x1398, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x1FFFFFFF
SYS_PWR_CFG	[2:0]	RW	Controls power state in system-level power-down mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.142 GPS_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x139C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x1FFFFFFF
SYS_PWR_CFG	[2:0]	RW	Controls power state in system-level power-down mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.143 GPS_ALIVE_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x13A0, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x1FFFFFFF
SYS_PWR_CFG	[2:0]	RW	Controls power state in system-level power-down mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.144 DRAM_FREQ_DOWN_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x13B0, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.145 DDRPHY_DLLOFF_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x13B4, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.146 CMU_SYSCLK_ISP_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x13B8, Reset Value = 0xFFFF_FFFE

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x0

8.8.1.147 CMU_SYSCLK_GPS_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x13BC, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.148 LPDDR_PHY_DLL_LOCK_SYS_PWR_REG

- Base Address: 0x1002_0000
- Address = Base Address + 0x13C0, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x7FFFFFFF
SYS_PWR_CFG	[0]	RW	Controls power state in LOWPWR mode Each bit represents power state in each power-down level HIGH: Power on LOW: Power down	0x1

8.8.1.149 ARM_CORE0_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2000, Reset Value = 0x0000_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
LOCAL_PWR_CFG	[1:0]	RWX	Controls power state in NORMAL mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.150 ARM_CORE0_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x2004, Reset Value = 0x0003_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0xC000
STATUS	[1:0]	R	Verifies power state in NORMAL mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.151 ARM_CORE0_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2008, Reset Value = 0x0101_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Do not modify this field.	0x00808
USE_DELAYED_ RESET_ ASSERTION	[12]	RWX	Decide whether the PMU delays assertion of reset of ARM_CORE until power is re-supplied or not. 0x1 = Delayed reset assertion 0x0 = Normal reset assertion	0
RSVD	[11:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RWX	Uses power control feedback to measure power on/off duration of ARM_CORE0. 0x1 = Enable 0x0 = Disable	0
USE_SC_ COUNTER	[0]	RWX	Uses counter to measure power on/off duration of ARM_CORE0. 0x1 = Enable 0x0 = Disable	1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.152 ARM_CORE1_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2080, Reset Value = 0x0000_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
LOCAL_PWR_CFG	[1:0]	RWX	Controls power state in NORMAL mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.153 ARM_CORE1_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x2084, Reset Value = 0x0003_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0xC000
STATUS	[1:0]	R	Verifies power state in NORMAL mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.154 ARM_CORE1_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2088, Reset Value = 0x0101_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Do not modify this field	0x00808
USE_DELAYED_ RESET_ ASSERTION	[12]	RWX	Decide whether the PMU delays assertion of reset of ARM_CORE until power is re-supplied or not. 0x1 = Delayed reset assertion 0x0 = Normal reset assertion	0
RSVD	[11:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RWX	Uses power control feedback to measure power on/off duration of ARM_CORE0. 0x1 = Enables 0x0 = Disables	0
USE_SC_ COUNTER	[0]	RWX	Uses counter to measure power on/off duration of ARM_CORE0. 0x1 = Enables 0x0 = Disables	1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.155 ARM_CORE2_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2100, Reset Value = 0x0000_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
LOCAL_PWR_CFG	[1:0]	RWX	Controls power state in NORMAL mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.156 ARM_CORE2_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x2104, Reset Value = 0x0003_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0xC000
STATUS	[1:0]	R	Verifies power state in NORMAL mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.157 ARM_CORE2_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2108, Reset Value = 0x0101_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Do not modify this field	0x00808
USE_DELAYED_ RESET_ ASSERTION	[12]	RWX	Decide whether the PMU delays assertion of reset of ARM_CORE until power is re-supplied or not. 0x1 = Delayed reset assertion 0x0 = Normal reset assertion	0
RSVD	[11:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RWX	Uses power control feedback to measure power on/off duration of ARM_CORE0. 0x1 = Enables 0x0 = Disables	0
USE_SC_ COUNTER	[0]	RWX	Uses counter to measure power on/off duration of ARM_CORE0. 0x1 = Enables 0x0 = Disables	1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.158 ARM_CORE3_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2180, Reset Value = 0x0000_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
LOCAL_PWR_CFG	[1:0]	RWX	Controls power state in NORMAL mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.159 ARM_CORE3_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x2184, Reset Value = 0x0003_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0xC000
STATUS	[1:0]	R	Verifies power state in NORMAL mode 0x3 = Power on 0x0 = Power down	0x3

8.8.1.160 ARM_CORE3_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2188, Reset Value = 0x0101_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Do not modify this field	0x00808
USE_DELAYED_ RESET_ ASSERTION	[12]	RWX	Decide whether the PMU delays assertion of reset of ARM_CORE until power is re-supplied or not. 0x1 = Delayed reset assertion 0x0 = Normal reset assertion	0
RSVD	[11:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RWX	Uses power control feedback to measure power on/off duration of ARM_CORE0. 0x1 = Enables 0x0 = Disables	0
USE_SC_ COUNTER	[0]	RWX	Uses counter to measure power on/off duration of ARM_CORE0. 0x1 = Enables 0x0 = Disables	1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.161 ISP_ARM_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2280, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
LOCAL_PWR_CFG	[0]	RW	Controls power state in NORMAL mode Each bit represents power state in each power-down level HIGH: Run LOW: Reset	0x1

8.8.1.162 ISP_ARM_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x2284, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:19]	–	Reserved	0x0
STATES	[18:16]	R	Verifies state machine status.	0x0
RSVD	[15:1]	–	Reserved	0x0
STATUS	[0]	R	Verifies power state in NORMAL mode Each bit represents power state in each power-down level HIGH: Run LOW: Reset	0x1

8.8.1.163 ISP_ARM_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2288, Reset Value = 0x0101_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
USE_ STANDBYWFE	[24]	RWX	Uses STANDBYWFE for judging whether ARM is ready for entering low-power mode. Either one of USE_STANDBYWFI or USE_STANDBYWFE should be activated at a time. Updates changes on USE_STANDBYWFE_ISP_ARM field of CENTRAL_SEQ_OPTION register automatically in this field.	0x1
RSVD	[23:17]	–	Reserved	0x0
USE_ STANDBYWFI	[16]	RWX	Uses STANDBYWFI for judging whether ARM is ready for entering low-power mode. You should activate either USE_STANDBYWFI or USE_STANDBYWFE at a time. Updates changes on USE_STANDBYWFI_ISP_ARM field of CENTRAL_SEQ_OPTION register automatically in this field.	0x1
ENABLE	[15]	RWX	By default, PMU disables ISP_ARM. Write 0x1 to this field to enable ISP_ARM.	0x0
RSVD	[14:0]	–	Reserved	0x0

8.8.1.164 ARM_COMMON_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2408, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Do not modify this field	0x0000000
USE_SC_ FEEDBACK	[1]	RWX	Uses power control feedback to measure power on/off duration of ARM_CORE0. 0x1 = Enable 0x0 = Disable	0
USE_SC_ COUNTER	[0]	RWX	Uses counter to measure power on/off duration of ARM_CORE0. 0x1 = Enable 0x0 = Disable	1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.165 ARM_L2_0_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2600, Reset Value = 0x0000_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
LOCAL_PWR_CFG	[1:0]	RW	Controls power state in NORMAL mode Each bit represents power state in each power-down level 0x3 = Power on 0x0 = Power off when USE_RETENTION is "0", Power retention when USE_RETENTION is "1".	0x3

8.8.1.166 ARM_L2_0_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x2604, Reset Value = 0x0000_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
STATUS	[1:0]	R	Verifies power state in NORMAL mode Each bit represents power state in each power-down level. 0x3 = Power on 0x0 = Power off when USE_RETENTION is "0", Power retention when USE_RETENTION is "1".	0x3

8.8.1.167 ARM_L2_0_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2608, Reset Value = 0x0000_0010

Name	Bit	Type	Description	Reset Value
RSVD	[31:21]	–	Reserved	0x0
IGNORE_OUTPUT_UPDATE_DONE	[20]	RW	By default, some state waits until Reset/CLAMPOUT is properly updated to loads. This field disables this waiting function.	0x0
RSVD	[19:5]	–	Reserved	0x0
USE_RETENTION	[4]	RW	Decides whether to use retention capability.	0x1
RSVD	[3:0]	–	Reserved	0x0

8.8.1.168 ARM_L2_1_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2620, Reset Value = 0x0000_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
LOCAL_PWR_CFG	[1:0]	RW	Controls power state in NORMAL mode Each bit represents power state in each power-down level 0x3 = Power on 0x0 = Power off when USE_RETENTION is "0", Power retention when USE_RETENTION is "1".	0x3

8.8.1.169 ARM_L2_1_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x2624, Reset Value = 0x0000_0003

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
STATUS	[1:0]	R	Verifies power state in NORMAL mode Each bit represents power state in each power-down level 0x3 = Power on 0x0 = Power off when USE_RETENTION is "0", Power retention when USE_RETENTION is "1".	0x3

8.8.1.170 ARM_L2_1_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2628, Reset Value = 0x0000_0010

Name	Bit	Type	Description	Reset Value
RSVD	[31:21]	–	Reserved	0x0
IGNORE_OUTPUT _UPDATE_DONE	[20]	RW	By default, some state waits until Reset/CLAMPOUT is properly updated to loads. This field disables this waiting function.	0x0
RSVD	[19:5]	–	Reserved	0x0
USE_RETENTION	[4]	RW	Decides whether to use retention capability.	0x1
RSVD	[3:0]	–	Reserved	0x0

8.8.1.171 DRAM_FREQ_DOWN_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x29A8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0x0
AUTOMATIC_ WAKEUP	[29]	RW	0 = S/W restores DRAM speed after finishing the power-up by writing WAKEUP_FROM_LOWPWR register bit to "1". 1= H/W restores DRAM speed automatically during wakeup sequence.	0x0
WAKEUP_FROM_ LOWPWR	[28]	RWX	Restores DRAM speed after finishing the power-up.	0x0
RSVD	[27:0]	–	Reserved	0x0

8.8.1.172 DDRPHY_DLLOFF_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2DC8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0x0
AUTOMATIC_ WAKEUP	[29]	RW	0 = S/W re-enables DDRPHY DLL after finishing power-up by writing WAKEUP_FROM_LOWPWR register bit to "1". 1 = H/W re-enables DDRPHY DLL automatically during wakeup sequence.	0x0
WAKEUP_FROM_ LOWPWR	[28]	RWX	Re-enables DDRPHY DLL after power-down.	0x0
RSVD	[27:0]	–	Reserved	0x0

8.8.1.173 OneNANDXL_MEM_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2E08, Reset Value = 0x0000_0010

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:5]	–	Reserved	0x0
USE_RETENTION	[4]	RW	Decides whether to use retention capability.	0x1
RSVD	[3:0]	–	Reserved	0x0

8.8.1.174 HSI_MEM_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2E28, Reset Value = 0x0000_0010

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:5]	–	Reserved	0x0
USE_RETENTION	[4]	RW	Decides whether to use retention capability.	0x1
RSVD	[3:0]	–	Reserved	0x0

8.8.1.175 G2D_ACP_MEM_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2E48, Reset Value = 0x0000_0010

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:5]	–	Reserved	0x0
USE_RETENTION	[4]	RW	Decides whether to use retention capability.	0x1
RSVD	[3:0]	–	Reserved	0x0

8.8.1.176 USBOTG_MEM_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2E68, Reset Value = 0x0000_0010

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:5]	–	Reserved	0x0
USE_RETENTION	[4]	RW	Decides whether to use retention capability.	0x1
RSVD	[3:0]	–	Reserved	0x0

8.8.1.177 SDMMC_MEM_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2E88, Reset Value = 0x0000_0010

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:5]	–	Reserved	0x0
USE_RETENTION	[4]	RW	Decides whether to use retention capability.	0x1
RSVD	[3:0]	–	Reserved	0x0

8.8.1.178 CSSYS_MEM_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2EA8, Reset Value = 0x0000_0010

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:5]	–	Reserved	0x0
USE_RETENTION	[4]	RW	Decides whether to use retention capability.	0x1
RSVD	[3:0]	–	Reserved	0x0

8.8.1.179 SECSS_MEM_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2EC8, Reset Value = 0x0000_0010

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:5]	–	Reserved	0x0
USE_RETENTION	[4]	RW	Decides whether to use retention capability.	0x1
RSVD	[3:0]	–	Reserved	0x0

8.8.1.180 ROTATOR_MEM_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x2F48, Reset Value = 0x0000_0010

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:5]	–	Reserved	0x0
USE_RETENTION	[4]	RW	Decides whether to use retention capability.	0x1
RSVD	[3:0]	–	Reserved	0x0

8.8.1.181 PAD_RETENTION_MAUDIO_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3028, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
WAKEUP_FROM_ LOWPWR	[28]	RWX	In activate-related, PAD enters retention mode from retention state in case of system-level low-power mode. In case-related, LOCAL_PWR_CFG puts PAD in retention state. Writing to this register can cause undefined behavior.	0x0
RSVD	[27:0]	–	Reserved	0x0

8.8.1.182 PAD_RETENTION_GPIO_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3108, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
WAKEUP_FROM_ LOWPWR	[28]	RWX	In activate-related, PAD enters retention mode from retention state in case of system-level low-power mode. In case-related, LOCAL_PWR_CFG puts PAD in retention state. Writing to this register can cause undefined behavior.	0x0
RSVD	[27:0]	–	Reserved	0x0

8.8.1.183 PAD_RETENTION_UART_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3128, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
WAKEUP_FROM_ LOWPWR	[28]	RWX	In activate-related, PAD enters retention mode from retention state in case of system-level low-power mode. In case-related, LOCAL_PWR_CFG puts PAD in retention state. Writing to this register can cause undefined behavior.	0x0
RSVD	[27:0]	–	Reserved	0x0

8.8.1.184 PAD_RETENTION_MMCA_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3148, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
WAKEUP_FROM_ LOWPWR	[28]	RWX	In activate-related, PAD enters retention mode from retention state in case of system-level low-power mode. In case-related, LOCAL_PWR_CFG puts PAD in retention state. Writing to this register can cause undefined behavior.	0x0
RSVD	[27:0]	–	Reserved	0x0

8.8.1.185 PAD_RETENTION_MMCB_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3168, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
WAKEUP_FROM_ LOWPWR	[28]	RWX	In activate-related, PAD enters retention mode from retention state in case of system-level low-power mode. In case-related, LOCAL_PWR_CFG puts PAD in retention state. Writing to this register can cause undefined behavior.	0x0
RSVD	[27:0]	–	Reserved	0x0

8.8.1.186 PAD_RETENTION_EBIA_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3188, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
WAKEUP_FROM_ LOWPWR	[28]	RWX	In activate-related, PAD enters retention mode from retention state in case of system-level low-power mode. In case-related, LOCAL_PWR_CFG puts PAD in retention state. Writing to this register can cause undefined behavior.	0x0
RSVD	[27:0]	–	Reserved	0x0

8.8.1.187 PAD_RETENTION_EBIB_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x31A8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
WAKEUP_FROM_ LOWPWR	[28]	RWX	In activate-related, PAD enters retention mode from retention state in case of system-level low-power mode. In case-related, LOCAL_PWR_CFG puts PAD in retention state. Writing to this register can cause undefined behavior.	0x0
RSVD	[27:0]	–	Reserved	0x0

8.8.1.188 PAD_RETENTION_GPIO_COREBLK_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x31E8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0x0
WAKEUP_FROM_ LOWPWR	[28]	RWX	In activate-related, PAD enters retention mode from retention state in case of system-level low-power mode. In case-related, LOCAL_PWR_CFG puts PAD in retention state. Writing to this register can cause undefined behavior.	0x0
RSVD	[27:0]	–	Reserved	0x0

8.8.1.189 PS_HOLD_CONTROL

- Base Address: 0x1002_0000
- Address = Base Address + 0x330C, Reset Value = 0x0000_5200

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	0x14
EN	[9]	RW	Enables signal for PSHOLD port. 0 = Input 1 = Output You should reset PSHOLD control only when cold booted.	0x1
DATA	[8]	RW	PAD driving value You should reset PSHOLD control only when cold booted.	0x0
RSVD	[7:0]	–	Reserved	0x0

8.8.1.190 XUSBXTI_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3400, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
LOCAL_PWR_CFG	[0]	RW	Controls XUSBXTI pad in NORMAL mode 0x1 = Enable 0x0 = Disable	0x1

8.8.1.191 XUSBXTI_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x3404, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
STATUS	[0]	R	Verifies the status of XUSBXTI pad in NORMAL mode 0x1 = Enables 0x0 = Disables	0x1

8.8.1.192 XUSBXTI_DURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x341C, Reset Value = 0xFFF0_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	0xFFF
DUR_STABLE	[19:0]	RW	Sets the time required to stabilize XUSBXTI. DUR_STABLE appends 0xF. You should compare this value to counter value. NOTE: Stabilization time = DUR_STABLE × 16 + 15 (0xF)	0x0

8.8.1.193 XXTI_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3420, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
LOCAL_PWR_CFG	[0]	RW	Controls XXTI pad in NORMAL mode 0x1 = Enables 0x0 = Disables	0x1

8.8.1.194 XXTI_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x3424, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
STATUS	[0]	R	Verifies the status of XXTI pad in NORMAL mode 0x1 = Enables 0x0 = Disables	0x1

8.8.1.195 XXTI_DURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x343C, Reset Value = 0xFFF0_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	0xFFF
DUR_STABLE	[19:0]	RW	Sets the time required to stabilize XXTI. DUR_STABLE appends 0xF. You should compare this value to counter value. NOTE: Stabilization time = DUR_STABLE × 16 + 15 (0xF)	0x0

8.8.1.196 EXT_REGULATOR_DURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x361C, Reset Value = 0xFFF0_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	0xFFF
DUR_STABLE	[19:0]	RW	Sets the time required to stabilize EXT_REGULATOR. DUR_STABLE appends 0xF. You should compare this value to counter value. NOTE: Stabilization time = DUR_STABLE × 16 + 15 (0xF)	0x3FFF

8.8.1.197 CAM_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C00, Reset Value = 0x0000_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
LOCAL_PWR_CFG	[2:0]	RW	Controls power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.198 CAM_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C04, Reset Value = 0x0006_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0xC000
STATUS	[2:0]	R	Verifies power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.199 CAM_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C08, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RW	Uses power control feedback to measure power on/off duration of CAM. You should either activate USE_SC_COUNTER or USE_SC_FEEDBACK.	0x0
USE_SC_ COUNTER	[0]	RW	Uses counter to measure power on/off duration of CAM. You should either activate USE_SC_COUNTER or USE_SC_FEEDBACK.	0x1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.200 TV_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C20, Reset Value = 0x0000_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
LOCAL_PWR_CFG	[2:0]	RW	Controls power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.201 TV_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C24, Reset Value = 0x0006_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0xC000
STATUS	[2:0]	R	Verifies power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.202 TV_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C28, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RW	Uses power control feedback to measure power on/off duration of TV. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x0
USE_SC_ COUNTER	[0]	RW	Uses counter to measure power on/off duration of TV. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.203 MFC_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C40, Reset Value = 0x0000_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
LOCAL_PWR_CFG	[2:0]	RW	Controls power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.204 MFC_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C44, Reset Value = 0x0006_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0xC000
STATUS	[2:0]	R	Verifies power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.205 MFC_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C48, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RW	Uses power control feedback to measure power on/off duration of MFC. You should either activate USE_SC_COUNTER or USE_SC_FEEDBACK.	0x0
USE_SC_ COUNTER	[0]	RW	Uses counter to measure power on/off duration of MFC. You should either activate USE_SC_COUNTER or USE_SC_FEEDBACK.	0x1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.206 G3D_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C60, Reset Value = 0x0000_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
LOCAL_PWR_CFG	[2:0]	RW	Controls power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.207 G3D_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C64, Reset Value = 0x0006_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0xC000
STATUS	[2:0]	R	Verifies power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.208 G3D_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C68, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RW	Uses power control feedback to measure power on/off duration of G3D. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x0
USE_SC_ COUNTER	[0]	RW	Uses counter to measure power on/off duration of G3D. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.209 LCD0_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C80, Reset Value = 0x0000_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
LOCAL_PWR_CFG	[2:0]	RW	Controls power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.210 LCD0_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C84, Reset Value = 0x0006_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0xC000
STATUS	[2:0]	R	Verifies power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.211 LCD0_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3C88, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RW	Uses power control feedback to measure power on/off duration of LCD0. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x0
USE_SC_ COUNTER	[0]	RW	Uses counter to measure power on/off duration of LCD0. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.212 ISP_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3CA0, Reset Value = 0x0000_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
LOCAL_PWR_CFG	[2:0]	RW	Controls power state in NORMAL mode Each bit represents power state in each power-down level. HIGH: Power on LOW: Power down	0x7

8.8.1.213 ISP_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x3CA4, Reset Value = 0x0000_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:22]	–	Reserved	0x0
STATES	[21:16]	R	Verifies state machine status.	0x0
RSVD	[15:3]	–	Reserved	0x0
STATUS	[2:0]	R	Verifies power state in NORMAL mode Each bit represents power state in each power-down level. HIGH: Power on LOW: Power down	0x7

8.8.1.214 ISP_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3CA8, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RW	Uses power control feedback to measure power on/off duration of ISP. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x0
USE_SC_ COUNTER	[0]	RW	Uses counter to measure power on/off duration of ISP. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.215 ISP_DURATION0

- Base Address: 0x1002_0000
- Address = Base Address + 0x3CB0, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0xFF
DUR_WAIT_ RESET	[23:20]	RW	Sets minimum amount of time to assert reset.	0xF
DUR_CHG_RESET	[19:16]	RW	Sets required time after PMU asserts/de-asserts reset before system starts to respond to it.	0xF
RSVD	[15:12]	–	Reserved	0xF
DUR_SCPRE	[11:8]	RW	Sets SCPRE on/off duration for ISP	0xF
DUR_SCALL	[7:4]	RW	Sets SCALL on/off duration for ISP	0xF
RSVD	[3:0]	–	Reserved	0xF

8.8.1.216 ISP_DURATION2

- Base Address: 0x1002_0000
- Address = Base Address + 0x3CB8, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0xFFFFFFF
DUR_ISO	[3:0]	RW	Sets required time after PMU asserts/de-asserts isolation control before system starts to respond to it.	0xF

8.8.1.217 MAUDIO_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3CC0, Reset Value = 0x0000_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
LOCAL_PWR_CFG	[2:0]	RW	Controls power state in NORMAL mode Each bit represents power state in each power-down level. HIGH: Power on LOW: Power down	0x7

8.8.1.218 MAUDIO_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x3CC4, Reset Value = 0x0006_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:22]	–	Reserved	0x0
STATES	[21:16]	R	Verifies state machine status.	0x6
RSVD	[15:3]	–	Reserved	0x0
STATUS	[2:0]	R	Verifies power state in NORMAL mode Each bit represents power state in each power down level. HIGH: Power on LOW: Power down	0x7

8.8.1.219 MAUDIO_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3CC8, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RW	Uses power control feedback to measure power on/off duration of MAUDIO. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x0
USE_SC_ COUNTER	[0]	RW	Uses counter to measure power on/off duration of MAUDIO. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.220 GPS_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3CE0, Reset Value = 0x0000_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
LOCAL_PWR_CFG	[2:0]	RW	Controls power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.221 GPS_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x3CE4, Reset Value = 0x0006_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0xC000
STATUS	[2:0]	R	Verifies power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.222 GPS_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3CE8, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RW	Uses power control feedback to measure power on/off duration of GPS. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x0
USE_SC_ COUNTER	[0]	RW	Use counter to measure power on/off duration of GPS. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

8.8.1.223 GPS_ALIVE_CONFIGURATION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3D00, Reset Value = 0x0000_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
LOCAL_PWR_CFG	[2:0]	RW	Controls power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.224 GPS_ALIVE_STATUS

- Base Address: 0x1002_0000
- Address = Base Address + 0x3D04, Reset Value = 0x0006_0007

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0xC000
STATUS	[2:0]	R	Verifies power state in NORMAL mode 0x7 = Power on 0x0 = Power down	0x7

8.8.1.225 GPS_ALIVE_OPTION

- Base Address: 0x1002_0000
- Address = Base Address + 0x3D08, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
EMULATION	[31]	RW	Uses emulation mode for power off. In emulation mode, PMU performs all power-down sequences.	0x0
RSVD	[30:2]	–	Reserved	0x0
USE_SC_ FEEDBACK	[1]	RW	Uses power control feedback to measure power on/off duration of GPS_ALIVE. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x0
USE_SC_ COUNTER	[0]	RW	Uses counter to measure power on/off duration of GPS_ALIVE. Note that either one of USE_SC_COUNTER and USE_SC_FEEDBACK should be activated.	0x1

NOTE: Either one of USE_SC_FEEDBACK and USE_SC_COUNTER should be activated.

9 Interrupt Controller

9.1 Overview

Generic Interrupt Controller (GIC) is a centralized resource that supports and manages interrupts in a system.

GIC provides:

- Registers for managing interrupt sources, interrupt behavior, and interrupt routing to one or multiple processors
- Support for
  - The ARM architecture Security Extensions
  - Enabling, disabling, and generating processor interrupts from hardware (peripheral) interrupt sources
  - Generating software interrupts
  - Interrupt masking and prioritization

GIC takes the interrupts asserted at the system level and sends appropriate signals to each connected processor. When GIC implements the Security Extensions, it can implement two interrupt requests to a connected processor. The architecture identifies these requests as IRQ and FIQ.

9.1.1 Features

The features of GIC are:

- Supports three interrupt types:
  - Software Generated Interrupt (SGI)
  - Private Peripheral Interrupt (PPI)
  - Shared Peripheral Interrupt (SPI)
- Programmable interrupts that enable you to set the:
  - Security state for an interrupt.
  - Priority level of an interrupt.
  - Enabling or disabling of an interrupt.
  - Processors that receive an interrupt.

9.1.2 Security Extensions Support

The ARM GIC architecture Security Extensions support:

- Configuring each interrupt as either Secure or Non-secure
- Signaling Secure interrupts to the target processor by using either the IRQ or FIQ exception request
- Handling priority of secure and Non-secure interrupts, which is a unified scheme.
- Optional lockdown of the configuration of some Secure interrupts.

In an implementation that includes the Security Extensions:

- System software individually defines each implemented interrupt as either Secure or Non-secure.
- The behavior of processor accesses to registers in the GIC depends on whether the access is Secure or Non-secure. When accessing GIC registers:
  - A Non-secure read of a register field that holds state information for a Secure interrupt returns zero
  - GIC ignores any Non-secure write to a register field that holds state information for a secure interrupt.

Non-secure accesses can only read or set information corresponding to Non-secure interrupts. Secure accesses can read or set information corresponding to both Non-secure and Secure interrupts.

- A Non-secure interrupt signals an IRQ interrupt request to a target processor.
- A Secure interrupt can signal either an IRQ or FIQ interrupt request to a target processor.

9.1.3 Implementation-Specific Configurable Features

During implementation of GIC, the features that depend on the configuration are:

- Exynos 4412 SCP GIC Configuration
- Total 160 interrupts including Software Generated Interrupts (SGIs), Private Peripheral Interrupts (PPIs) and Shared Peripheral Interrupts (SPIs) are supported.
- For SPI, you can service maximal 32 × 4 = 128 interrupt requests.

Table 9-1 describes the GIC configuration values.

Table 9-1 GIC Configuration Values

Items	Configuration Values
AMBA Protocol	AXI
Software Generated Interrupts (SGI)	16
Private Peripheral Interrupts (PPI)	8
Shared Peripheral Interrupts (SPI)	128
Priority Level	256
Legacy interrupt Support	No
Number of CPUs	2
CPU Interface AXI ID Width	10
Distributer AXI ID Width	10
Security Domains	2 (Supports TrustZone technology)
Lockable SPIs	31
Legacy dialog	– (Legacy interrupts are not used)
SGI Register Level Selection	0xF (default value)
SPI Register Level Selection	0x3FF (default value)
PPI Register Level Selection	0xF (default value)
PPI sensitivity	ppi_cx[0] – ppi_cx[5]: edge ppi_cx[6] – ppi_cx[10]: level ppi_cx[11]: edge ppi_cx[12]: level ppi_cx[13] – ppi_cx[14]: edge ppi_cx[15]: level
PPI Registering	Synchronized (for all PPI)
SPI Registering	Synchronized (for all SPI)

9.1.4 Terminology

This section includes:

- Interrupt states
- Interrupt types
- Model for Handling Interrupts
- Processor Security State and Secure and Non-Secure GIC Accesses
- Banking

9.1.4.1 Interrupt States

The states that apply at the interface between GIC and each of the processor supported in the system are:

- **Inactive:** An interrupt that is not active or pending.
- **Pending:** An interrupt from a source to the GIC that is recognized as asserted in hardware or generated by software and is waiting to be serviced by a target processor.
- **Active:** An interrupt from a source to the GIC that has been acknowledged by a processor, and is being serviced but has not completed.
- **Active and Pending:** A processor that is servicing the interrupt and GIC has a pending interrupt from the same source.

9.1.4.2 Interrupt Types

A device that implements this GIC architecture can manage the following types of interrupt:

- **Peripheral Interrupt:** A signal asserts this interrupt to the GIC. The types of peripheral interrupt that GIC architecture defines are:
  - **PPI:** This is a peripheral interrupt that is specific to a single processor.
  - **SPI:** This is a peripheral interrupt that the Distributor can route to any combination of processors.
  - Each peripheral interrupt is either Edge-Triggered or Level-Sensitive:
    - **Edge-triggered:** This is an interrupt that is asserted on detection of a rising edge of an interrupt signal and then, regardless of the state of the signal, remains asserted until it is cleared by the conditions defined by this specification.
    - **Level-sensitive:** This is an interrupt that is asserted whenever the interrupt signal level is HIGH, and disserted whenever the level is LOW.

9.1.4.2.1 Software-Generated Interrupt

Software generates this interrupt when writing to a specific register in GIC. The system uses SGIs for inter-processor communication. A software interrupt has edge-triggered properties. On a peripheral input, the software triggering of the interrupt is equivalent to the edge transition of the interrupt signal. Model for Handling Interrupts

In a multiprocessor implementation, there are two models for handling interrupts:

- **1-N model:** Only one processor handles this interrupt. The system must implement a mechanism to determine which processor handles an interrupt that is programmed to target more than one processor.
- **N-N model:** All processors receive the interrupt independently. When a processor acknowledges the interrupt, the interrupt pending state is cleared only for that processor. The interrupt remains pending for the other processors.

9.1.4.3 Spurious Interrupts

It is possible that an interrupt that the GIC has signaled to a processor is no longer required. So, when the processor acknowledges the interrupt, GIC returns a special interrupt ID that identifies the interrupt as a spurious interrupt.

This occurs due to:

- The change in state of the interrupt.
- Software that has re-programmed the GIC to change the processing requirements for the interrupt.
- The interrupt that is handling the 1-N model and the other processor has acknowledged the interrupt.

9.1.4.4 Processor Security State and Secure and Non-Secure GIC Accesses

A processor that implements the ARM Security Extensions has a security state:

Secure or Non-secure:

- A processor in the Non-secure state can make only Non-secure accesses to a GIC.
- A processor in the Secure state can make both Secure and Non-secure accesses to a GIC.
- Software that is running in Non-secure state is described as Non-secure software.
- Software that is running in Secure state is described as Secure software.

9.1.4.5 Banking

This section includes:

- Interrupt Banking
- Register Banking

9.1.4.5.1 Interrupt Banking

In a multiprocessor implementation, for PPIs and SGIs, GIC can have multiple interrupts with the same interrupt ID. Such an interrupt is called a banked interrupt and is identified uniquely by the combination of its interrupt ID and its associated CPU interface.

9.1.4.5.2 Register Banking

Register banking refers to implementing multiple copies of a register at the same address. This occurs in:

- Multiprocessor implementation for some registers that are corresponding to banked interrupts
- GIC that implements the Security Extensions to provide separate secure and Non-secure copies of some registers.

9.2 Interrupt Source

This section includes:

- Interrupt source connection
- GIC interrupt table

9.2.1 Interrupt Sources Connection

Figure 9-1 illustrates the interrupt sources connection.

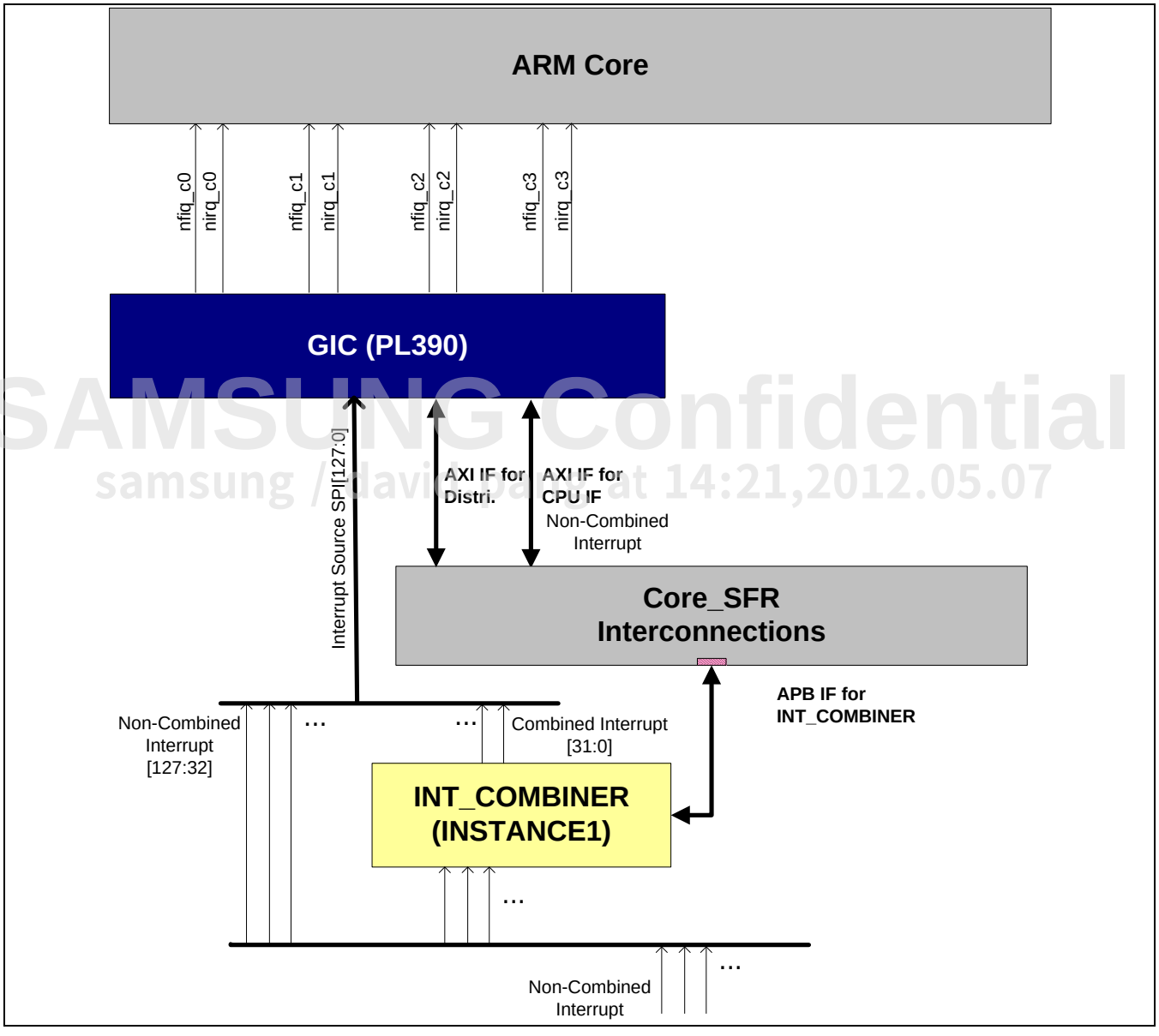


Figure 9-1 Interrupt Sources Connection

GIC interrupt sources are passed INT_COMBINER block that is combined interrupt sources for GIC.

9.2.2 GIC Interrupt Table

Total 160 interrupts including Software Generated Interrupts (SGIs[15:0], ID[15:0]), Private Peripheral Interrupts (PPIs[15:0], ID[31:16]) and Shared Peripheral Interrupts (SPIs[127:0], ID[159:32]) are supported. For SPI, you can service a maximal 32 × 4 = 128 interrupt requests.

Table 9-2 describes the GIC interrupt (SPI[127:]).

Table 9-2 GIC Interrupt Table (SPI[127:0])

SPI Port No	ID	Int_I_Combiner	Interrupt Source	Source Block
127	159	–	G3D_IRQGP	–
126	158	–	G3D_IRQPP3	–
125	157	–	G3D_IRQPP2	–
124	156	–	G3D_IRQPP1	–
123	155	–	G3D_IRQPP0	–
122	154	–	G3D_IRQGPMMU	–
121	153	–	G3D_IRQPPMMU3	–
120	152	–	G3D_IRQPPMMU2	–
119	151	–	G3D_IRQPPMMU1	–
118	150	–	G3D_IRQPPMMU0	–
117	149	–	G3D_IRQPMU	–
116	148	–	C2C_SSCM[1]	–
115	147	–	TSI	–
114	146	–	CEC	–
113	145	–	SLIMBUS	–
112	144	–	SSS	–
111	143	–	GPS	–
110	142	–	PMU	–
109	141	–	KEYPAD	–
108	140	IntG17_7	L2_IRQ	MCT
		IntG17_6	Reserved	
		IntG17_5	SYSMMU_ISP_CX[1]	
		IntG17_4	SYSMMU_FIMC_FD[1]	–
		IntG17_3	SYSMMU_FIMC_DRC[1]	–
		IntG17_2	SYSMMU_FIMC_ISP[1]	–
		IntG17_1	SYSMMU_FIMC_Lite1[1]	–
		IntG17_0	SYSMMU_FIMC_Lite0[1]	–
107	139	IntG16_7	L3_IRQ	MCT
		IntG16_6	Reserved	
		IntG16_5	SYSMMU_ISP_CX[0]	

SPI Port No	ID	Int_I_Combiner	Interrupt Source	Source Block
		IntG16_4	SYSMMU_FIMC_FD[0]	–
		IntG16_3	SYSMMU_FIMC_DRC[0]	–
		IntG16_2	SYSMMU_FIMC_ISP[0]	–
		IntG16_1	SYSMMU_FIMC_Lite1[0]	–
		IntG16_0	SYSMMU_FIMC_Lite0[0]	–
106	138	–	FIMC_lite1	–
105	137	–	FIMC_lite0	–
104	136	–	SPDIF	–
103	135	–	PCM2	–
102	134	–	PCM1	–
101	133	–	PCM0	–
100	132	–	AC97	–
99	131	–	I2S2	–
98	130	–	I2S1	–
97	129	–	I2S0	–
96	128	–	AUDIO_SS	–
95	127	–	ISP[1]	–
94	126	–	MFC	–
93	125	–	HDMI_I2C	–
92	124	–	HDMI	–
91	123	–	MIXER	–
90	122	–	ISP[0]	–
89	121	–	G2D	–
88	120	–	JPEG	–
87	119	–	FIMC3	–
86	118	–	FIMC2	–
85	117	–	FIMC1	–
84	116	–	FIMC0	–
83	115	–	ROTATOR	–
82	114	–	Reserved	–
81	113	–	Reserved	–
80	112	–	MIPI_CSI_2LANE	–
79	111	–	MIPI_DSI_4LANE	–
78	110	–	MIPI_CSI_4LANE	–
77	109	–	SDMMC	–
76	108	–	HSMMC3	–
75	107	–	HSMMC2	–

SPI Port No	ID	Int_I_Combiner	Interrupt Source	Source Block
74	106	–	HSMMC1	–
73	105	–	HSMMC0	–
72	104	–	GPIO_C2C	–
71	103	–	HSOTG	–
70	102	–	UHOST	USB HOST
69	101	–	G1_IRQ	MCT
68	100	–	SPI2	–
67	99	–	SPI1	–
66	98	–	SPI0	–
65	97	–	I2C7	–
64	96	–	I2C6	–
63	95	–	I2C5	–
62	94	–	I2C4	–
61	93	–	I2C3	–
60	92	–	I2C2	–
59	91	–	I2C1	–
58	90	–	I2C0	–
57	89	–	G0_IRQ	–
56	88	–	Reserved	–
55	87	–	UART3	–
54	86	–	UART2	–
53	85	–	UART1	–
52	84	–	UART0	–
51	83	–	NFC	–
50	82	–	IEM_IEC	–
49	81	–	IEM_APC	–
48	80	IntG18_7	Reserved	–
		IntG18_6	CPU_nIRQOUT[2]	–
		IntG18_5	PARITYFAILSCU[2]	Parity fails for SCU from CPU2
		IntG18_4	PARITYFAIL2	L1 parity fails for CPU2
		IntG18_3	nCTIIRQ[2]	F4Q CTI interrupt for CPU2
		IntG18_2	PMUIRQ[2]	F4Q PMU interrupt from CPU2
		IntG18_1	Reserved	–
		IntG18_0	L1_IRQ	MCT
47	79	–	GPIO_LB	–
46	78	–	GPIO_RT	–
45	77	–	RTC_TIC	–

SPI Port No	ID	Int_I_Combiner	Interrupt Source	Source Block
44	76	–	RTC_ALARM	–
43	75	–	WDT	–
42	74	IntG19_7	Reserved	–
		IntG19_6	CPU_nIRQOUT[3]	–
		IntG19_5	PARITYFAILSCU[3]	Parity fails for SCU from CPU3
		IntG19_4	PARITYFAIL3	L1 parity fails for CPU3
		IntG19_3	nCTIIRQ[3]	F4Q CTI interrupt for CPU3
		IntG19_2	PMUIRQ[3]	F4Q PMU interrupt from CPU3
		IntG19_1	Reserved	–
		IntG19_0	L0_IRQ	MCT
41	73	–	TIMER4	–
40	72	–	TIMER3	–
39	71	–	TIMER2	–
38	70	–	TIMER1	–
37	69	–	TIMER0	–
36	68	–	PDMA1	–
35	67	–	PDMA0	–
34	66	–	MDMA	–
33	65	–	C2C_SSCM[0]	–
32	64	–	EINT16_31	External Interrupt
31	63	–	EINT[15]	External Interrupt
30	62	–	EINT[14]	External Interrupt
29	61	–	EINT[13]	External Interrupt
28	60	–	EINT[12]	External Interrupt
27	59	–	EINT[11]	External Interrupt
26	58	–	EINT[10]	External Interrupt
25	57	–	EINT[9]	External Interrupt
24	56	–	EINT[8]	External Interrupt
23	55	–	EINT[7]	External Interrupt
22	54	–	EINT[6]	External Interrupt
21	53	–	EINT[5]	External Interrupt
20	52	–	EINT[4]	External Interrupt
19	51	–	EINT[3]	External Interrupt
18	50	–	EINT[2]	External Interrupt
17	49	–	EINT[1]	External Interrupt
16	48	–	EINT[0]	External Interrupt
15	47	IntG15_7	DECERRINTR	F4D

SPI Port No	ID	Int_I_Combiner	Interrupt Source	Source Block
		IntG15_6	SLVERRINTR	F4D
		IntG15_5	ERRRDINTR	F4D
		IntG15_4	ERRRTINTR	F4D
		IntG15_3	ERRWDINTR	F4D
		IntG15_2	ERRWTINTR	F4D
		IntG15_1	ECNTRINTR	F4D
		IntG15_0	SCUEVABORT	F4D
14	46	IntG14_6	CPU_nIRQOUT[1]	F4D
		IntG14_5	Reserved	–
		IntG14_4	Reserved	–
		IntG14_3	Reserved	–
		IntG14_2	Reserved	–
		IntG14_1	Reserved	–
		IntG14_0	Reserved	–
13	45	IntG13_5	CPU_nIRQOUT[0]	F4D
		IntG13_4	Reserved	–
		IntG13_3	Reserved	–
		IntG13_2	Reserved	–
		IntG13_1	Reserved	–
		IntG13_0	Reserved	–
12	44	IntG12_7	G3	MCT
		IntG12_6	G2	
		IntG12_5	G1	
		IntG12_4	G0	
		IntG12_3	Reserved	–
		IntG12_2	Reserved	–
		IntG12_1	MIPI_HSI	MIPI
		IntG12_0	UART4	UART
11	43	IntG11_3	LCD0[3]	–
		IntG11_2	LCD0[2]	–
		IntG11_1	LCD0[1]	–
		IntG11_0	LCD0[0]	–
10	42	IntG10_7	DMC1_PPC_PEREV_M	DMC1
		IntG10_6	DMC1_PPC_PEREV_A	DMC1
		IntG10_5	DMC0_PPC_PEREV_M	DMC0
		IntG10_4	DMC0_PPC_PEREV_A	DMC0
		IntG10_3	ADC	General ADC

SPI Port No	ID	Int_I_Combiner	Interrupt Source	Source Block
		IntG10_2	L2CACHE	F4D
		IntG10_1	RP_TIMER	–
		IntG10_0	GPIO_AUDIO	–
9	41	IntG9_7	PPMU_ISP_X	PPMU for ISP X
		IntG9_6	PPMU_MFC_M1	PPMU for MFC_M1
		IntG9_5	PPMU_MFC_M0	PPMU for MFC_M0
		IntG9_4	PPMU_3D	PPMU for 3D
		IntG9_3	PPMU_TV_M0	PPMU for TV_M0
		IntG9_2	PPMU_FILE_D_M0	PPMU for FILE_D_M0
		IntG9_1	PPMU_ISP_MX	PPMU for ISP MX
		IntG9_0	PPMU_LCD0	PPMU for LCD0
8	40	IntG8_7	PPMU_IMAGE_M0	PPMU for IMAGE_M0
		IntG8_6	PPMU_CAMIF_M0	PPMU for CAMIF_M0
		IntG8_5	PPMU_D_RIGHT_M0	PPMU for D_right_M0
		IntG8_4	PPMU_D_LEFT_M0	PPMU for D_left_M0
		IntG8_3	PPMU_ACP0_M0	PPMU for ACP0_M0
		IntG8_2	PPMU_XIU_R_S1	PPMU for XIU_R_S1
		IntG8_1	PPMU_XIU_R	PPMU for XIU_R
		IntG8_0	PPMU_XIU_L	PPMU for XIU_L
7	39	IntG7_7	Reserved	–
		IntG7_6	SYSMMU_MFC_M1[1]	System MMU for MFC_M1
		IntG7_5	SYSMMU_MFC_M0[1]	System MMU for MFC_M0
		IntG7_4	SYSMMU_TV_M0[1]	System MMU for TV_M0
		IntG7_3	Reserved	–
		IntG7_2	SYSMMU_LCD0_M0[1]	System MMU for LCD0_M0
		IntG7_1	SYSMMU_GPS[1]	System MMU for GPS
		IntG7_0	SYSMMU_ROTATOR[1]	System MMU for Rotator
6	38	IntG6_7	SYSMMU_2D[1]	System MMU for 2D
		IntG6_6	SYSMMU_JPEG[1]	System MMU for JPEG
		IntG6_5	SYSMMU_FIMC3[1]	System MMU for FIMC3
		IntG6_4	SYSMMU_FIMC2[1]	System MMU for FIMC2
		IntG6_3	SYSMMU_FIMC1[1]	System MMU for FIMC1
		IntG6_2	SYSMMU_FIMC0[1]	System MMU for FIMC0
		IntG6_1	SYSMMU_SSS[1]	System MMU for SSS
		IntG6_0	SYSMMU_MDMA[1]	System MMU for MDMA
5	37	IntG5_7	Reserved	–
		IntG5_6	SYSMMU_MFC_M1[0]	System MMU for MFC_M1

SPI Port No	ID	Int_I_Combiner	Interrupt Source	Source Block
		IntG5_5	SYSMMU_MFC_M0[0]	System MMU for MFC_M0
		IntG5_4	SYSMMU_TV_M0[0]	System MMU for TV_M0
		IntG5_3	Reserved	–
		IntG5_2	SYSMMU_LCD0_M0[0]	System MMU for LCD0_M0
		IntG5_1	SYSMMU_GPS[0]	System MMU for GPS
		IntG5_0	SYSMMU_ROTATOR[0]	System MMU for Rotator
4	36	IntG4_7	SYSMMU_2D[0]	System MMU for 2D
		IntG4_6	SYSMMU_JPEG[0]	System MMU for JPEG
		IntG4_5	SYSMMU_FIMC3[0]	System MMU for FIMC3
		IntG4_4	SYSMMU_FIMC2[0]	System MMU for FIMC2
		IntG4_3	SYSMMU_FIMC1[0]	System MMU for FIMC1
		IntG4_2	SYSMMU_FIMC0[0]	System MMU for FIMC0
		IntG4_1	SYSMMU_SSS[0]	System MMU for SSS
		IntG4_0	SYSMMU_MDMA[0]	System MMU for MDMA
3	35	IntG3_6	nCTIIRQ_ISP	ISP CTI interrupt
		IntG3_5	PMUIRQ_ISP	ISP PMU interrupt
		IntG3_4	TMU	–
		IntG3_3	nCTIIRQ[1]	F4D CTI interrupt for CPU1
		IntG3_2	PMUIRQ[1]	F4D PMU interrupt from CPU1
		IntG3_1	PARITYFAILSCU[1]	Parity fails for SCU from CPU1
		IntG3_0	PARITYFAIL1	L1 parity fails for CPU1
2	34	IntG2_6	PARRINTR	Parity error on L2 tag RAM
		IntG2_5	PARRDINTR	Parity error on L2 data RAM
		IntG2_4	TMU	–
		IntG2_3	nCTIIRQ[0]	F4D CTI interrupt for CPU0
		IntG2_2	PMUIRQ[0]	F4D PMU interrupt from CPU0
		IntG2_1	PARITYFAILSCU[0]	Parity fails for SCU from CPU0
		IntG2_0	PARITYFAIL0	L1 parity fails for CPU0
1	33	IntG1_3	TZASC1[1]	–
		IntG1_2	TZASC1[0]	–
		IntG1_1	TZASC0[1]	–
		IntG1_0	TZASC0[0]	–
0	32	IntG0_3	MDNIE_LCD0[3]	–
		IntG0_2	MDNIE_LCD0[2]	–
		IntG0_1	MDNIE_LCD0[1]	–
		IntG0_0	MDNIE_LCD0[0]	–

Table 9-3 describes the GIC interrupt (PPI[15:0]).

Table 9-3 GIC Interrupt Table (PPI[15:0])

PPI Port No	ID	Interrupt Source	Source Block
15	31	Reserved	–
14	30	Reserved	–
13	29	Reserved	–
12	28	L3_IRQ (for CPU3) or L2_IRQ (for CPU2) or L1_IRQ (for CPU1) or L0_IRQ (for CPU0)	MCT
11	27	Reserved	–
10	26	G3_IRQ (for CPU3) or G2_IRQ (for CPU2) or G1_IRQ (for CPU1) or G0_IRQ (for CPU0)	MCT
9	25	Reserved	–
8	24	Reserved	–
7	23	Reserved	–
6	22	Reserved	–
5	21	Reserved	–
4	20	Reserved	–
3	19	Reserved	–
2	18	Reserved	–
1	17	Reserved	–
0	16	Reserved	–

9.3 Functional Overview

This section includes:

- Functional interfaces
- Distributor
- CPU Interfaces

9.3.1 Functional Interfaces

Figure 9-2 illustrates the GIC block diagram.

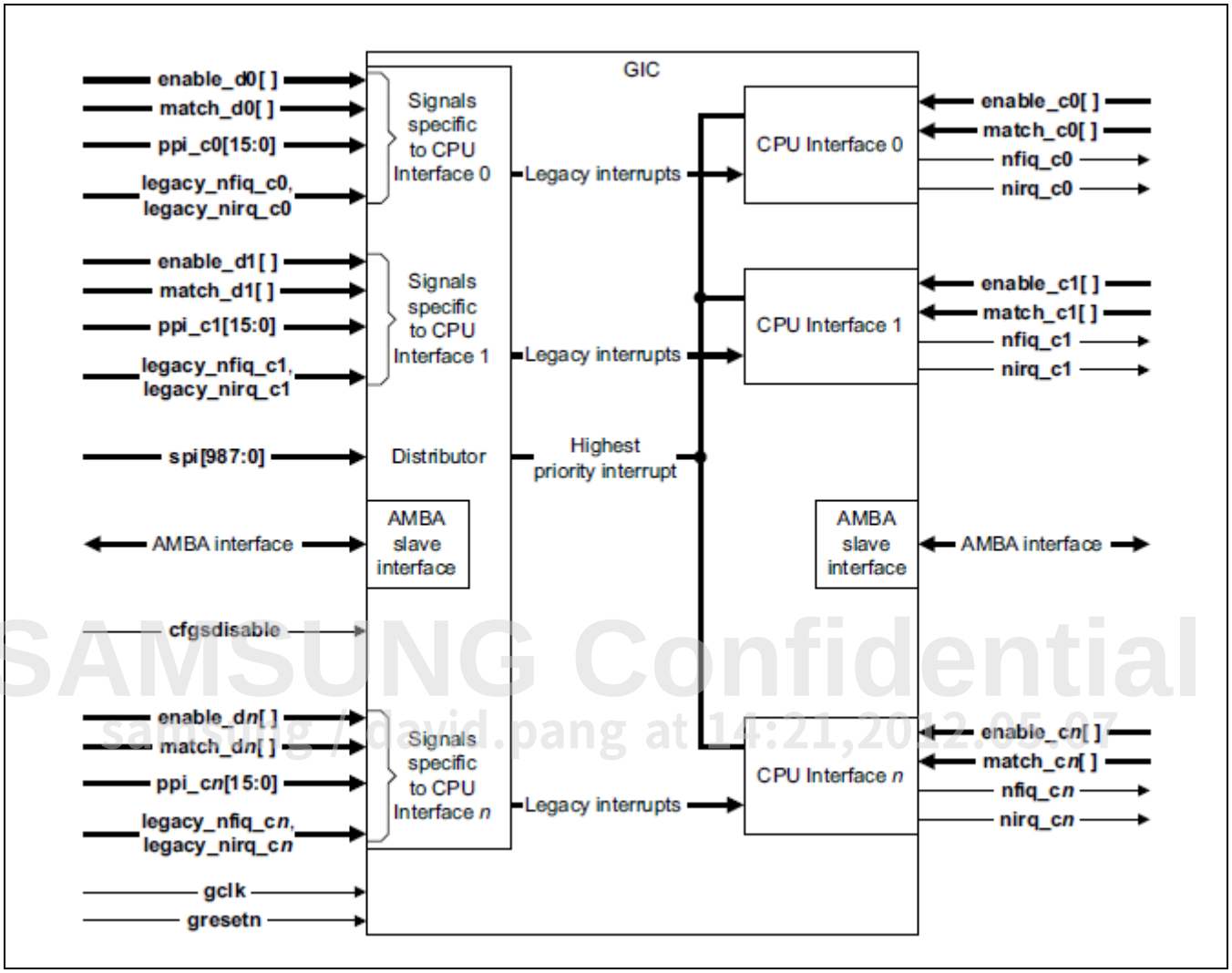


Figure 9-2 GIC Block Diagram

NOTE: Figure 9-2 illustrates all the signals but some configurations of GIC might not contain all of these signals.

The main blocks of the GIC are:

- AMBA slave interface
- Distributor
- CPU interface
- Clock and reset
- Enable and match signals

9.3.1.1 AMBA Slave Interfaces

The AMBA slave interfaces provide access to the GIC registers that enable you to program the system configuration parameters and obtain status information.

GIC provides two AMBA slave interfaces: One for the Distributor and one that the CPU interfaces share.

- AXI slave interface

Both the AXI slave interfaces use a 32-bit data bus.

The AXI slave interfaces include AXI channels, which are:

- Write-Address (AW)
- Write-Data (W)
- Write-response (B)
- Read-Address (AR)
- Read-Data (R)

9.3.1.2 Distributor

Distributor receives interrupts and provides the highest priority interrupt to the corresponding CPU interface. An interrupt with a lower priority is forwarded to the appropriate CPU Interfaces when it becomes the highest priority pending interrupt.

Figure 9-3 illustrates the Distributor.

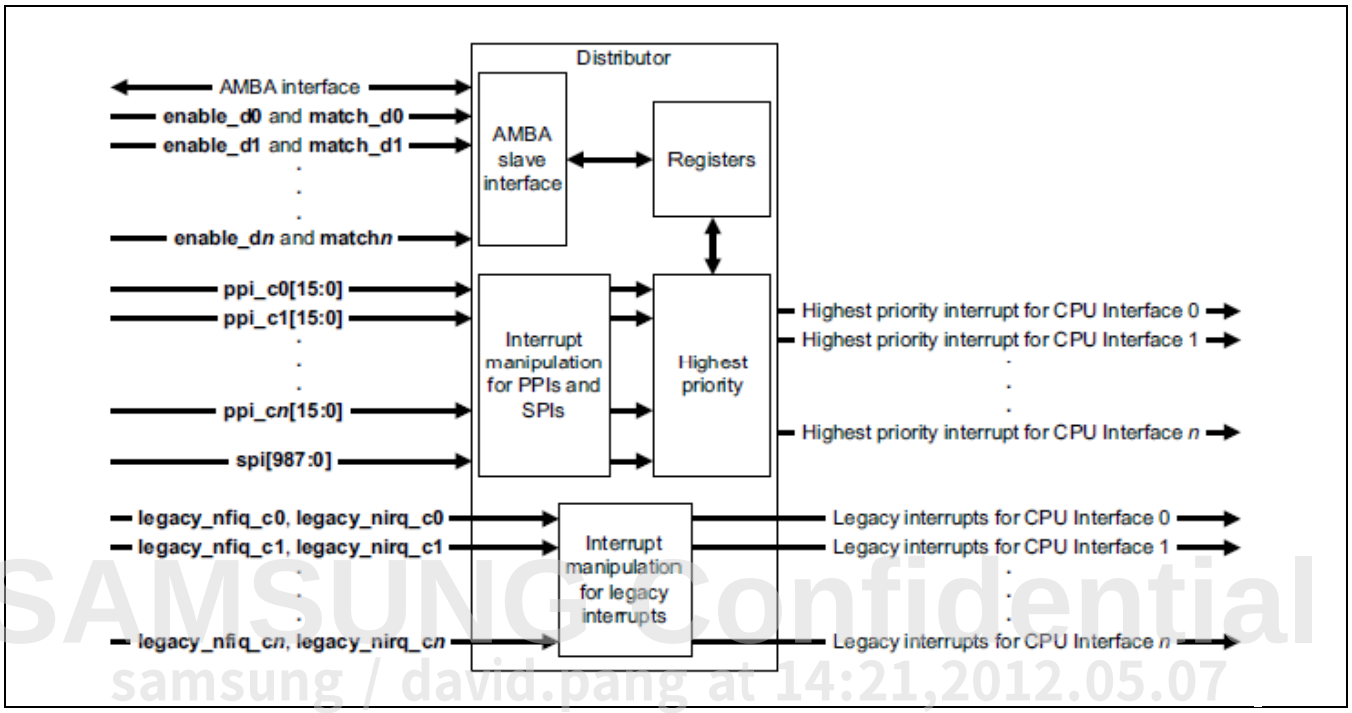


Figure 9-3 Distributor

In multiprocessor configurations, the Distributor provides up to 16 PPIs and 1–988 SPIs for each CPU interfaces. . For each SPI, you can program Distributor to control how many CPU Interfaces it routes the interrupt to.

9.3.1.3 CPU Interface

A CPU interface contains a programmable interrupt priority mask.

CPU interface accepts pending interrupts only if the priority of the interrupt is higher than the:

- Programmed interrupt priority mask.
- Interrupts that the processor is currently servicing.

Figure 9-4 illustrates the CPU interface.

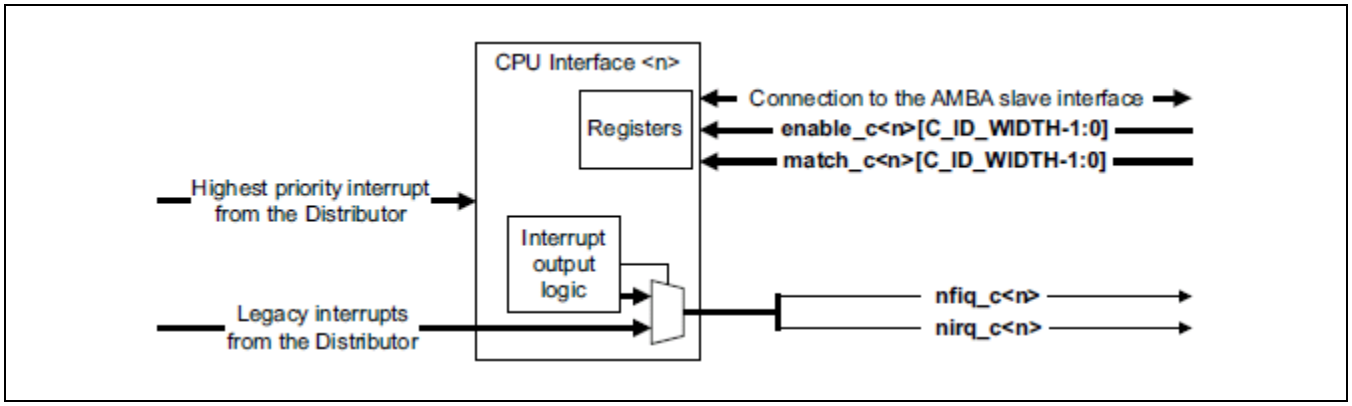


Figure 9-4 CPU Interface

9.3.1.4 Clock and Reset

Table 9-4 describes the two signals: GCLK and GRESETN.

Table 9-4 Clock and Reset

Signal	Type	Sources	Description
gclk	Input	Clock source	Clock for the GIC
gresetn	Input	Reset source	Reset for the GIC. This signal is active LOW.

9.3.2 The Distributor

The Distributor centralizes all interrupt sources and determines the priority of each interrupt. For each CPU interface, the Distributor dispatches the interrupt with the highest priority to the interface for priority masking and preemption handling.

The Distributor provides a programming interface for:

- Enabling the forwarding of interrupts to the CPU interfaces globally.
- Enabling or disabling each interrupt.
- Setting the priority level of each interrupt.
- Setting the target processor list of each interrupt.
- Setting each peripheral interrupt to be level-sensitive or edge-triggered.
- Setting each interrupt as either secure or Non-secure if the GIC implements the Security Extensions.
- Sending an SGI to one or more target processors.

The Distributor also provides:

- Visibility of the state of each interrupt.
- A mechanism for software to set or clear the pending state of a peripheral interrupt.

9.3.2.1 Interrupt IDs

Interrupts from sources are identified using ID numbers. Each CPU interface can see up to 1020 interrupts. The distributor supports up to 1244 interrupts because of banking of SPIs and PPIs.

GIC assigns interrupt ID numbers ID0-ID1019 as follows:

- Interrupt numbers ID32-ID1019 are used for SPIs.
- Interrupt numbers ID0-ID31 are used for interrupts that are private to a CPU interface and are banked in the Distributor, in a banked interrupt, the Distributor can have multiple interrupts with the same ID. You can identify a banked interrupt uniquely by its ID number and its associated CPU interface number. The bank interrupt IDs that are used for SGIs and PPIs, respectively are:
  - ID0-ID15
  - ID16-ID31
  - In a multiprocessor system:
    - A PPI is signaled to a particular CPU interface, and is private to that interface. While prioritizing interrupts for a CPU interface, the distributor considers only the PPIs that are signaled to that interface.
    - Each connected processor issues an SGI by writing to the ICDSGIR in the Distributor. Refer to the Software Generated Interrupt Register (ICDSGIR). Each SGI can target multiple processors. In the distributor and in a targeted processor, an SGI is identified uniquely by the combination of its interrupt number, ID0-ID15, and the processor source ID, CPUID0-CPUID7, of the processor that issued the SGI. Banking SGIs refers to the GIC that can handle multiple software interrupts simultaneously without resource conflicts. The Distributor ignores any write to the ICDSGIR that is not from a processor that is connected to one of the CPU interfaces.

The system software sets the priority of each interrupt independent of its interrupt number.

9.3.3 CPU Interfaces

Each CPU interface block provides:

- Interface for a processor that operates with the GIC.
- Programming interface for:
  - Enabling the signaling of interrupt requests by the CPU interface.
  - Acknowledging an interrupt.
  - Indicating completion of the processing of an interrupt.
  - Setting an interrupt priority mask for the processor.
  - Defining the preemption policy for the processor.
  - Determining the highest priority pending interrupt for the processor.

When enabled, a CPU interface takes the highest priority pending interrupt for its connected processor CPU interface determines whether the interrupt has sufficient priority for it to signal the interrupt request to the processor.

To determine whether to signal the interrupt request to the processor, the CPU interface considers the interrupt priority mask and the preemption settings for the processor. At any time, the connected processor can read the priority of its highest priority active interrupt from a CPU interface register.

The processor acknowledges the interrupt request by reading the CPU interface Interrupt Acknowledge Register. The CPU interface returns one of these:

- The ID number of the highest priority pending interrupt. The CPU returns this ID number if that interrupt is of sufficient priority to generate an interrupt exception on the processor. This is the normal response to an interrupt acknowledgement.
- Exceptionally, an ID number that indicates a spurious interrupt.

When the processor acknowledges the interrupt at the CPU interface, the Distributor changes the status of the interrupt from pending to either active, or active and pending. Now, the CPU interface can signal another interrupt to the processor to preempt interrupts that are active on the processor.

When there is no pending interrupt with sufficient priority for signaling to the processor, the interface disserts the interrupt request signal to the processor.

When the interrupt handler on the processor has completed the processing of an interrupt, it writes to the CPU interface to indicate interrupt completion. Hence, the distributor changes the status of the interrupt either:

- From active to inactive or
- From active and pending to pending.

9.4 Interrupt Handling and Prioritization

This section includes:

- About interrupt handling and prioritization
- General handling of interrupts
- Interrupt prioritization
- The effect of the Security Extensions on interrupt handling
- The effect of the Security Extensions on interrupt prioritization

9.4.1 About Interrupt Handling and Prioritization

Interrupt handling describes:

- How GIC recognizes interrupts.
- How software programs GIC to configure and control interrupts.
- How GIC maintains the state machine for each interrupt on each CPU interface.
- How the exception model of processor interacts with GIC.

Prioritization describes:

- The configuration and control of interrupt priority.
- The order of execution of pending interrupts.
- The determination of when interrupts are visible to target processor, which includes:
  - Interrupt priority masking.
  - Interrupt grouping.
  - Preemption of an active interrupt.

9.4.1.1 Handling Different Interrupt Types in a Multiprocessor System

GIC supports peripheral interrupts and SGIs.

In a multiprocessor implementation, the GIC handles:

- SGIs that use an N-N model.
- Peripheral (Hardware) interrupts that use a 1-N model.

9.4.1.2 Identifying the Supported Interrupts

GIC defines different ID values for the different types of interrupt. However, there is no requirement for GIC to implement a continuous block of interrupt IDs for any interrupt type.

To handle interrupts efficiently, software should know what interrupt IDs are supported by the GIC. Software can use ICDISERs to discover this information.

When the processor implements the Security Extensions, Secure software determines which interrupts are visible to Non-secure software. The Non-secure software should know which interrupts it can see, and might use this discovery process to find this information.

ICDISER0 provides the Set-enables bits for:

- SGIs: SGIs use interrupt IDs 15-0 that correspond to register bits[15:0].
- PPIs: PPIs use interrupt IDs 31-16 that correspond to register bits[31:16].

The remaining ICDISERs, from ICDISER1, provide the Set-enable bits for the SPIs, starting at interrupt ID 32.

Software identifies those interrupts that are supported:

1. Read the ICDICTR. Refer to Interrupt Controller Type Register (ICDICTR). The ITLinesNumber field identifies the number of implemented ICDISERs, and therefore the maximum number of SPIs that might be supported.
2. Write 0 to the ICDDCR. Enable bit to disable forwarding of interrupts to CPU interfaces. Refer to Distributor Control Register for more information.
3. For each implemented ICDISER that start with ICDISER0:
   - Write 0xFFFFFFFF to the ICDISER.
   - Read the value of the ICDISER. Bits that read as 1 correspond to supported interrupt IDs.

Software uses the ICDICERs to discover which interrupts are enabled permanently. Refer to Interrupt Clear-Enable Registers (ICDICERn) for more information. It does this discovery as follows. For each implemented ICDICER, starting with ICDICER0:

- Write 0xFFFFFFFF to the ICDICER. This disables all interrupts that can be disabled.
- Read the value of the ICDICER. Bits that read as 1 correspond to interrupts that are enabled permanently.
- Write 1 to any bits in the ICDICER that correspond to interrupts that should be re-enabled.

GIC implements the same number of ICDISERs and ICDICERs.

When software has completed its discovery, it writes 1 to the ICDDCR. Enable bit to enable forwarding of interrupts to CPU interfaces.

If GIC implements the Security Extensions, software can use secure accesses to:

- Discover all the supported interrupt IDs.
- Write to the ICDISRs. This is to configure interrupts as secure or Non-secure. Refer to Interrupt Security Registers (ICDISRn).

Software that uses the Non-secure accesses can discover only the interrupts that are configured as Non-secure.

After the software discovers its supported interrupts, when secure software changes the security configuration of any interrupts, it should communicate the effect of those changes to the Non-secure software.

9.4.2 General Handling of Interrupts

The Distributor maintains a state machine for each supported interrupt on each CPU interface. The possible states of an interrupt are:

- Inactive
- Pending
- Active
- Active and pending

When GIC recognizes an interrupt request, it marks its states as pending. Regenerating a pending interrupt does not affect the state of the interrupt.

GIC operates on interrupts as follows:

1. GIC determines whether each interrupt is enabled. A disabled interrupt has no further effect on GIC.
2. For each enabled interrupt that is pending, the Distributor determines the targeted processor or processors.
3. For each processor, the Distributor determines the highest priority pending interrupt. This is based on the priority information that it holds for each interrupt and forwards the interrupt to the CPU interface.
4. The CPU interface compares the interrupt priority with the current interrupt priority for the processor. This comparison is determined by a combination of the Priority Mask Register, the current preemption settings, and the highest priority active interrupt for the processor. If the interrupt has sufficient priority, GIC signals an interrupt exception request to the processor.
5. When the processor takes the interrupt exception, it reads the ICCIAR in its CPU interface to acknowledge the interrupt. Refer to Interrupt Acknowledge Register (ICCIAR). This read returns an Interrupt ID that the processor uses to select the correct interrupt handler. When it recognizes this read, GIC changes the state of the interrupt:
   - From pending to active and pending if the pending state of the interrupt persists when the interrupt becomes active, or if the interrupt is generated again.
   - Otherwise, GIC changes its state from pending to active.

NOTE:

1. A level-sensitive peripheral interrupt persists when it is acknowledged by the processor. This is because the interrupt signal to the GIC remains asserted until the interrupt service routine (ISR) that running on the peripheral is asserting the signal.
2. In a multiprocessor implementation, the GIC handles:
   - SGIs: SGIs use N-N model, where the acknowledgement of an interrupt by one processor has no effect on the state of the interrupt on other CPU interfaces.
   - Peripheral interrupts: Peripheral interrupts use 1-N model, where the acknowledgement of an interrupt by one processor removes the pending status of the interrupt on any other targeted processors.

6. When the processor has completed handling the interrupt, it signals this completion by writing to the ICCEOIR in GIC, Refer to End of Interrupt Register (ICCEOIR).
   The GIC requires the order of completion of interrupts by a particular processor to be the reverse of the order of acknowledgement so the last interrupt acknowledged must be the first interrupt completed.
   When the processor writes to the ICCEOIR, the GIC changes the state of the interrupt for the corresponding CPU interface, either:
   - From active to inactive Or
   - From active and pending to pending

If there is no pending interrupt of sufficient priority for the CPU interface to signal it to the processor, the interface de-asserts the interrupt exception request to the processor.

A CPU interface never signals any interrupt to the connected processor that is active and pending. However, it only signals interrupts that are pending and have sufficient priority:

- For SPIs, the interface never signals any interrupt that is active and pending on any CPU interface.
- For SGIs, the interface never signals any interrupt that is active and pending on this interface. However, it does not consider whether the interrupt is active and pending on any other interface.
- Any PPI is private to this interface and the interface does not signal it if it is active and pending.

9.4.2.1 Interrupt Controls in GIC

This sub-section includes:

- Enabling interrupt
- Setting and clearing pending state of an interrupt
- Finding the active or pending state of an interrupt
- Generating an SGI

9.4.2.1.1 Enabling Interrupt

For peripheral interrupts, a processor:

- Enables an interrupt by writing to the appropriate ICDISER bit. Refer to Interrupt Set-Enables Registers (ICDISERn) for more information.
- Disables an interrupt by writing to the appropriate ICDICER bit. Refer to Interrupt Clear-Enable Registers (ICDICERn) for more information.

Writes to the ICDISERs and ICDICERs control whether the Distributor forwards interrupts to the CPU interfaces. Disabling an interrupt by writing to the appropriate ICDICER does not prevent that interrupt from changing state, for example, changing state to pending.

9.4.2.1.2 Setting and Clearing Pending State of an Interrupt

For peripheral interrupts, a processor:

- Sets the pending state by writing to the appropriate ICDISPR bit. Refer to Interrupt Set-Pending Registers (ICDISPRn) for more information.
- Clears the pending state by writing to the appropriate ICDICPR bit. Refer to Interrupt Clear-Pending Registers (ICDICPRn) for more information.

For a level-sensitive interrupt:

- If the hardware signal of an interrupt is asserted when a processor writes to the corresponding ICDICPR bit then the write to the register has no effect on the pending state of the interrupt.
- If a processor writes a "1" to an ICDISPR bit, then the corresponding interrupt becomes pending regardless of the state of the hardware signal of that interrupt. The interrupt remains pending regardless of the assertion or de-assertion of the signal.

For SGIs, GIC ignores writes to the corresponding ICDISPR and ICDISCR bits. A processor cannot change the state of a SGI by writing to these registers.

9.4.2.1.3 Finding the Active or Pending State of an Interrupt

A processor can find:

- The pending state of an interrupt by reading the corresponding ICDISPR or ICDICPR bit.
- The active state of an interrupt by reading the corresponding ICDABR bit. Refer to Active Bit Registers (ICDABRn) for more information.

If the interrupt is pending or active, the corresponding register bit is set to1. If an interrupt is pending and active, the corresponding bit is set to1 in both registers.

For an SGI, the corresponding ICDISPR and ICDICPR bits read-as-one (RAO). This happens when there is a pending interrupt from at least one generating processor that targets the processor reading the ICDISPR or ICDICPR.

9.4.2.1.4 Generating an SGI

A processor generates an SGI by writing to an ICDSGIR. Refer to Software-Generated Interrupt Register for more information. An SGI can target multiple processors, and the ICDSGIR write specifies the target processor list. The ICDSGIR includes optimization for:

- Interrupting only the processor that writes to the ICDSGIR
- Interrupting all processors other than the one that writes to the ICDSGIR.

SGIs from different processors use the same interrupt IDs. Therefore, any target processor can receive SGIs with the same interrupt ID from different processors. On the CPU interface of the target processor, the pending status of each of these interrupts is independent of the pending status of any other interrupt. However, only one interrupt with this ID can be active. Reading the ICCIAR for an SGI returns both the interrupt ID and the CPU ID of the processor that generated the interrupt, uniquely identifying the interrupt.

In a multiprocessor implementation, the interrupt priority of each SGI interrupt ID is defined independently for each CPU interface, Refer to Interrupt Priority Registers (ICDIPRn). This means that, for each CPU interface, all SGIs with a particular interrupt ID that are pending on that interface have the same priority and must be handled serially. How the CPU interface serializes these SGIs is IMPLEMENTATION DEFINED.

9.4.2.2 Implications of the 1-N Model

In a multiprocessor implementation, GIC uses a 1-N model to handle peripheral interrupts that target multiple processors. This means that when GIC recognizes an interrupt acknowledgement from one of the target processors, it clears the pending state of the interrupt on all the other targeted processors. This model means the first available processor can handle this interrupt. However, the interrupt might generate an interrupt exception on multiple targeted processors. For example, two of the targeted processors recognize the interrupt exception request from the GIC at similar times.

When multiple target processors attempt to acknowledge the interrupt:

- A processor reads the ICCIAR and obtains the interrupt ID of the interrupt to be serviced. Refer to Interrupt Acknowledge Register (ICCIAR) for more information. Multiple target processors might have obtained this interrupt ID if the processors read their ICCIARs at very similar times. The system Interrupt Handling and Prioritization Unrestricted Access Non-Confidential requires software on the target processors to ensure that only one processor runs its interrupt service routine. To achieve this, implement a lock on the interrupt service routine (ISR) in shared memory. This operates as follows:
  - Each target processor that obtains the interrupt ID from its read of the ICCIAR runs a semaphore routine. It does this by attempting to obtain a lock on the ISR corresponding to the specified ID value.
  - If a processor fails to obtain the lock, it does no further processing of the interrupt. However, the processor writes the interrupt ID to its ICCEOIR. Refer to End of Interrupt Register (ICCEOIR) for more information.
  - The processor that obtains the lock handles the interrupt and then writes the interrupt ID to its ICCEOIR.
- A processor reads the ICCIAR and obtains the interrupt ID 1023 by indicating a spurious interrupt. The processor can return from its interrupt service routine without writing to its ICCEOIR. The spurious interrupt ID indicates that the original interrupt is no longer pending. This indication is typically because of another target processor that is handling it.

For any processor when an interrupt is active and pending, GIC does not signal an interrupt exception request for this interrupt to any processor until it clears the active status.

NOTE: A GIC implementation t ensures that only one processor can make a 1-N interrupt active by removing the need for a lock on the ISR. This is not required by the architecture, and generic GIC code should not rely on this behaviour.

9.4.2.3 Interrupt Handling State Machine

The Distributor maintains a state machine for each supported interrupt on each CPU interface.

Figure 9-5 illustrates interrupt handling state machine and the possible state transitions.

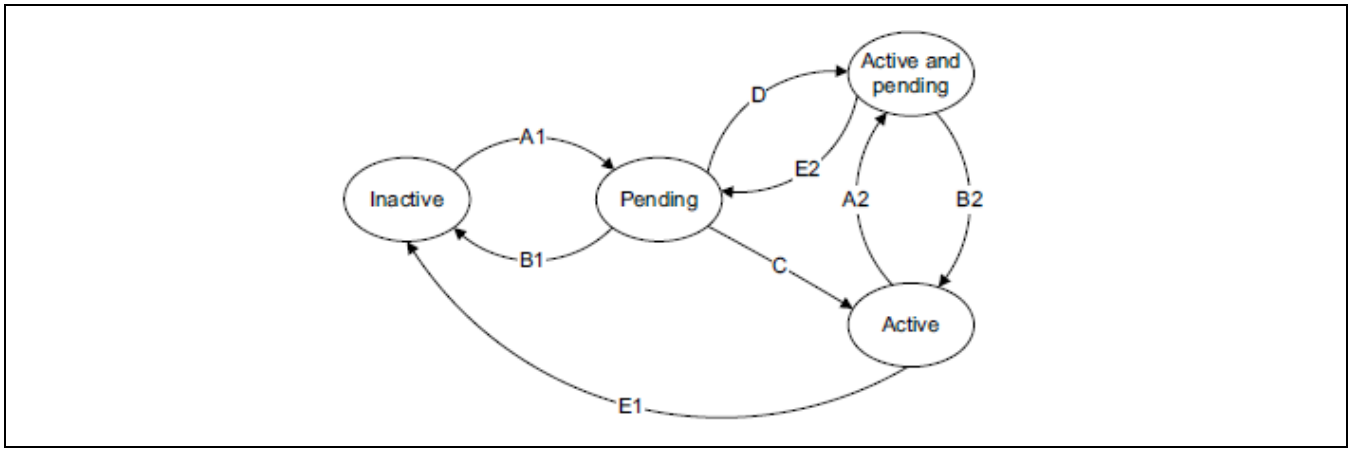


Figure 9-5 Interrupt Handling State Machine

When you enable the Distributor and CPU interfaces, the conditions that cause each of the state transitions are as follows:

Transition A1 or A2, Add Pending Status

For an SGI:

- Transition occurs on a write to an ICDSGIR that specifies the processor as a target.
- Transition occurs only if the security configuration of the specified SGI, for the appropriate CPU interface, corresponds to the ICDSGIR.SATT bit value. This happens if the GIC implements the Secure Extensions and the write to the ICDSGIR in Secure.

For an SPI or PPI, transition occurs if either:

- A peripheral asserts an interrupt signal or
- Software writes to an ICDISPR.

Transition B1 or B2, Remove Pending Status

Transition not applicable to SGIs:

- A pending SGI should transition through the active state or reset to remove its pending status.
- An active and pending SGI should transition through the pending state or reset to remove its pending status.

For an SPI or PPI, transition occurs if either:

- The level-sensitive interrupt is pending only because of the assertion of an input signal, and that signal is de-asserted or
- The interrupt is pending because of the assertion of an edge-triggered interrupt signal, or a write to an ICDISPR. The software then writes to the corresponding ICDICPR.

Transition C

If the interrupt is enabled and of sufficient priority to be signalled to the processor, transition occurs when software reads from the ICCIAR.

Transition D

For an SGI, transition occurs if the associated SGI is enabled and the Distributor forwards it to the CPU interface at the same time that the processor reads the ICCIAR to acknowledge a previous instance of the SGI. Whether this transition occurs, depends on the timing of the read of the ICCIAR relative to the reforwarding of the SGI.

For an SPI or PPI:

- Transition occurs when:
  - The interrupt is enabled.
  - Software reads from the ICCIAR. This read adds the active state to the interrupt.
  - Interrupt signal remains asserted for a level-sensitive interrupt. This is because the peripheral does not dissert the interrupt until the processor has serviced the interrupt.
- For an edge-triggered interrupt, whether this transition occurs depends on the timing of the read of the ICCIAR relative to the detection of the reassertion of the interrupt. Otherwise the read of the ICCIAR causes transition C, possibly followed by transition A2.

Transition E1 or E2, Remove Active Status

Transition occurs when software writes to the ICCEOIR.

9.4.2.4 Special Interrupt Numbers

The GIC architecture reserves interrupt ID numbers 1020-1023 for special purposes. In a GIC that does not implement the Security Extensions; the only ID number used is ID 1023. This value is returned to a processor, in response to an interrupt acknowledge, if there is no pending interrupt with sufficient priority for it to be signalled to the processor, it is described as a response to a spurious interrupt.

9.4.3 Interrupt Prioritization

This sub-section includes:

- Preemption
- Priority masking
- Priority grouping

Software configures interrupt prioritization in GIC by assigning a priority value to each interrupt source. Priority values are 8-bit unsigned binary. A GIC supports a minimum of 16 and a maximum of 256 priority levels.

Table 9-5 describes the effect of not implementing some LS priority field bit.

Table 9-5 Effect of not Implementing Some LS Priority Field Bit

Implemented Priority Bits	Possible Priority Field Values	Number of Priority Levels
[7:0]	0x00-0xFF (0-255), all values	256
[7:1]	0x00-0xFE (0-254), even values only	128
[7:2]	0x00-0xFC (0-252), in steps of 4	64
[7:3]	0x00-0xF8 (0-248), in steps of 8	32
[7:4]	0x00-0xF0 (0-240), in steps of 16	16

In the GIC prioritization scheme, lower numbers have higher priority. That means, the lower the assigned priority value, the higher is the priority of the interrupt. The highest interrupt priority always has priority field value 0, and the lowest value depends on the number of implemented priority levels as described in *Table 9-1* and *Table 9-5*.

The ICDIPRs hold the priority value for each supported interrupt. Refer to Interrupt Priority Registers (ICDIPRn) for more information. To determine the number of priority bits implemented, write 0xFF to an ICDIPR priority field and read the stored value.

NOTE: ARM recommends that, before checking the priority range in this way

1. For a peripheral interrupt, software first disables the interrupt.
2. For an SGI, software first verifies that the interrupt is inactive.

An implementation might reserve an interrupt for a particular purpose and assign a fixed priority to that interrupt. This means that the priority value for that interrupt is read-only.

This model aligns with the priority grouping mechanism as described in Priority grouping.

When an interrupt is active on a CPU interface, GIC signals a higher-priority interrupt on that CPU interface, Refer to Preemption section for more information.

Software sets the priority of each interrupt in the appropriate ICDIPR. Refer to Interrupt Priority Registers (ICDIPRn) for more information.

9.4.3.1 Preemption

A CPU interface supports forwarding of higher priority pending interrupts to a target processor before an active interrupt completes. A pending interrupt is only forwarded if it has a higher priority than all of:

- The priority of the highest priority active interrupt on the target processor, the running priority for the processor, Refer to Running Priority Register (ICCRPR)
- The priority mask. Refer to Priority Masking sub-section.
- The priority group. Refer to Priority Grouping sub-section.

Preemption occurs when the processor acknowledges the new interrupt, and starts to service by preferring to the previously active interrupt or the currently running process. When this occurs, the initial active interrupt is said to have been preempted.

Interrupt nesting is sometimes described as starting to service an interrupt while another interrupt is still active.

9.4.3.2 Priority Masking

ICCPMR for a CPU interface defines a priority threshold for the target processor. Refer Interrupt Priority Mask Register (ICCPMR) for more information. GIC only signals pending interrupts with a higher priority than this threshold value to the target processor. A value of zero, which is the register reset value, masks all interrupts to the associated processor.

GIC always masks an interrupt that has the largest supported priority field value. This provides an additional means of preventing an interrupt that is being signaled to any processor.

9.4.3.3 Priority Grouping

Priority grouping splits each priority value into two fields:

- Group priority
- Sub-priority

GIC uses the group priority field to determine whether a pending interrupt has sufficient priority to preempt a currently active interrupt.

The binary point field in the ICCBPR controls the split of the priority bits into two parts. This 3-bit field specifies how many of the least significant bits of the 8-bit interrupt priority field are excluded from the group priority field.

Table 9-6 describes the priority grouping by binary point.

Table 9-6 Priority Grouping by Binary Point

Binary Point Value	Group Priority Field	Sub Priority Field	Field with Binary Point
0	[7:1]	0	ggggggg.s
1	[7:2]	[1:0]	gggggg.ss
2	[7:3]	[2:0]	ggggg.sss
3	[7:4]	[3:0]	gggg.ssss
4	[7:5]	[4:0]	ggg.sssss
5	[7:6]	[5:0]	gg.ssssss
6	[7]	[6:0]	g.sssssss
7	No preemption	[7:0]	.ssssssss

Refer to Binary Point Register (ICCBPR) for more information about the ICCBPR.

9.4.4 The Effect of the Security Extensions on Interrupt Handling

If a GIC CPU interface implements the Security Extensions, it provides two interrupt output signals:

- IRQ
- FIQ

The CPU interface always uses the IRQ exception request for Non-secure interrupts. Software can configure the CPU interface to use either IRQ or FIQ exception requests for secure interrupts.

9.4.4.1 Security Extensions Support

Software detects support for the Security Extensions by reading the ICDICTR. SecurityExtn bit Refer to Interrupt Controller Type Register (ICDICTR).

Secure software enables secure writes to the ICDISRs to configure each interrupt as Secure or Non-secure. Refer to Interrupt Security Registers (ICDISRn) for more information.

In addition:

- The banking of registers provides independent control of Secure and Non-secure interrupts.
- The Secure copy of the ICCICR has additional fields to control the processing of Secure and Non-secure interrupts. Refer to CPU Interface Control Register (ICCICR) for more information.

  These fields are:
  - **The SBPR bit:** Affects the preemption of Non-secure interrupts. Refer to Control of preemption by Non-secure interrupts for more information.
  - **The FIQEn bit:** Controls whether the interface signals Secure interrupts to the processor by using the IRQ or FIQ interrupt exception requests.
  - **The AckCtl bit:** Affects the acknowledgment of Non-secure interrupts. Refer to Effect of the Security Extensions on interrupt acknowledgement.
  - **The EnableNS bit:** that controls whether Non-secure interrupts are signaled to the processor, and is an alias of the Enable bit in the Non-secure ICCICR.
- The Non-secure copy of the ICCBPR is aliased as the ICCABPR, Refer to Aliased Binary Point Register (ICCABPR) for more information. This is a secure register, which is only accessible by secure accesses.

9.4.4.2 Special Interrupt Numbers when the Security Extensions are Implemented

Special interrupt numbers on page 3-11 describes the use of interrupt ID 1023 to indicate a spurious interrupt. The complete list of the interrupt ID numbers that the GIC architecture reserves for special purposes are:

- **1020-1021:** Reserved.
- **1022:** Used only if GIC implements the Security Extensions

GIC returns this value to a processor in response to an interrupt acknowledgement only when all of the following apply:

- The interrupt acknowledge is a Secure read
- The highest priority pending interrupt is Non-secure
- The AckCtl bit in the Secure ICCICR is set to 0
- The priority of the interrupt is sufficient for it to be signalled to the processor.

1023: This value is returned to a processor, in response to an interrupt acknowledge, if there is no pending interrupt with sufficient priority for it to be signalled to the processor.

The Secure software treats values of 1022 and 1023 as spurious interrupts on a processor that implements the Security Extensions.

9.4.4.3 Effect of the Security Extensions on Interrupt Acknowledgement

When a processor takes an interrupt, it acknowledges the interrupt by reading the ICCIAR. Refer to General handling of interrupts for more information. A Read of the ICCIAR always acknowledges the highest priority pending interrupt for the processor that performs the read.

If the highest priority pending interrupt is a secure interrupt, the processor must make a secure read of the ICCIAR to acknowledge it.

By default, the processor must make a Non-secure read of the ICCIAR to acknowledge a Non-secure interrupt. When the AckCtl bit in the Secure ICCICR is set to "1", the processor can make a secure read of the ICCIAR to acknowledge a Non-secure interrupt.

When the read of the ICCIAR does not match the security of the interrupt by taking into account the AckCtl bit value for a Non-secure interrupt, the ICCIAR read does not acknowledge any interrupt and returns the value:

- 1022 for a Secure read when the highest priority interrupt is Non-secure
- 1023 for a Non-secure read when the highest priority interrupt is Secure.

Refer to Effect of the Security Extensions on reads of the ICCIAR for more information.

9.4.5 The Effect of the Security Extensions on Interrupt Prioritization

When GIC supports the Security Extensions:

- Secure software must program the ICDISRs to configure each supported interrupt as either secure or Non-secure. Refer to Interrupt Security Registers (ICDISRn) for more information.
- GIC provides secure and Non-secure views of the interrupt priority settings.
- The minimum number of supported priority values increases from 16 to 32.

9.5 Register Description

9.5.1 Register Map Summary

- Base Address: 0x1048_0000

Register	Offset	Description	Reset Value
ICCICR_CPU0	0x0000	CPU interface control register	0x0000_0000
ICCPMR_CPU0	0x0004	Interrupt priority mask register	0x0000_0000
ICCBPR_CPU0	0x0008	Binary point register	0x0000_0000
ICCIAR_CPU0	0x000C	Interrupt acknowledge register	0x0000_03FF
ICCEOIR_CPU0	0x0010	End of interrupt register	Undefined
ICCRPR_CPU0	0x0014	Running priority register	0x0000_00FF
ICCHPIR_CPU0	0x0018	Highest pending interrupt register	0x0000_03FF
ICCABPR_CPU0	0x001C	Aliased binary point register	0x0000_0000
INTEG_EN_C_CPU0	0x0040	Integration test enable register	0x0000_0000
INTERRUPT_ OUT_CPU0	0x0044	Interrupt output register	0x0000_0000
ICCIIDR	0x00FC	CPU interface identification register	0x3901_043B
ICCICR_CPU1	0x4000	CPU interface control register	0x0000_0000
ICCPMR_CPU1	0x4004	Interrupt priority mask register	0x0000_0000
ICCBPR_CPU1	0x4008	Binary point register	0x0000_0000
ICCIAR_CPU1	0x400C	Interrupt acknowledge register	0x0000_03FF
ICCEOIR_CPU1	0x4010	End of interrupt register	Undefined
ICCRPR_CPU1	0x4014	Running priority register	0x0000_00FF
ICCHPIR_CPU1	0x4018	Highest pending interrupt register	0x0000_03FF
ICCABPR_CPU1	0x401C	Aliased binary point register	0x0000_0000
INTEG_C_EN_CPU1	0x4040	Integration test enable register	0x0000_0000
INTERRUPT_ OUT_CPU1	0x4044	Interrupt output register	0x0000_0000
ICCICR_CPU2	0x8000	CPU interface control register	0x0000_0000
ICCPMR_CPU2	0x8004	Interrupt priority mask register	0x0000_0000
ICCBPR_CPU2	0x8008	Binary point register	0x0000_0000
ICCIAR_CPU2	0x800C	Interrupt acknowledge register	0x0000_03FF
ICCEOIR_CPU2	0x8010	End of interrupt register	Undefined
ICCRPR_CPU2	0x8014	Running priority register	0x0000_00FF
ICCHPIR_CPU2	0x8018	Highest pending interrupt register	0x0000_03FF
ICCABPR_CPU2	0x801C	Aliased binary point register	0x0000_0000
INTEG_C_EN_CPU2	0x8040	Integration test enable register	0x0000_0000
INTERRUPT_ OUT_CPU2	0x8044	Interrupt output register	0x0000_0000

Register	Offset	Description	Reset Value
ICCICR_CPU3	0xC000	CPU interface control register	0x0000_0000
ICCPMR_CPU3	0xC004	Interrupt priority mask register	0x0000_0000
ICCBPR_CPU3	0xC008	Binary point register	0x0000_0000
ICCIAR_CPU3	0xC00C	Interrupt acknowledge register	0x0000_03FF
ICCEOIR_CPU3	0xC010	End of interrupt register	Undefined
ICCRPR_CPU3	0xC014	Running priority register	0x0000_00FF
ICCHPIR_CPU3	0xC018	Highest pending interrupt register	0x0000_03FF
ICCABPR_CPU3	0xC01C	Aliased binary point register	0x0000_0000
INTEG_C_EN_CPU3	0xC040	Integration test enable register	0x0000_0000
INTERRUPT_ OUT_CPU3	0xC044	Interrupt output register	0x0000_0000

- Base Address: 0x1049_0000

Register	Offset	Description	Reset Value
ICDDCR	0x0000	Distributor control register	0x0000_0000
ICDICTR	0x0004	Interrupt controller type register	0x0000_FC24
ICDIIDR	0x0008	Distributor implementer identification register	0x0000_043B
ICDISR0_CPU0	0x0080	Interrupt security registers (SGI,PPI)	0x0000_0000
ICDISR1	0x0084	Interrupt security registers (SPI[31:0])	0x0000_0000
ICDISR2	0x0088	Interrupt security registers (SPI[63:32])	0x0000_0000
ICDISR3	0x008C	Interrupt security registers (SPI[95:64])	0x0000_0000
ICDISR4	0x0090	Interrupt security registers (SPI[127:96])	0x0000_0000
ICDISER0_CPU0	0x0100	Interrupt set-enable register (SGI,PPI)	0x0000_FFFF
ICDISER1	0x0104	Interrupt set-enable register (SPI[31:0])	0x0000_0000
ICDISER2	0x0108	Interrupt set-enable register (SPI[63:32])	0x0000_0000
ICDISER3	0x010C	Interrupt set-enable register (SPI[95:64])	0x0000_0000
ICDISER4	0x0110	Interrupt set-enable register (SPI[127:96])	0x0000_0000
ICDICER0_CPU0	0x0180	Interrupt clear-enable register (SGI,PPI)	0x0000_FFFF
ICDICER1	0x0184	Interrupt clear-enable register (SPI[31:0])	0x0000_0000
ICDICER2	0x0188	Interrupt clear-enable register (SPI[63:32])	0x0000_0000
ICDICER3	0x018C	Interrupt clear-enable register (SPI[95:64])	0x0000_0000
ICDICER4	0x0190	Interrupt clear-enable register (SPI[127:96])	0x0000_0000
ICDISPR0_CPU0	0x0200	Interrupt pending-set register (SGI,PPI)	0x0000_0000
ICDISPR1	0x0204	Interrupt pending-set register (SPI[31:0])	0x0000_0000
ICDISPR2	0x0208	Interrupt pending-set register (SPI[63:32])	0x0000_0000
ICDISPR3	0x020C	Interrupt pending-set register (SPI[95:64])	0x0000_0000
ICDISPR4	0x0210	Interrupt pending-set register (SPI[127:96])	0x0000_0000
ICDICPR0_CPU0	0x0280	Interrupt pending-clear register (SGI,PPI)	0x0000_0000
ICDICPR1	0x0284	Interrupt pending-clear register (SPI[31:0])	0x0000_0000
ICDICPR2	0x0288	Interrupt pending-clear register (SPI[63:32])	0x0000_0000
ICDICPR3	0x028C	Interrupt pending-clear register (SPI[95:64])	0x0000_0000
ICDICPR4	0x0290	Interrupt pending-clear register(SPI[127:96])	0x0000_0000
ICDABR0_CPU0	0x0300	Active bit register (SGI, PPI)	0x0000_0000
ICDABR1	0x0304	Active bit register (SPI[31:0])	0x0000_0000
ICDABR2	0x0308	Active bit register (SPI[63:32])	0x0000_0000
ICDABR3	0x030C	Active bit register (SPI[95:64])	0x0000_0000
ICDABR4	0x0310	Active bit register (SPI[127:96])	0x0000_0000
ICDIPR0_CPU0	0x0400	Priority level register (SGI[3:0])	0x0000_0000
ICDIPR1_CPU0	0x0404	Priority level register (SGI[7:4])	0x0000_0000
ICDIPR2_CPU0	0x0408	Priority level register (SGI[11:8])	0x0000_0000

Register	Offset	Description	Reset Value
ICDIPR3_CPU0	0x040C	Priority level register (SGI[15:12])	0x0000_0000
ICDIPR4_CPU0	0x0410	Priority level register (PPI[3:0])	0x0000_0000
ICDIPR5_CPU0	0x0414	Priority level register (PPI[7:4])	0x0000_0000
ICDIPR6_CPU0	0x0418	Priority level register (PPI[11:8])	0x0000_0000
ICDIPR7_CPU0	0x041C	Priority level register (PPI[15:12])	0x0000_0000
ICDIPR8	0x0420	Priority level register (SPI[3:0])	0x0000_0000
ICDIPR9	0x0424	Priority level register (SPI[7:4])	0x0000_0000
ICDIPR10	0x0428	Priority level register (SPI[11:8])	0x0000_0000
ICDIPR11	0x042C	Priority level register (SPI[15:12])	0x0000_0000
ICDIPR12	0x0430	Priority level register (SPI[19:16])	0x0000_0000
ICDIPR13	0x0434	Priority level register (SPI[23:20])	0x0000_0000
ICDIPR14	0x0438	Priority level register (SPI[27:24])	0x0000_0000
ICDIPR15	0x043C	Priority level register (SPI[31:28])	0x0000_0000
ICDIPR16	0x0440	Priority level register (SPI[35:32])	0x0000_0000
ICDIPR17	0x0444	Priority level register (SPI[39:36])	0x0000_0000
ICDIPR18	0x0448	Priority level register (SPI[43:40])	0x0000_0000
ICDIPR19	0x044C	Priority level register (SPI[47:44])	0x0000_0000
ICDIPR20	0x0450	Priority level register (SPI[51:48])	0x0000_0000
ICDIPR21	0x0454	Priority level register (SPI[55:52])	0x0000_0000
ICDIPR22	0x0458	Priority level register (SPI[59:56])	0x0000_0000
ICDIPR23	0x045C	Priority level register (SPI[63:60])	0x0000_0000
ICDIPR24	0x0460	Priority level register (SPI[67:64])	0x0000_0000
ICDIPR25	0x0464	Priority level register (SPI[71:68])	0x0000_0000
ICDIPR26	0x0468	Priority level register (SPI[75:72])	0x0000_0000
ICDIPR27	0x046C	Priority level register (SPI[79:76])	0x0000_0000
ICDIPR28	0x0470	Priority level register (SPI[83:80])	0x0000_0000
ICDIPR29	0x0474	Priority level register (SPI[87:84])	0x0000_0000
ICDIPR30	0x0478	Priority level register (SPI[91:98])	0x0000_0000
ICDIPR31	0x047C	Priority level register (SPI[95:92])	0x0000_0000
ICDIPR32	0x0480	Priority level register (SPI[99:96])	0x0000_0000
ICDIPR33	0x0484	Priority level register (SPI[103:100])	0x0000_0000
ICDIPR34	0x0488	Priority level register (SPI[107:104])	0x0000_0000
ICDIPR35	0x048C	Priority level register (SPI[111:108])	0x0000_0000
ICDIPR36	0x0490	Priority level register (SPI[115:112])	0x0000_0000
ICDIPR37	0x0494	Priority level register (SPI[119:116])	0x0000_0000
ICDIPR38	0x0498	Priority level register (SPI[123:120])	0x0000_0000
ICDIPR39	0x049C	Priority level register (SPI[127:124])	0x0000_0000

Register	Offset	Description	Reset Value
ICDIPTR0_CPU0	0x0800	Processor targets register (SGI[3:0])	0x0101_0101
ICDIPTR1_CPU0	0x0804	Processor targets register (SGI[7:4])	0x0101_0101
ICDIPTR2_CPU0	0x0808	Processor targets register (SGI[11:8])	0x0101_0101
ICDIPTR3_CPU0	0x080C	Processor targets register (SGI[15:12])	0x0101_0101
ICDIPTR4_CPU0	0x0810	Processor targets register (PPI[3:0])	0x0101_0101
ICDIPTR5_CPU0	0x0814	Processor targets register (PPI[7:4])	0x0101_0101
ICDIPTR6_CPU0	0x0818	Processor targets register (PPI[11:8])	0x0101_0101
ICDIPTR7_CPU0	0x081C	Processor targets register (PPI[15:12])	0x0101_0101
ICDIPTR8	0x0820	Processor targets register (SPI[3:0])	0x0000_0000
ICDIPTR9	0x0824	Processor targets register (SPI[7:4])	0x0000_0000
ICDIPTR10	0x0828	Processor targets register (SPI[11:8])	0x0000_0000
ICDIPTR11	0x082C	Processor targets register (SPI[15:12])	0x0000_0000
ICDIPTR12	0x0830	Processor targets register (SPI[19:16])	0x0000_0000
ICDIPTR13	0x0834	Processor targets register (SPI[23:20])	0x0000_0000
ICDIPTR14	0x0838	Processor targets register (SPI[27:24])	0x0000_0000
ICDIPTR15	0x083C	Processor targets register (SPI[31:28])	0x0000_0000
ICDIPTR16	0x0840	Processor targets register (SPI[35:32])	0x0000_0000
ICDIPTR17	0x0844	Processor targets register (SPI[39:36])	0x0000_0000
ICDIPTR18	0x0848	Processor targets register (SPI[43:40])	0x0000_0000
ICDIPTR19	0x084C	Processor targets register (SPI[47:44])	0x0000_0000
ICDIPTR20	0x0850	Processor targets register (SPI[51:48])	0x0000_0000
ICDIPTR21	0x0854	Processor targets register (SPI[55:52])	0x0000_0000
ICDIPTR22	0x0858	Processor targets register (SPI[59:56])	0x0000_0000
ICDIPTR23	0x085C	Processor targets register (SPI[63:60])	0x0000_0000
ICDIPTR24	0x0860	Processor targets register (SPI[67:64])	0x0000_0000
ICDIPTR25	0x0864	Processor targets register (SPI[71:68])	0x0000_0000
ICDIPTR26	0x0868	Processor targets register (SPI[75:72])	0x0000_0000
ICDIPTR27	0x086C	Processor targets register (SPI[79:76])	0x0000_0000
ICDIPTR28	0x0870	Processor targets register (SPI[83:80])	0x0000_0000
ICDIPTR29	0x0874	Processor targets register (SPI[87:84])	0x0000_0000
ICDIPTR30	0x0878	Processor targets register (SPI[91:98])	0x0000_0000
ICDIPTR31	0x087C	Processor targets register (SPI[95:92])	0x0000_0000
ICDIPTR32	0x0880	Processor targets register (SPI[99:96])	0x0000_0000
ICDIPTR33	0x0884	Processor targets register (SPI[103:100])	0x0000_0000
ICDIPTR34	0x0888	Processor targets register (SPI[107:104])	0x0000_0000
ICDIPTR35	0x088C	Processor targets register (SPI[111:108])	0x0000_0000
ICDIPTR36	0x0890	Processor targets register (SPI[115:112])	0x0000_0000

Register	Offset	Description	Reset Value
ICDIPTR37	0x0894	Processor targets register (SPI[119:116])	0x0000_0000
ICDIPTR38	0x0898	Processor targets register (SPI[123:120])	0x0000_0000
ICDIPTR39	0x089C	Processor targets register (SPI[127:124])	0x0000_0000
ICDICFR0_CPU0	0x0C00	Interrupt configuration register (SGI[15:0])	0xAAAA_AAAA
ICDICFR1_CPU0	0x0C04	Interrupt configuration register (PPI[15:0])	0x7DD5_5FFF
ICDICFR2	0x0C08	Interrupt configuration register (SPI[15:0])	0x5555_5555
ICDICFR3	0x0C0C	Interrupt configuration register (SPI[31:16])	0x5555_5555
ICDICFR4	0x0C10	Interrupt configuration register (SPI[47:32])	0x5555_5555
ICDICFR5	0x0C14	Interrupt configuration register (SPI[63:48])	0x5555_5555
ICDICFR6	0x0C18	Interrupt configuration register (SPI[79:64])	0x5555_5555
ICDICFR7	0x0C1C	Interrupt configuration register (SPI[95:80])	0x5555_5555
ICDICFR8	0x0C20	Interrupt configuration register (SPI[111:95])	0x5555_5555
ICDICFR9	0x0C24	Interrupt configuration register (SPI[127:112])	0x5555_5555
PPI_STATUS_CPU0	0x0D00	PPI status register	0x0000_0000
SPI_STATUS0	0x0D04	SPI[31:0] status register	0x0000_0000
SPI_STATUS1	0x0D08	SPI[63:32] status register	0x0000_0000
SPI_STATUS2	0x0D0C	SPI[95:64] status register	0x0000_0000
SPI_STATUS3	0x0D10	SPI[127:96] status register	0x0000_0000
ICDSGIR	0x0F00	Software generated interrupt register	Undefined
ICDISR0_CPU1	0x4080	Interrupt security registers (SGI,PPI)	0x0000_0000
ICDISER0_CPU1	0x4100	Interrupt set-enable register (SGI,PPI)	0x0000_FFFF
ICDICER0_CPU1	0x4180	Interrupt clear-enable register (SGI,PPI)	0x0000_FFFF
ICDISPR0_CPU1	0x4200	Interrupt pending-set register (SGI,PPI)	0x0000_0000
ICDICPR0_CPU1	0x4280	Interrupt pending-clear register (SGI,PPI)	0x0000_0000
ICDABR0_CPU1	0x4300	Active status register (SGI, PPI)	0x0000_0000
ICDIPR0_CPU1	0x4400	Priority level register (SGI[3:0])	0x0000_0000
ICDIPR1_CPU1	0x4404	Priority level register (SGI[7:4])	0x0000_0000
ICDIPR2_CPU1	0x4408	Priority level register (SGI[11:8])	0x0000_0000
ICDIPR3_CPU1	0x440C	Priority level register (SGI[15:12])	0x0000_0000
ICDIPR4_CPU1	0x4410	Priority level register (PPI[3:0])	0x0000_0000
ICDIPR5_CPU1	0x4414	Priority level register (PPI[7:4])	0x0000_0000
ICDIPR6_CPU1	0x4418	Priority level register (PPI[11:8])	0x0000_0000
ICDIPR7_CPU1	0x441C	Priority level register (PPI[15:12])	0x0000_0000
ICDIPTR0_CPU1	0x4800	Processor targets register (SGI[3:0])	0x0202_0202
ICDIPTR1_CPU1	0x4804	Processor targets register (SGI[7:4])	0x0202_0202
ICDIPTR2_CPU1	0x4808	Processor targets register (SGI[11:8])	0x0202_0202
ICDIPTR3_CPU1	0x480C	Processor targets register (SGI[15:12])	0x0202_0202

Register	Offset	Description	Reset Value
ICDIPTR4_CPU1	0x4810	Processor targets register (PPI[3:0])	0x0202_0202
ICDIPTR5_CPU1	0x4814	Processor targets register (PPI[7:4])	0x0202_0202
ICDIPTR6_CPU1	0x4818	Processor targets register (PPI[11:8])	0x0202_0202
ICDIPTR7_CPU1	0x481C	Processor targets register (PPI[15:12])	0x0202_0202
ICDICFR0_CPU1	0x4C00	Interrupt configuration register (SGI[15:0])	0xAAAA_AAAA
ICDICFR1_CPU1	0x4C04	Interrupt configuration register (PPI[15:0])	0x7DD5_5FFF
PPI_STATUS_CPU1	0x4D00	PPI status register	0x0000_0000
ICDISR0_CPU2	0x8080	Interrupt security registers (SGI,PPI)	0x0000_0000
ICDISER0_CPU2	0x8100	Interrupt set-enable register (SGI,PPI)	0x0000_FFFF
ICDICER0_CPU2	0x8180	Interrupt clear-enable register (SGI,PPI)	0x0000_FFFF
ICDISPR0_CPU2	0x8200	Interrupt pending-set register (SGI,PPI)	0x0000_0000
ICDICPR0_CPU2	0x8280	Interrupt pending-clear register (SGI,PPI)	0x0000_0000
ICDABR0_CPU2	0x8300	Active status register (SGI, PPI)	0x0000_0000
ICDIPR0_CPU2	0x8400	Priority level register (SGI[3:0])	0x0000_0000
ICDIPR1_CPU2	0x8404	Priority level register (SGI[7:4])	0x0000_0000
ICDIPR2_CPU2	0x8408	Priority level register (SGI[11:8])	0x0000_0000
ICDIPR3_CPU2	0x840C	Priority level register (SGI[15:12])	0x0000_0000
ICDIPR4_CPU2	0x8410	Priority level register (PPI[3:0])	0x0000_0000
ICDIPR5_CPU2	0x8414	Priority level register (PPI[7:4])	0x0000_0000
ICDIPR6_CPU2	0x8418	Priority level register (PPI[11:8])	0x0000_0000
ICDIPR7_CPU2	0x841C	Priority level register (PPI[15:12])	0x0000_0000
ICDIPTR0_CPU2	0x8800	Processor targets register (SGI[3:0])	0x0202_0202
ICDIPTR1_CPU2	0x8804	Processor targets register (SGI[7:4])	0x0202_0202
ICDIPTR2_CPU2	0x8808	Processor targets register (SGI[11:8])	0x0202_0202
ICDIPTR3_CPU2	0x880C	Processor targets register (SGI[15:12])	0x0202_0202
ICDIPTR4_CPU2	0x8810	Processor targets register (PPI[3:0])	0x0202_0202
ICDIPTR5_CPU2	0x8814	Processor targets register (PPI[7:4])	0x0202_0202
ICDIPTR6_CPU2	0x8818	Processor targets register (PPI[11:8])	0x0202_0202
ICDIPTR7_CPU2	0x881C	Processor targets register (PPI[15:12])	0x0202_0202
ICDICFR0_CPU2	0x8C00	Interrupt configuration register (SGI[15:0])	0xAAAA_AAAA
ICDICFR1_CPU2	0x8C04	Interrupt configuration register (PPI[15:0])	0x7DD5_5FFF
PPI_STATUS_CPU2	0x8D00	PPI status register	0x0000_0000
ICDISR0_CPU3	0xC080	Interrupt security registers (SGI,PPI)	0x0000_0000
ICDISER0_CPU3	0xC100	Interrupt set-enable register (SGI,PPI)	0x0000_FFFF
ICDICER0_CPU3	0xC180	Interrupt clear-enable register (SGI,PPI)	0x0000_FFFF
ICDISPR0_CPU3	0xC200	Interrupt pending-set register (SGI,PPI)	0x0000_0000
ICDICPR0_CPU3	0xC280	Interrupt pending-clear register (SGI,PPI)	0x0000_0000

Register	Offset	Description	Reset Value
ICDABR0_CPU3	0xC300	Active status register (SGI, PPI)	0x0000_0000
ICDIPR0_CPU3	0xC400	Priority level register (SGI[3:0])	0x0000_0000
ICDIPR1_CPU3	0xC404	Priority level register (SGI[7:4])	0x0000_0000
ICDIPR2_CPU3	0xC408	Priority level register (SGI[11:8])	0x0000_0000
ICDIPR3_CPU3	0xC40C	Priority level register (SGI[15:12])	0x0000_0000
ICDIPR4_CPU3	0xC410	Priority level register (PPI[3:0])	0x0000_0000
ICDIPR5_CPU3	0xC414	Priority level register (PPI[7:4])	0x0000_0000
ICDIPR6_CPU3	0xC418	Priority level register (PPI[11:8])	0x0000_0000
ICDIPR7_CPU3	0xC41C	Priority level register (PPI[15:12])	0x0000_0000
ICDIPTR0_CPU3	0xC800	Processor targets register (SGI[3:0])	0x0202_0202
ICDIPTR1_CPU3	0xC804	Processor targets register (SGI[7:4])	0x0202_0202
ICDIPTR2_CPU3	0xC808	Processor targets register (SGI[11:8])	0x0202_0202
ICDIPTR3_CPU3	0xC80C	Processor targets register (SGI[15:12])	0x0202_0202
ICDIPTR4_CPU3	0xC810	Processor targets register (PPI[3:0])	0x0202_0202
ICDIPTR5_CPU3	0xC814	Processor targets register (PPI[7:4])	0x0202_0202
ICDIPTR6_CPU3	0xC818	Processor targets register (PPI[11:8])	0x0202_0202
ICDIPTR7_CPU3	0xC81C	Processor targets register (PPI[15:12])	0x0202_0202
ICDICFR0_CPU3	0xCC00	Interrupt configuration register (SGI[15:0])	0xAAAA_AAAA
ICDICFR1_CPU3	0xCC04	Interrupt configuration register (PPI[15:0])	0x7DD5_5FFF
PPI_STATUS_CPU3	0xCD00	PPI status register	0x0000_0000

9.5.1.1 ICCICR_CPUn

- Base Address: 0x1048_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000 (ICCICR_CPU0)
- Address = Base Address + 0x4000, Reset Value = 0x0000_0000 (ICCICR_CPU1)
- Address = Base Address + 0x8000, Reset Value = 0x0000_0000 (ICCICR_CPU2)
- Address = Base Address + 0xC000, Reset Value = 0x0000_0000 (ICCICR_CPU3)

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
Enable	[0]	RW	Global enable for signaling of interrupts by the CPU Interface to the connected processors. 0 = Disables signaling of interrupts 1 = Enables signaling of interrupts	0x0

It enables the signaling of interrupts to the target processors. In a GIC that implements the Security Extensions, this register provides additional global controls for handling Secure interrupts. This register is banked to provide Secure and Non-secure copies.

9.5.1.2 ICCPMR_CPUn

- Base Address: 0x1048_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000 (ICCPMR_CPU0)
- Address = Base Address + 0x4004, Reset Value = 0x0000_0000 (ICCPMR_CPU1)
- Address = Base Address + 0x8004, Reset Value = 0x0000_0000 (ICCPMR_CPU2)
- Address = Base Address + 0xC004, Reset Value = 0x0000_0000 (ICCPMR_CPU3)

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x0
Priority	[7:0]	RW	The priority mask level for the CPU0 interface When the priority of an interrupt is higher than the value that this field indicates, the interface signals the interrupt to the processor. 256 priority levels support 0x00 – 0xFF (0 to 255), all values	0x0

This register provides an interrupt priority filter. Only interrupts with higher priority than the value in this register can be signaled to the processor.

9.5.1.3 ICCBPR_CPUn

- Base Address: 0x1048_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000 (ICCBPR_CPU0)
- Address = Base Address + 0x4008, Reset Value = 0x0000_0000 (ICCBPR_CPU1)
- Address = Base Address + 0x8008, Reset Value = 0x0000_0000 (ICCBPR_CPU2)
- Address = Base Address + 0xC008, Reset Value = 0x0000_0000 (ICCBPR_CPU3)

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
Binary point	[2:0]	RW	The value of this field controls how the 8-bit interrupt priority field should be split into a group priority field. The group priority field is used to determine interrupt preemption, and a sub-priority field. Refer to Priority grouping for more information.	0x0

The register defines the point at which the priority value fields are split into two parts, the group priority field and the sub-priority field. The group priority field is used to determine interrupt preemption.

9.5.1.4 ICCIAR_CPUn

- Base Address: 0x1048_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_03FF (ICCIAR_CPU0)
- Address = Base Address + 0x400C, Reset Value = 0x0000_03FF (ICCIAR_CPU1)
- Address = Base Address + 0x800C, Reset Value = 0x0000_03FF (ICCIAR_CPU2)
- Address = Base Address + 0xC00C, Reset Value = 0x0000_03FF (ICCIAR_CPU3)

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	0x0
CPUID	[12:10]	R	For SGIs, in a multiprocessor implementation, this field identifies the processor that requests the interrupt. It returns the number of the CPU interface that made the request. For all other interrupts, this field returns as zero.	0x0
ACKINTID	[9:0]	R	The interrupt ID	0x3FF

9.5.1.4.1 Effect of the Security Extensions on Reads of the ICCIAR

When a CPU interface implements the Security Extensions, a read of the ICCIAR that returns a valid interrupt ID depends on:

- Whether there is a pending interrupt of sufficient priority for it to be signaled to the processor, and if so, whether the highest priority pending interrupt is a Secure or a Non-secure interrupt
- Whether the ICCIAR read access is Secure or Non-secure
- The value of the ICCICR.AckCtl bit. Refer to CPU Interface Control Register (ICCICR).

Reads of the ICCIAR that do not return a valid interrupt ID returns a spurious interrupt ID: ID 1022 or 1023

Table 9-7 describes all possible ICCIAR reads for a CPU interface that implements the Security Extensions.

Table 9-7 Security Extension of the ICCIAR

Security of Highest Priority Pending Interrupt	ICCIAR read	ICCICR.AckCtl	Returned Interrupt ID
Non-secure	Non-secure	X	ID of Non-secure interrupt
	Secure	1	ID of Non-secure interrupt
		0	Interrupt ID 1022
Secure	Non-secure	X	Interrupt ID 1023
	Secure	X	ID of Secure interrupt
No pending interrupts or signaling of interrupts by CPU interface disabled	X	X	Interrupt ID 1023

9.5.1.5 ICCEOIR_CPUn

- Base Address: 0x1048_0000
- Address = Base Address + 0x0010, Reset Value = Undefined (ICCEOIR_CPU0)
- Address = Base Address + 0x4010, Reset Value = Undefined (ICCEOIR_CPU1)
- Address = Base Address + 0x8010, Reset Value = Undefined (ICCEOIR_CPU2)
- Address = Base Address + 0xC010, Reset Value = Undefined (ICCEOIR_CPU3)

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	–
CPUID	[12:10]	W	During a multiprocessor implementation, on completion of the processing of an SGI, this field contains the CPUID value from the corresponding ICCIAR access.	–
EOIINTID	[9:0]	W	The ACKINTID value from the corresponding ICCIAR access.	–

9.5.1.5.1 Effect of the Security Extensions on Writes to the ICCEOIR

When a CPU interface implements the Security Extensions, a write to the ICCEOIR that removes the active status of the identified interrupt depends on:

- Whether the ICCIAR write is Secure or Non-secure
- The value of the ICCICR.AckCtl bit. Refer to CPU Interface Control Register (ICCICR) for more information.

Table 9-8 describes all possible results of a write to the ICCEOIR.

Table 9-8 Security Extension of the ICCEOIR

Active Interrupt Is	ICCEOIR Write	ICCICR.AckCtl	Active Status Removed
Non-secure	Non-secure	X	Yes
	Secure	1	Yes
		0	No
Secure	Non-secure	X	No
	Secure	X	Yes

9.5.1.6 ICCRPR_CPUn

- Base Address: 0x1048_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_00FF (ICCRPR_CPU0)
- Address = Base Address + 0x4014, Reset Value = 0x0000_00FF (ICCRPR_CPU1)
- Address = Base Address + 0x8014, Reset Value = 0x0000_00FF (ICCRPR_CPU2)
- Address = Base Address + 0xC014, Reset Value = 0x0000_00FF (ICCRPR_CPU3)

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x0
Priority	[7:0]	R	The priority value of the highest priority interrupt that is active on the CPU interface.	0xFF

9.5.1.7 ICCHPIR_CPUn

- Base Address: 0x1048_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_03FF (ICCHPIR_CPU0)
- Address = Base Address + 0x4018, Reset Value = 0x0000_03FF (ICCHPIR_CPU1)
- Address = Base Address + 0x8018, Reset Value = 0x0000_03FF (ICCHPIR_CPU2)
- Address = Base Address + 0xC018, Reset Value = 0x0000_03FF (ICCHPIR_CPU3)

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	0x0
CPUID	[12:10]	R	If the PENDINTID field returns the ID of an SGI, this field contains the CPUID value for that interrupt. This identifies the processor that generates the interrupt.	0x0
PENDINTID	[9:0]	R	The interrupt ID of the highest priority pending interrupt (only for SGI).	0x3FF

9.5.1.7.1 Effect of the Security Extensions on Reads of the ICCHPIR

When a CPU interface implements the Security Extensions, a read of the ICCHPIR that returns a valid interrupt ID depends on:

- Whether there is a pending interrupt of sufficient priority for it to be signaled to the processor, and if so, whether the highest priority pending interrupt is a Secure or a Non-secure interrupt
- Whether the ICCHPIR read access is Secure or Non-secure
- The value of the ICCICR.AckCtl bit. Refer to CPU Interface Control Register (ICCICR) for more information.

Table 9-9 describes all possible ICCHPIR reads for a CPU interface that implements the Security Extensions.

Table 9-9 Security Extension of the ICCHPIR

Security of Highest Priority Pending Interrupt	ICCHPIR Read	ICCICR.AckCtl	Returned Interrupt ID
Non-secure	Non-secure	X	ID of Non-secure interrupt
	Secure	1	ID of Non-secure interrupt
		0	Interrupt ID 1022
Secure	Non-secure	X	Interrupt ID 1023
	Secure	X	ID of Secure interrupt
No pending interrupts or signaling of interrupts by CPU interface disabled	X	X	Interrupt ID 1023

9.5.1.8 ICCABPR_CPUn

- Base Address: 0x1048_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000 (ICCABPR_CPU0)
- Address = Base Address + 0x401C, Reset Value = 0x0000_0000 (ICCABPR_CPU1)
- Address = Base Address + 0x801C, Reset Value = 0x0000_0000 (ICCABPR_CPU2)
- Address = Base Address + 0xC01C, Reset Value = 0x0000_0000 (ICCABPR_CPU3)

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0x0
Binary point	[2:0]	RW	The value of this field controls how the 8-bit interrupt priority field is split into a group priority field. The group priority field is used to determine interrupt preemption, and a sub-priority field. Refer to Priority grouping for more information.	0x0

This register provides an alias of the Non-secure ICCBPR. Refer to Binary Point Register (ICCBPR) Secure register for more information. Only implemented if the GIC implements the Security extensions.

9.5.1.9 INTEG_EN_C_CPUn

- Base Address: 0x1048_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000 (INTEG_C_EN_CPU0)
- Address = Base Address + 0x4040, Reset Value = 0x0000_0000 (INTEG_C_EN_CPU1)
- Address = Base Address + 0x8040, Reset Value = 0x0000_0000 (INTEG_C_EN_CPU2)
- Address = Base Address + 0xC040, Reset Value = 0x0000_0000 (INTEG_C_EN_CPU3)

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
Integ_en_c	[0]	RW	Enables the integration test logic. 0 = Disables the integration test logic 1 = Integration test logic controls the status of nfiq_c<n> and nirq_c<n> output signals. (<n> is a number from 0 to 1 that identifies one of the CPU Interfaces).	0x0

This register enables the integration test logic. Use it to modify the status of the nfiq_c<n> and nirq_c<n> output signals.

9.5.1.10 INTERRUPT_OUT_CPUn

- Base Address: 0x1048_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000 (INTERRUPT_OUT_CPU0)
- Address = Base Address + 0x4044, Reset Value = 0x0000_0000 (INTERRUPT_OUT_CPU1)
- Address = Base Address + 0x8044, Reset Value = 0x0000_0000 (INTERRUPT_OUT_CPU2)
- Address = Base Address + 0xC044, Reset Value = 0x0000_0000 (INTERRUPT_OUT_CPU3)

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
set_nfiq_c	[1]	RW	The value of this field controls how the 8-bit interrupt priority field should be split into a group priority field Use the group priority field is determine Interrupt preemption, and a sub-priority field. Refer to Priority grouping for more information.	0x0
set_nirq_c	[0]	RW	For CPU Interface <n>, reads return the status of nirq_c<n> and writes set the status of nirq_c<n>: 0 = nirq_c<n> a is LOW 1 = nirq_c<n> a is HIGH.	0x0

9.5.1.11 ICCIIDR

- Base Address: 0x1048_0000
- Address = Base Address + 0x00FC, Reset Value = 0x3901_043B

Name	Bit	Type	Description	Reset Value
Id_num	[31:0]	RO	GIC Identification number	0x3901_043B

9.5.1.12 ICDDCR

- Base Address: 0x1049_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x0
Enable	[0]	RW	Global enabled for monitoring peripheral interrupt signals and forwarding pending interrupts to the CPU interfaces. 0 = GIC ignores all peripheral interrupt signals and does not forward pending interrupts to the CPU interfaces. 1 = GIC monitors the peripheral interrupt signals and forwards pending interrupts to the CPU interfaces.	0x0

This register enables forwarding of pending interrupts to the CPU interfaces.

9.5.1.13 ICDICTR

- Base Address: 0x1049_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_FC24

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0
LSPI	[15:11]	R	When GIC implements the Security Extensions, the value of this field is the maximum number of implemented lockable SPIs from 0 (0b00000) to 31 (0b11111).	0x1F
SecurityExtn	[10]	R	Indicates whether GIC implements the Security Extensions. 0 = Does not implement Security Extensions 1 = Implements Security Extensions.	0x1
RSVD	[9:8]	–	Reserved	0x0
CPUNumber	[7:5]	R	Indicates the number of implemented CPU interfaces. The number of implemented CPU interfaces is more than the value of this field. For example, when this field is 0b011, there are four CPU interfaces.	0x1
ITLinesNumber	[4:0]	R	Indicates the maximum number of interrupts that the GIC supports. If the value of this field is N, the maximum number of interrupts is 32 (N + 1). The interrupt ID range is from 0 to one lesser than the number of IDs. For example: 0b00011: Up to 128 interrupt lines, interrupt IDs 0-127. The maximum number of interrupts is 1020 (0b11111).	0x4

This register provides information about the configuration of GIC. It indicates:

- Whether the GIC implements the Security Extensions.
- The maximum number of interrupt IDs that the GIC supports.
- The number of CPU interfaces implemented.
- If GIC implements the Security Extensions, the maximum number of implemented Lockable Shared Peripheral Interrupts (LSPIs).

NOTE: This field defines the interrupt IDs. Regardless of the range of interrupt IDs, interrupt IDs 1020-1023 are reserved for special purposes. Refer to Special interrupt numbers for more information.

The ITLinesNumber field only indicates the maximum number of SPIs that the GIC supports. This value determines the number of implemented interrupt registers, that is, the number of instances of the registers. described in the following sections:

- Interrupt Security Registers (ICDISRn)
- Interrupt Set-Enable Registers (ICDISERn)
- Interrupt Clear-Enable Registers (ICDICERn)
- Interrupt Set-Pending Registers (ICDISPRn)
- Interrupt Clear-Pending Registers (ICDICPRn)
- Active Bit Registers (ICDABRn)
- Interrupt Priority Registers (ICDIPRn)
- Interrupt Processor Targets Registers (ICDIPTRn)
- Interrupt Configuration Registers (ICDICFRn)

9.5.1.14 ICDIIDR

- Base Address: 0x1049_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_043B

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x0
Revision	[23:12]	R	A revision number. Typically, this field is used to distinguish minor revisions of a product.	0x0
Implementer	[11:0]	R	Contains the JEP106 code of the company that implements the GIC Distributor: Bit[11:8]: The JEP106 continuation code of the implementer. Bit[7]: Always 0. Bit[6:0]: The JEP106 identity code of the implementer.	0x43B

This register provides information about the implementer and revision of the Distributor.

NOTE: For an ARM implementation, the value of Implementer field is 0x43B.

9.5.1.15 ICDISR_CPU

- Base Address: 0x1049_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000 (ICDISR0_CPU0)
- Address = Base Address + 0x0084, Reset Value = 0x0000_0000 (ICDISR1_CPU0)
- Address = Base Address + 0x0088, Reset Value = 0x0000_0000 (ICDISR2_CPU0)
- Address = Base Address + 0x008C, Reset Value = 0x0000_0000 (ICDISR3_CPU0)
- Address = Base Address + 0x0090, Reset Value = 0x0000_0000 (ICDISR4_CPU0)
- Address = Base Address + 0x4080, Reset Value = 0x0000_0000 (ICDISR0_CPU1)
- Address = Base Address + 0x8080, Reset Value = 0x0000_0000 (ICDISR0_CPU2)
- Address = Base Address + 0xC080, Reset Value = 0x0000_0000 (ICDISR0_CPU3)

Name	Bit	Type	Description	Reset Value
Security status bits	[31:0]	RW	For each bit: 0 = The corresponding interrupt is Secure. 1 = The corresponding interrupt is Non-secure.	0x0

The ICDISRs provide a Security status bit for each interrupt supported by the GIC. Each bit controls the security status of the corresponding interrupt. These bits are accessible by Secure accesses only.

The number of implemented ICDISRs is (ICDICTR.ITLinesNumber + 1). The implemented ICDISRs number upwards from ICDISR0. In a multiprocessor implementation, ICDISR0 has two registers for each connected processor (for CPU0, Address = 0x1049_0080 and for CPU1, Address = 0x1049_8080). These registers hold the Security status bits for interrupts 0-31.

Figure 9-6 illustrates the ICDISRn address map.

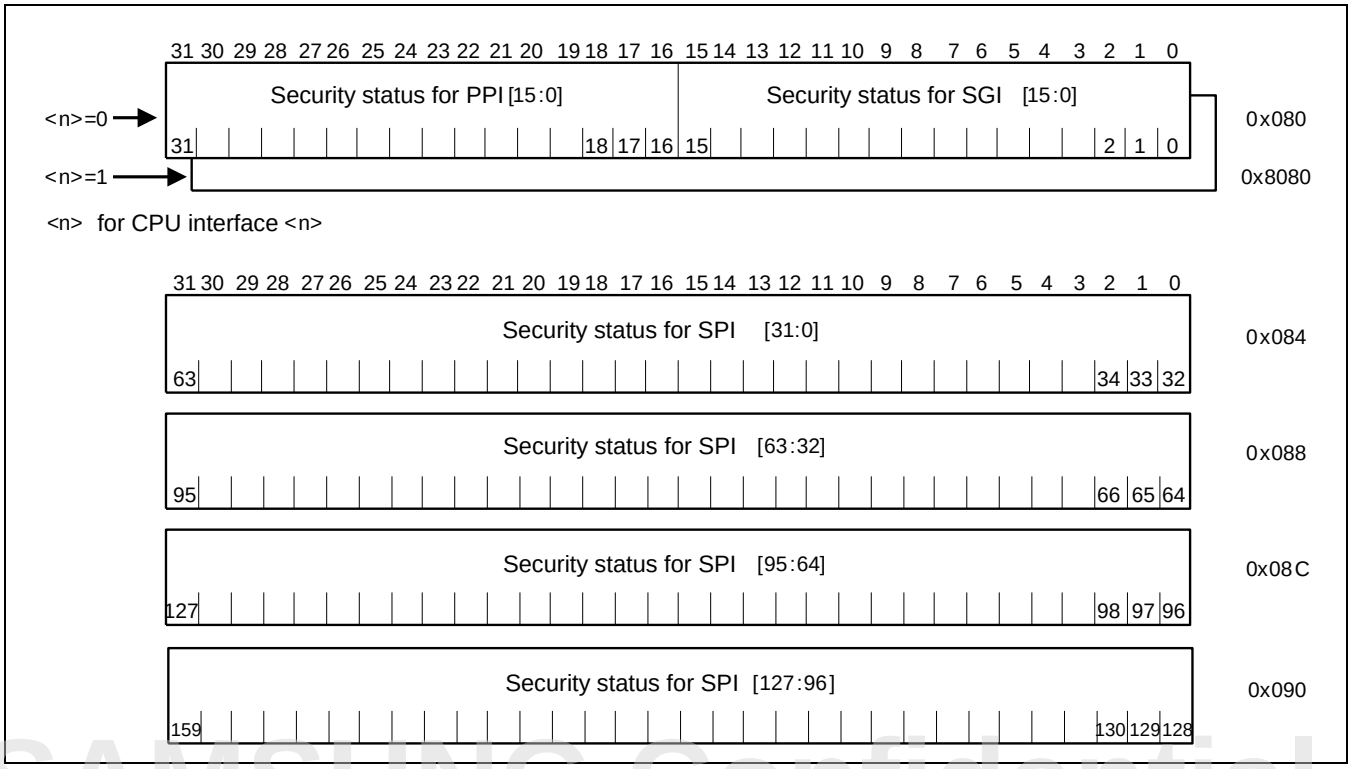


Figure 9-6 ICDISRn Address Map

ARM recommends that you should statically allocate each implemented interrupt as either Secure or Non-secure. To change the security status of an interrupt, ensure that all the status information for that interrupt is drained before you update the appropriate interrupt Security status bit. Normally, the reset value of all ICDISRs is zero. Therefore, all interrupts are secure unless Secure accesses to the appropriate ICDISRs re-programs as Non-secure.

9.5.1.16 ICDISER_CPU

- Base Address: 0x1049_0000
- Address = Base Address + 0x0100, Reset Value = 0x0000_FFFF (ICDISER0_CPU0)
- Address = Base Address + 0x0104, Reset Value = 0x0000_0000 (ICDISER1_CPU0)
- Address = Base Address + 0x0108, Reset Value = 0x0000_0000 (ICDISER2_CPU0)
- Address = Base Address + 0x010C, Reset Value = 0x0000_0000 (ICDISER3_CPU0)
- Address = Base Address + 0x0110, Reset Value = 0x0000_0000 (ICDISER4_CPU0)
- Address = Base Address + 0x4100, Reset Value = 0x0000_FFFF (ICDISER0_CPU1)
- Address = Base Address + 0x8100, Reset Value = 0x0000_FFFF (ICDISER0_CPU2)
- Address = Base Address + 0xC100, Reset Value = 0x0000_FFFF (ICDISER0_CPU3)

Name	Bit	Type	Description	Reset Value
Set-enable bits	[31:0]	RW	For SPIs and PPIs, for each bit: Reads) 0 = Disables the corresponding interrupt. 1 = Enables the corresponding interrupt. Writes) 0 = No effect. 1 = Enables the corresponding interrupt. A subsequent Read of this bit returns the value 1.	0x0

The ICDISERs provide a Set-enable bit for each interrupt supported by the GIC. Writing "1" to a Set-enable bit enables forwarding of the corresponding interrupt to the CPU interfaces. Reading a bit identifies whether the interrupt is enabled. These registers are available in all configurations of GIC. If GIC implements the Security Extensions, these registers are Common.

The Distributor does not provide registers for INTIDs less than 16 because SGIs are always enabled.

Figure 9-7 illustrates the ICDISERn address map.

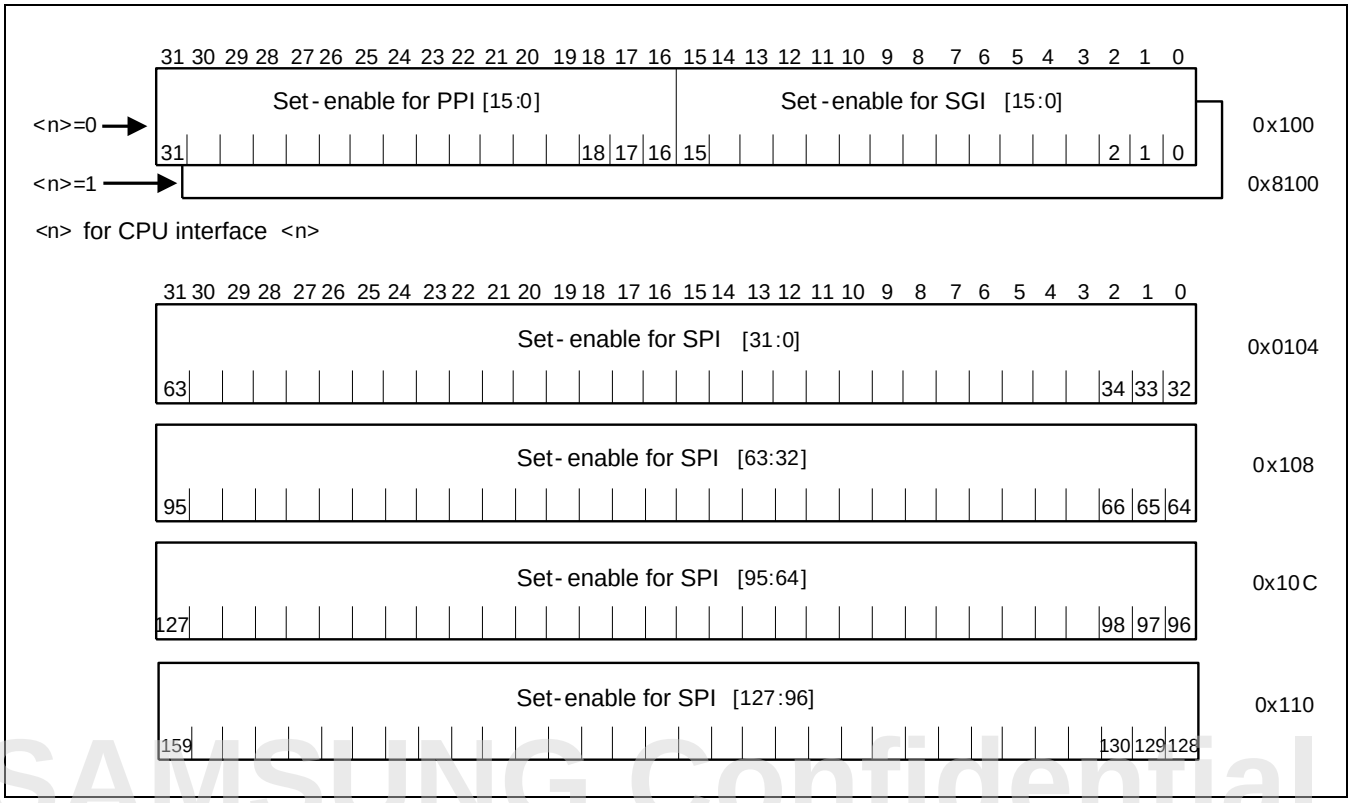


Figure 9-7 ICDISERn Address Map

9.5.1.17 ICDICER_CPU

- Base Address: 0x1049_0000
- Address = Base Address + 0x0180, Reset Value = 0x0000_FFFF (ICDICER0_CPU0)
- Address = Base Address + 0x0184, Reset Value = 0x0000_0000 (ICDICER1_CPU0)
- Address = Base Address + 0x0188, Reset Value = 0x0000_0000 (ICDICER2_CPU0)
- Address = Base Address + 0x018C, Reset Value = 0x0000_0000 (ICDICER3_CPU0)
- Address = Base Address + 0x0190, Reset Value = 0x0000_0000 (ICDICER4_CPU0)
- Address = Base Address + 0x4180, Reset Value = 0x0000_FFFF (ICDICER0_CPU1)
- Address = Base Address + 0x8180, Reset Value = 0x0000_FFFF (ICDICER0_CPU2)
- Address = Base Address + 0xC180, Reset Value = 0x0000_FFFF (ICDICER0_CPU3)

Name	Bit	Type	Description	Reset Value
Set-enable bits	[31:0]	RW	For SPIs and PPIs, for each bit: Reads) 0 = Disables the corresponding interrupt. 1 = Enables the corresponding interrupt. Writes) 0 = No effect. 1 = Disables the corresponding interrupt. A subsequent Read of this bit returns the value "0".	0x0

The ICDICERs provide a Clear-enable bit for each interrupt supported by the GIC. Writing "1" to a Clear-enable bit disables forwarding of the corresponding interrupt to the CPU interfaces. Reading a bit identifies whether the interrupt is enabled. These registers are available in all configurations of GIC. If GIC implements the Security Extensions, these registers are Common.

The Distributor does not provide registers for INTIDs less than 16 because SGIs are always enabled.

Figure 9-8 illustrates the ICDICERn address map.

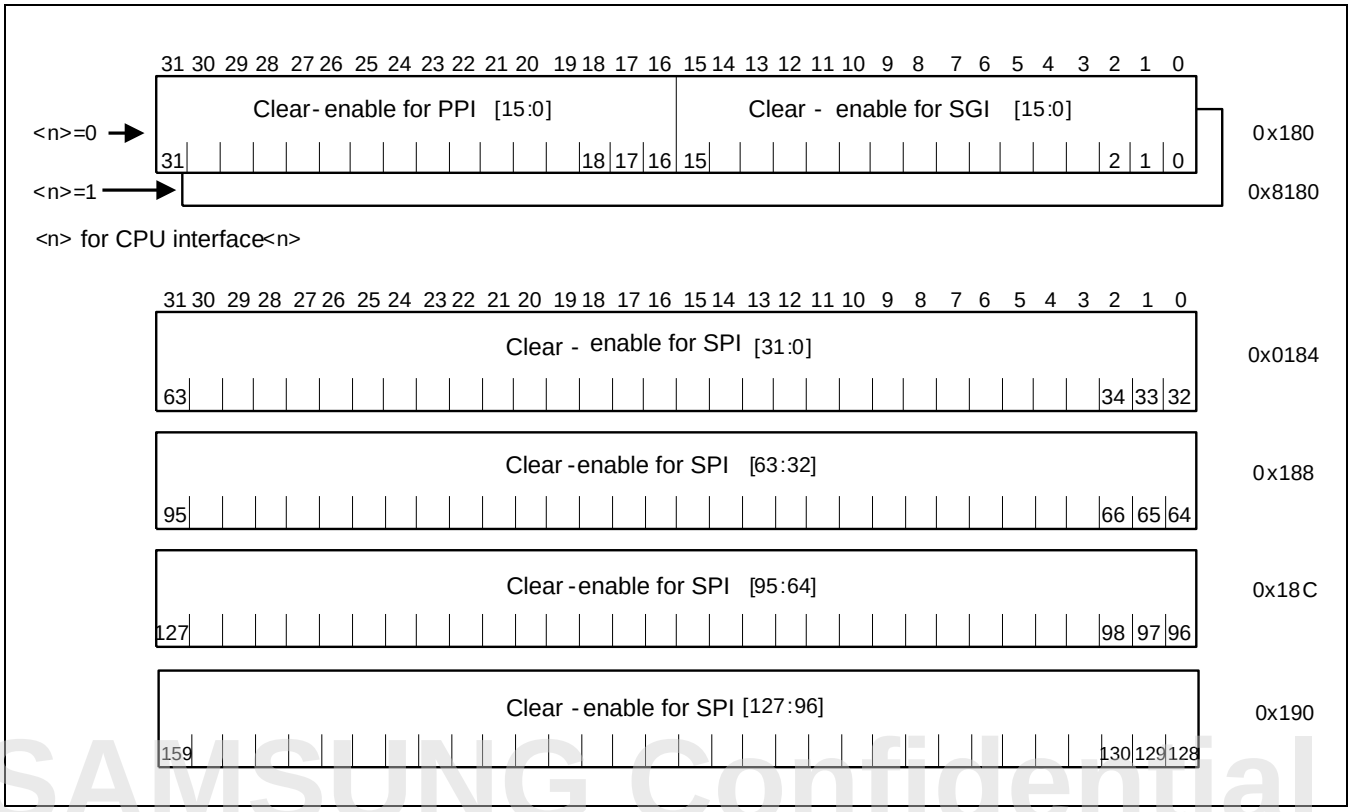


Figure 9-8 ICDICERn Address Map

9.5.1.18 ICDISPR_CPU

- Base Address: 0x1049_0000
- Address = Base Address + 0x0200, Reset Value = 0x0000_0000 (ICDISPR0_CPU0)
- Address = Base Address + 0x0204, Reset Value = 0x0000_0000 (ICDISPR1_CPU0)
- Address = Base Address + 0x0208, Reset Value = 0x0000_0000 (ICDISPR2_CPU0)
- Address = Base Address + 0x020C, Reset Value = 0x0000_0000 (ICDISPR3_CPU0)
- Address = Base Address + 0x0210, Reset Value = 0x0000_0000 (ICDISPR4_CPU0)
- Address = Base Address + 0x4200, Reset Value = 0x0000_0000 (ICDISPR0_CPU1)
- Address = Base Address + 0x8200, Reset Value = 0x0000_0000 (ICDISPR0_CPU2)
- Address = Base Address + 0xC200, Reset Value = 0x0000_0000 (ICDISPR0_CPU3)

Name	Bit	Type	Description	Reset Value
Set-pending bits	[31:0]	RW	**For each bit:** Reads) 0 = The corresponding interrupt is not pending on any processor. 1 = For SGIs and PPIs, the corresponding interrupt is pending on this processor. For SPIs, the corresponding interrupt is pending on at least one processor. **Writes for SPIs and PPIs:** 0 = No effect. 1 = The effect depends on whether the interrupt is edge-triggered or level-sensitive: **Edge-Triggered** Changes the status of the corresponding interrupt to either pending or active and pending: • Pending: If it was previously inactive • Active and pending: If it was previously active. No effect if the interrupt is already pending. **Level Sensitive** When the corresponding interrupt is not pending, changes the status of the corresponding interrupt to either pending or active and pending: • Pending: If it was previously inactive • Active and pending: If it was previously active. When the interrupt is already pending: • A Write has no effect because of a write to the ICDISPR. • Because the corresponding interrupt signal is asserted, the write has no effect on the status of the interrupt, but the interrupt remains pending if the interrupt signal is disserted. For SGIs, the write is ignored.	0x0

The ICDISPRs provide a Set-pending bit for each interrupt supported by the GIC. Writing "1" to a Set-pending bit sets the status of the corresponding peripheral interrupt to pending. Reading a bit identifies whether the interrupt is pending. These registers are available in all configurations of GIC. When GIC implements the Security Extensions, these registers are Common.

The INTIDs for the SGIs are read-only. The Distributor updates these bits by using information from the sgi_control Register.

Figure 9-9 illustrates the ICDISPRn address map.

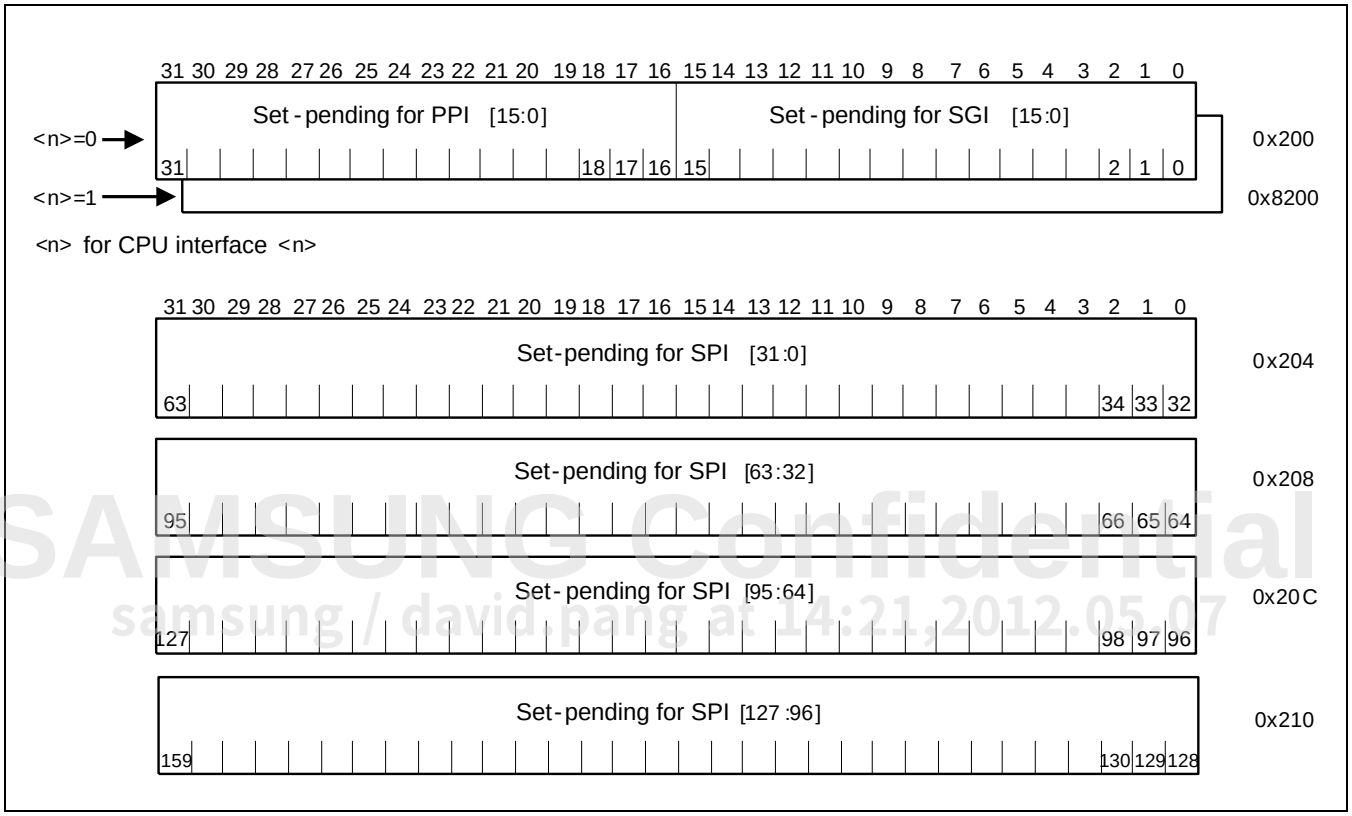


Figure 9-9 ICDISPRn Address Map

9.5.1.19 ICDICPR_CPU

- Base Address: 0x1049_0000
- Address = Base Address + 0x0280, Reset Value = 0x0000_0000 (ICDICPR0_CPU0)
- Address = Base Address + 0x0284, Reset Value = 0x0000_0000 (ICDICPR1_CPU0)
- Address = Base Address + 0x0288, Reset Value = 0x0000_0000 (ICDICPR2_CPU0)
- Address = Base Address + 0x028C, Reset Value = 0x0000_0000 (ICDICPR3_CPU0)
- Address = Base Address + 0x0290, Reset Value = 0x0000_0000 (ICDICPR4_CPU0)
- Address = Base Address + 0x4280, Reset Value = 0x0000_0000 (ICDICPR0_CPU1)
- Address = Base Address + 0x8280, Reset Value = 0x0000_0000 (ICDICPR0_CPU2)
- Address = Base Address + 0xC280, Reset Value = 0x0000_0000 (ICDICPR0_CPU3)

Name	Bit	Type	Description	Reset Value
Clear-pending bits	[31:0]	RW	**For each bit:** Reads) 0 = The corresponding interrupt is not pending on any processor. 1 = For SGIs and PPIs, the corresponding interrupt is pending on this processor. For SPIs, the corresponding interrupt is pending on at least one processor. **Writes for SPIs and PPIs:** 0 = No effect. 1 = The effect depends on whether the interrupt is edge-triggered or level-sensitive: **Edge-Triggered** Changes the status of the corresponding interrupt to either inactive or active: • Inactive: If it was previously pending. • Active: If it was previously active and pending. No effect if the interrupt is not pending. **Level Sensitive** When the corresponding interrupt is pending, only because of a write to the ICDISPR, the write changes the status of the interrupt to either inactive or active: • Inactive: If it was previously pending. • Active: If it was previously active and pending. Otherwise, the interrupt remains pending if the interrupt signal remains asserted. For SGIs, the write is ignored.	0x0

The ICDICPRs provide a Clear-pending bit for each interrupt supported by the GIC. Writing "1" to a Clear-pending bit clears the pending status of the corresponding peripheral interrupt. Reading a bit identifies whether the interrupt is pending. These registers are available in all configurations of the GIC. When the GIC implements the Security Extensions, these registers are Common.

The INTIDs for the SGIs are read-only. The Distributor updates these bits by using information from the sgi_control Register.

Figure 9-10 illustrates the ICDICPRn address map.

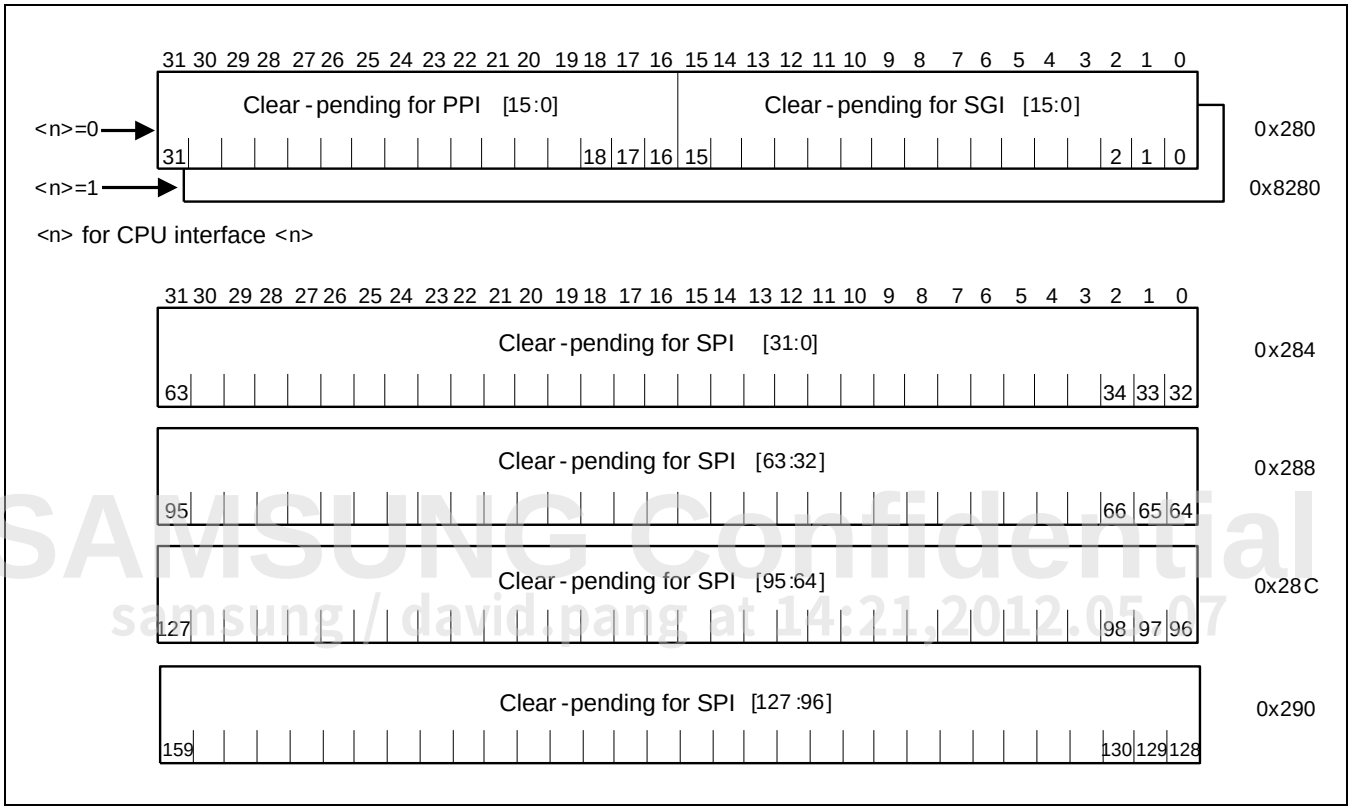


Figure 9-10 ICDICPRn Address Map

9.5.1.19.1 Control of the Pending Status of Level-Sensitive Interrupts

This sub-section describes the status of an interrupt as "includes pending" when the interrupt status is either:

- Pending or
- Active and pending.

For an edge-triggered interrupt, the includes pending status is latched on either a write to the ICDISPR or the assertion of the interrupt signal to the GIC.

For a level-sensitive interrupt, the includes pending status either:

- Is latched on a write to the ICDISPR
- Follows the state of the interrupt signal to the GIC, without any latching.

This means that the operation of the Set-pending and Clear-pending registers is more complicated for level-sensitive interrupts.

Figure 9-11 illustrates the logic of the pending status of a level-sensitive interrupt.

The logical output, status_includes_pending, is TRUE when the interrupt status includes pending, and FALSE otherwise.

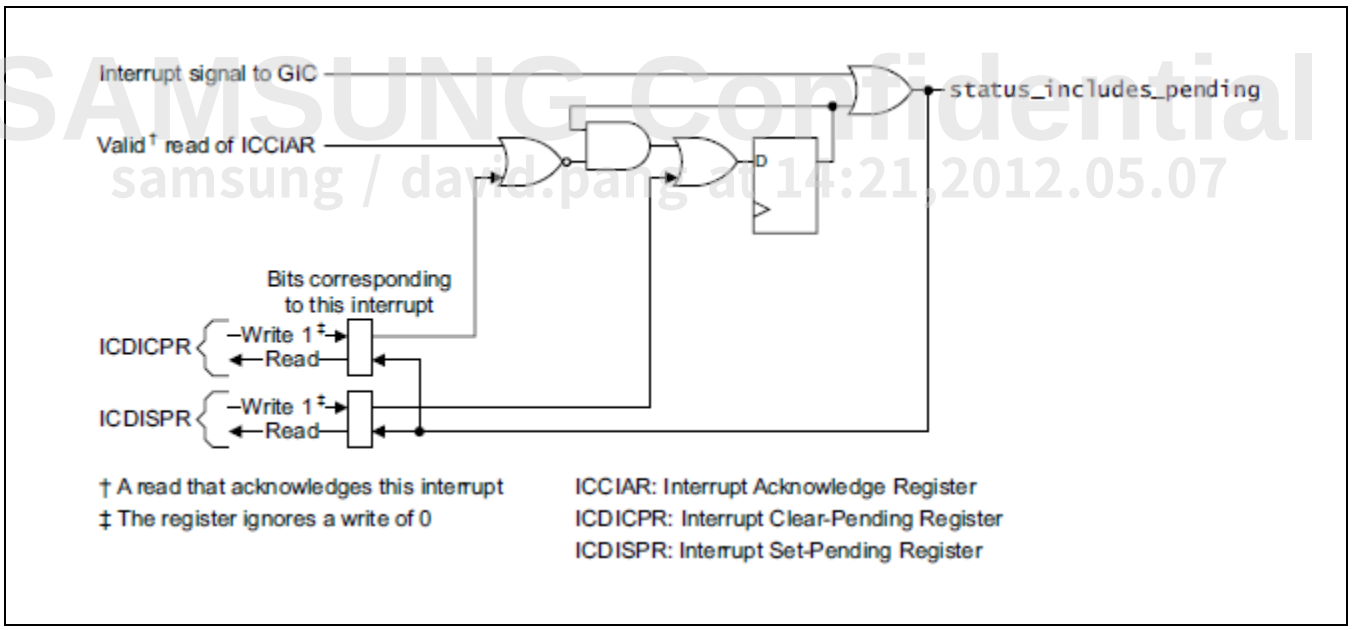


Figure 9-11 Logic of the Pending Status of a Level-sensitive Interrupt

9.5.1.20 ICDABR_CPU

- Base Address: 0x1049_0000
- Address = Base Address + 0x0300, Reset Value = 0x0000_0000 (ICDABR0_CPU0)
- Address = Base Address + 0x0304, Reset Value = 0x0000_0000 (ICDABR1_CPU0)
- Address = Base Address + 0x0308, Reset Value = 0x0000_0000 (ICDABR2_CPU0)
- Address = Base Address + 0x030C, Reset Value = 0x0000_0000 (ICDABR3_CPU0)
- Address = Base Address + 0x0310, Reset Value = 0x0000_0000 (ICDABR4_CPU0)
- Address = Base Address + 0x4300, Reset Value = 0x0000_0000 (ICDABR0_CPU1)
- Address = Base Address + 0x8300, Reset Value = 0x0000_0000 (ICDABR0_CPU2)
- Address = Base Address + 0xC300, Reset Value = 0x0000_0000 (ICDABR0_CPU3)

Name	Bit	Type	Description	Reset Value
Active bits	[31:0]	RW	For each bit: 0 = Corresponding interrupt is not active (NOTE). 1 = Corresponding interrupt is active (NOTE).	0x0

The ICDABRs provide an Active bit for each interrupt supported by the GIC. Reading an Active bit identifies whether the corresponding interrupt is active. The bit reads as "1" when the status of the interrupt is active or active and pending. Read the ICDSPR or ICDCPR to find the pending status of the interrupt. These registers are available in all configurations of GIC. If GIC implements the Security Extensions, these registers are Common.

NOTE: Active interrupts include interrupts that are active and pending.

Figure 9-12 illustrates the ICDABRn address map.

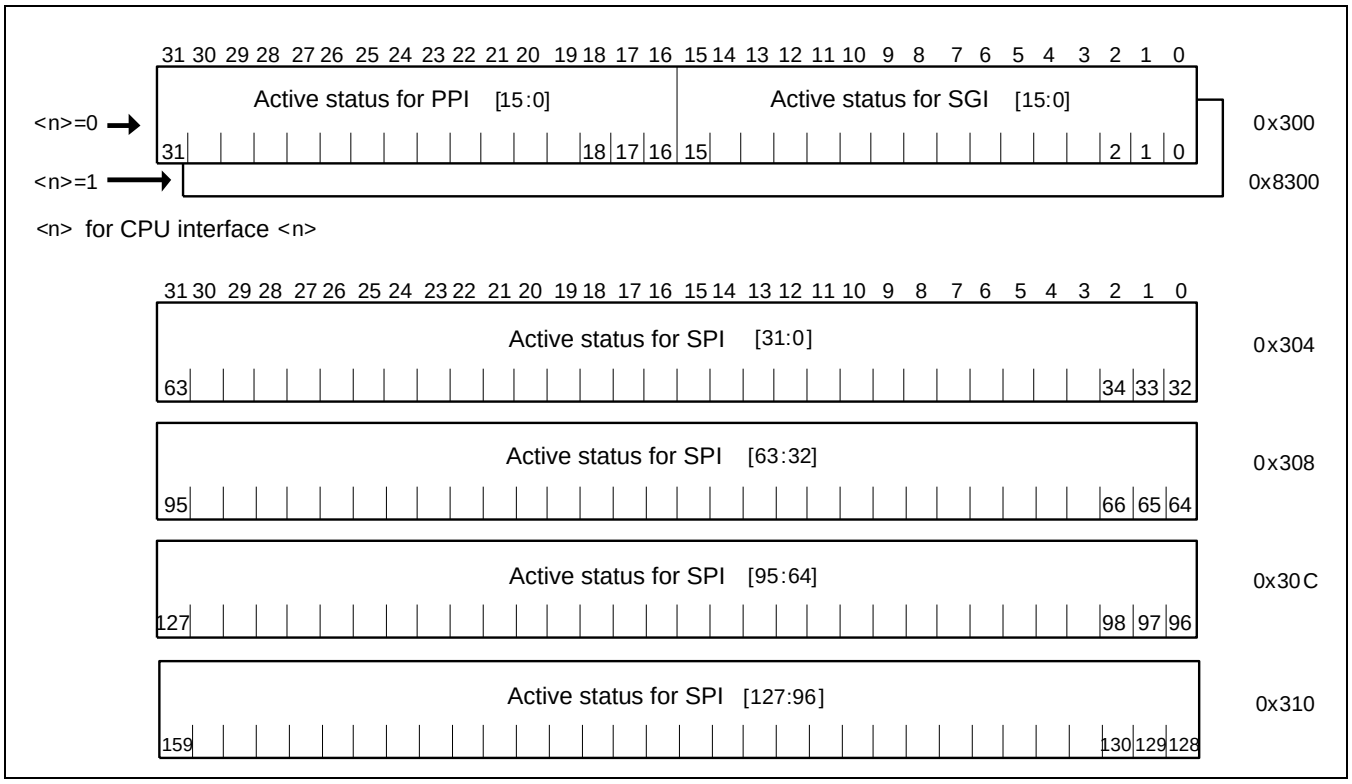


Figure 9-12 ICDABRn Address Map

9.5.1.21 ICDIPR_CPU

- Base Address: 0x1049_0000
- Address = Base Address + 0x0400, Reset Value = 0x0000_0000 (ICDIPR0_CPU0)
- Address = Base Address + 0x0404, Reset Value = 0x0000_0000 (ICDIPR1_CPU0)
- Address = Base Address + 0x0408, Reset Value = 0x0000_0000 (ICDIPR2_CPU0)
- Address = Base Address + 0x040C, Reset Value = 0x0000_0000 (ICDIPR3_CPU0)
- Address = Base Address + 0x0410, Reset Value = 0x0000_0000 (ICDIPR4_CPU0)
- Address = Base Address + 0x0414, Reset Value = 0x0000_0000 (ICDIPR5_CPU0)
- Address = Base Address + 0x0418, Reset Value = 0x0000_0000 (ICDIPR6_CPU0)
- Address = Base Address + 0x041C, Reset Value = 0x0000_0000 (ICDIPR7_CPU0)
- Address = Base Address + 0x0420, Reset Value = 0x0000_0000 (ICDIPR8_CPU0)
- Address = Base Address + 0x0424, Reset Value = 0x0000_0000 (ICDIPR9_CPU0)
- Address = Base Address + 0x0428, Reset Value = 0x0000_0000 (ICDIPR10_CPU0)
- Address = Base Address + 0x042C, Reset Value = 0x0000_0000 (ICDIPR11_CPU0)
- Address = Base Address + 0x0430, Reset Value = 0x0000_0000 (ICDIPR12_CPU0)
- Address = Base Address + 0x0434, Reset Value = 0x0000_0000 (ICDIPR13_CPU0)
- Address = Base Address + 0x0438, Reset Value = 0x0000_0000 (ICDIPR14_CPU0)
- Address = Base Address + 0x043C, Reset Value = 0x0000_0000 (ICDIPR15_CPU0)
- Address = Base Address + 0x0440, Reset Value = 0x0000_0000 (ICDIPR16_CPU0)
- Address = Base Address + 0x0444, Reset Value = 0x0000_0000 (ICDIPR17_CPU0)
- Address = Base Address + 0x0448, Reset Value = 0x0000_0000 (ICDIPR18_CPU0)
- Address = Base Address + 0x044C, Reset Value = 0x0000_0000 (ICDIPR19_CPU0)
- Address = Base Address + 0x0450, Reset Value = 0x0000_0000 (ICDIPR20_CPU0)
- Address = Base Address + 0x0454, Reset Value = 0x0000_0000 (ICDIPR21_CPU0)
- Address = Base Address + 0x0458, Reset Value = 0x0000_0000 (ICDIPR22_CPU0)
- Address = Base Address + 0x045C, Reset Value = 0x0000_0000 (ICDIPR23_CPU0)
- Address = Base Address + 0x0460, Reset Value = 0x0000_0000 (ICDIPR24_CPU0)
- Address = Base Address + 0x0464, Reset Value = 0x0000_0000 (ICDIPR25_CPU0)
- Address = Base Address + 0x0468, Reset Value = 0x0000_0000 (ICDIPR26_CPU0)
- Address = Base Address + 0x046C, Reset Value = 0x0000_0000 (ICDIPR27_CPU0)
- Address = Base Address + 0x0470, Reset Value = 0x0000_0000 (ICDIPR28_CPU0)
- Address = Base Address + 0x0474, Reset Value = 0x0000_0000 (ICDIPR29_CPU0)
- Address = Base Address + 0x0478, Reset Value = 0x0000_0000 (ICDIPR30_CPU0)
- Address = Base Address + 0x047C, Reset Value = 0x0000_0000 (ICDIPR31_CPU0)
- Address = Base Address + 0x0480, Reset Value = 0x0000_0000 (ICDIPR32_CPU0)
- Address = Base Address + 0x0484, Reset Value = 0x0000_0000 (ICDIPR33_CPU0)

- Address = Base Address + 0x0488, Reset Value = 0x0000_0000 (ICDIPR34_CPU0)
- Address = Base Address + 0x048C, Reset Value = 0x0000_0000 (ICDIPR35_CPU0)
- Address = Base Address + 0x0490, Reset Value = 0x0000_0000 (ICDIPR36_CPU0)
- Address = Base Address + 0x0494, Reset Value = 0x0000_0000 (ICDIPR37_CPU0)
- Address = Base Address + 0x0498, Reset Value = 0x0000_0000 (ICDIPR38_CPU0)
- Address = Base Address + 0x049C, Reset Value = 0x0000_0000 (ICDIPR39_CPU0)
- Address = Base Address + 0x4400, Reset Value = 0x0000_0000 (ICDIPR0_CPU1)
- Address = Base Address + 0x8400, Reset Value = 0x0000_0000 (ICDIPR0_CPU2)
- Address = Base Address + 0xC400, Reset Value = 0x0000_0000 (ICDIPR0_CPU3)

Name	Bit	Type	Description	Reset Value
Priority, byte offset 3	[31:24]	RW	Each priority field holds a priority value. Lower the value, greater is the priority of the corresponding interrupt.	0x0
Priority, byte offset 2	[23:16]	RW		0x0
Priority, byte offset 1	[15:8]	RW		0x0
Priority, byte offset 0	[7:0]	RW		0x0

The ICDIPRs provide an 8-bit Priority field for each interrupt supported by the GIC. This field stores the priority of the corresponding interrupt. These registers are available in all configurations of GIC. If GIC implements the Security Extensions, these registers are Common.

Figure 9-13 illustrates the ICDIPRn address map.

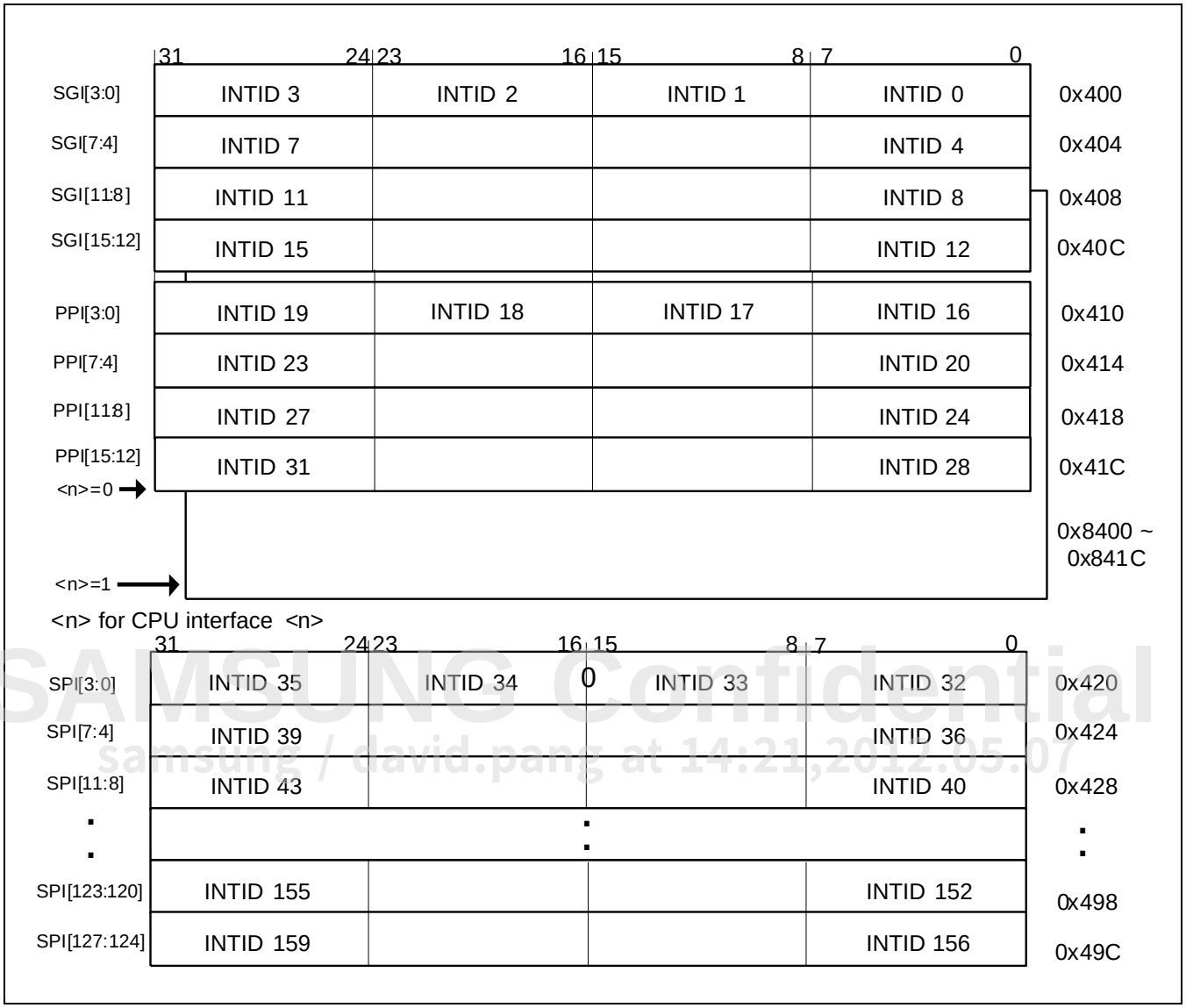


Figure 9-13 ICDIPRn Address Map

9.5.1.22 ICDIPTR_CPU

- Base Address: 0x1049_0000
- Address = Base Address + 0x0800, Reset Value = 0x0101_0101 (ICDIPTR0_CPU0)
- Address = Base Address + 0x0804, Reset Value = 0x0101_0101 (ICDIPTR1_CPU0)
- Address = Base Address + 0x0808, Reset Value = 0x0101_0101 (ICDIPTR2_CPU0)
- Address = Base Address + 0x080C, Reset Value = 0x0101_0101 (ICDIPTR3_CPU0)
- Address = Base Address + 0x0810, Reset Value = 0x0101_0101 (ICDIPTR4_CPU0)
- Address = Base Address + 0x0814, Reset Value = 0x0101_0101 (ICDIPTR5_CPU0)
- Address = Base Address + 0x0818, Reset Value = 0x0101_0101 (ICDIPTR6_CPU0)
- Address = Base Address + 0x081C, Reset Value = 0x0101_0101 (ICDIPTR7_CPU0)
- Address = Base Address + 0x0820, Reset Value = 0x0000_0000 (ICDIPTR8_CPU0)
- Address = Base Address + 0x0824, Reset Value = 0x0000_0000 (ICDIPTR9_CPU0)
- Address = Base Address + 0x0828, Reset Value = 0x0000_0000 (ICDIPTR10_CPU0)
- Address = Base Address + 0x082C, Reset Value = 0x0000_0000 (ICDIPTR11_CPU0)
- Address = Base Address + 0x0830, Reset Value = 0x0000_0000 (ICDIPTR12_CPU0)
- Address = Base Address + 0x0834, Reset Value = 0x0000_0000 (ICDIPTR13_CPU0)
- Address = Base Address + 0x0838, Reset Value = 0x0000_0000 (ICDIPTR14_CPU0)
- Address = Base Address + 0x083C, Reset Value = 0x0000_0000 (ICDIPTR15_CPU0)
- Address = Base Address + 0x0840, Reset Value = 0x0000_0000 (ICDIPTR16_CPU0)
- Address = Base Address + 0x0844, Reset Value = 0x0000_0000 (ICDIPTR17_CPU0)
- Address = Base Address + 0x0848, Reset Value = 0x0000_0000 (ICDIPTR18_CPU0)
- Address = Base Address + 0x084C, Reset Value = 0x0000_0000 (ICDIPTR19_CPU0)
- Address = Base Address + 0x0850, Reset Value = 0x0000_0000 (ICDIPTR20_CPU0)
- Address = Base Address + 0x0854, Reset Value = 0x0000_0000 (ICDIPTR21_CPU0)
- Address = Base Address + 0x0858, Reset Value = 0x0000_0000 (ICDIPTR22_CPU0)
- Address = Base Address + 0x085C, Reset Value = 0x0000_0000 (ICDIPTR23_CPU0)
- Address = Base Address + 0x0860, Reset Value = 0x0000_0000 (ICDIPTR24_CPU0)
- Address = Base Address + 0x0864, Reset Value = 0x0000_0000 (ICDIPTR25_CPU0)
- Address = Base Address + 0x0868, Reset Value = 0x0000_0000 (ICDIPTR26_CPU0)
- Address = Base Address + 0x086C, Reset Value = 0x0000_0000 (ICDIPTR27_CPU0)
- Address = Base Address + 0x0870, Reset Value = 0x0000_0000 (ICDIPTR28_CPU0)
- Address = Base Address + 0x0874, Reset Value = 0x0000_0000 (ICDIPTR29_CPU0)
- Address = Base Address + 0x0878, Reset Value = 0x0000_0000 (ICDIPTR30_CPU0)
- Address = Base Address + 0x087C, Reset Value = 0x0000_0000 (ICDIPTR31_CPU0)
- Address = Base Address + 0x0880, Reset Value = 0x0000_0000 (ICDIPTR32_CPU0)
- Address = Base Address + 0x0884, Reset Value = 0x0000_0000 (ICDIPTR33_CPU0)

- Address = Base Address + 0x0888, Reset Value = 0x0000_0000 (ICDIPTR34_CPU0)
- Address = Base Address + 0x088C, Reset Value = 0x0000_0000 (ICDIPTR35_CPU0)
- Address = Base Address + 0x0890, Reset Value = 0x0000_0000 (ICDIPTR36_CPU0)
- Address = Base Address + 0x0894, Reset Value = 0x0000_0000 (ICDIPTR37_CPU0)
- Address = Base Address + 0x0898, Reset Value = 0x0000_0000 (ICDIPTR38_CPU0)
- Address = Base Address + 0x089C, Reset Value = 0x0000_0000 (ICDIPTR39_CPU0)
- Address = Base Address + 0x4800, Reset Value = 0x0202_0202 (ICDIPTR0_CPU1)
- Address = Base Address + 0x8800, Reset Value = 0x0202_0202 (ICDIPTR0_CPU2)
- Address = Base Address + 0xC800, Reset Value = 0x0202_0202 (ICDIPTR0_CPU3)

Name	Bit	Type	Description	Reset Value
CPU targets, byte offset 3	[31:24]	RW	Processors in the system number from 0, and each bit in a CPU targets field refers to the corresponding processor. Refer to *Table 9-10* for more information.	0x0
CPU targets, byte offset 2	[23:16]	RW		0x0
CPU targets, byte offset 1	[15:8]	RW		0x0
CPU targets, byte offset 0	[7:0]	RW	For example, a value of 0x3 means that the Pending interrupt is sent to processors 0 and 1. For ICDIPTR0 to ICDIPTR7, a Read of any CPU targets field returns the number of the processor that performs the read.	0x0

The ICDIPTRs provide an 8-bit CPU targets field for each interrupt supported by the GIC. This field stores the list of processors that the interrupt is sent to if it is asserted. ICDIPTR0 to ICDIPTR7 are Read-only, and each field returns a value corresponding only to the processor that reads the register. These registers are available in all configurations of GIC. When GIC implements the Security Extensions, these registers are Common.

Table 9-10 describes the meaning of CPU targets field bit values.

Table 9-10 Meaning of CPU Targets Field Bit Values

CPU Targets Field Value	Interrupt Targets
0bxxxxxxx1	CPU interface 0
0bxxxxxx1x	CPU interface 1
0bxxxxx1xx	CPU interface 2
0bxxxx1xxx	CPU interface 3
0bxxx1xxxx	CPU interface 4
0bxx1xxxxx	CPU interface 5
0bx1xxxxxx	CPU interface 6
0b1xxxxxxx	CPU interface 7

9.5.1.22.1 The Effect of Changes to an ICDIPTR

Software can write to an ICDIPTR any time. The effects of any change to a CPU targets field value are:

- No effect on any active interrupt: This means that removing a CPU interface from a targets list does not cancel an active state for that interrupt on that CPU interface.
- Immediate effect on any pending interrupts: This means:
  - Adding a CPU interface to the target list of a pending interrupt makes that interrupt pending on that CPU interface
  - Removing a CPU interface from the target list of a pending interrupt removes the pending state of that interrupt on that CPU interface.
- If applied to an interrupt that is active and pending, it will not change the interrupt targets until it clears the active status.

Figure 9-14 illustrates ICDIPTRn address map.

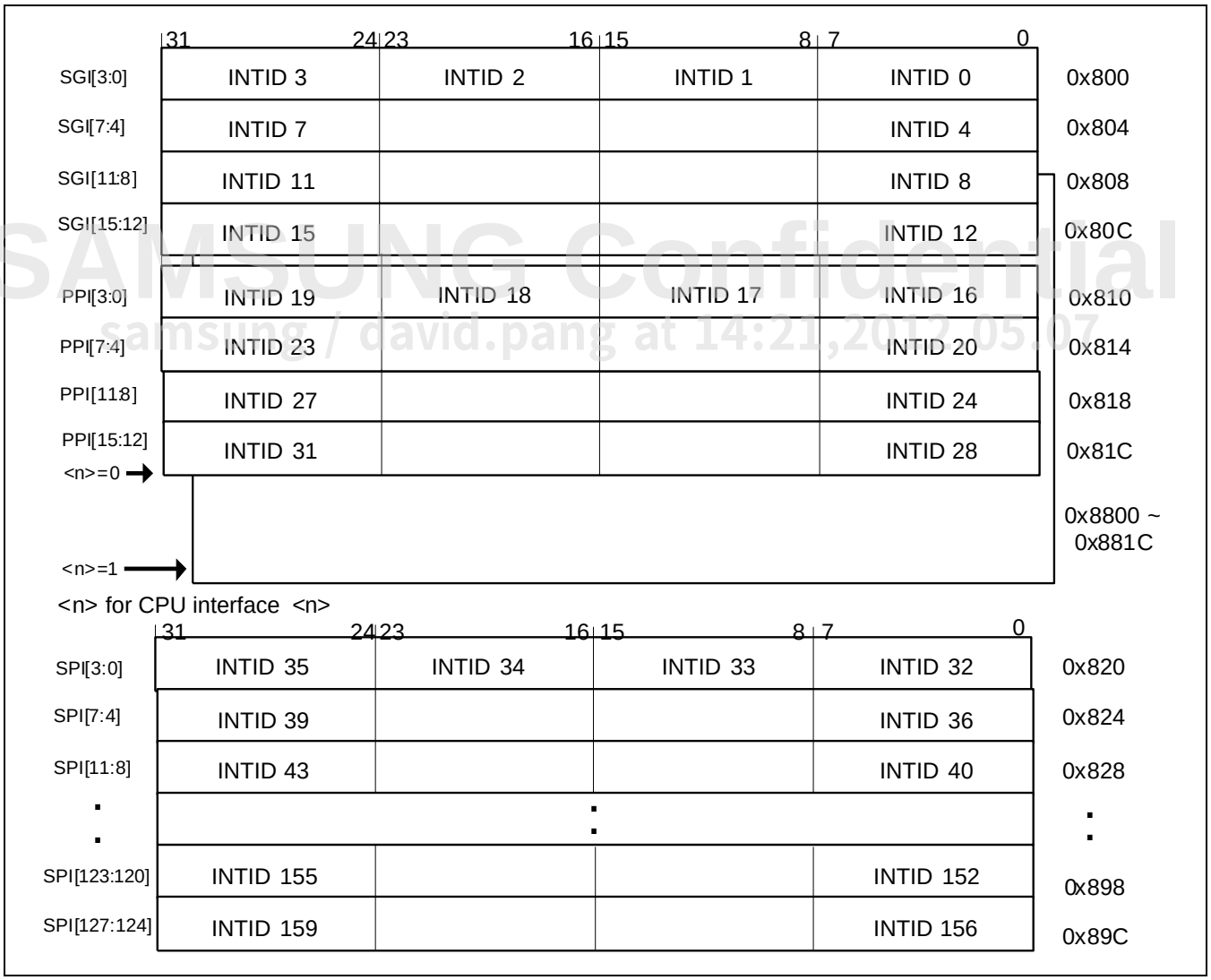


Figure 9-14 ICDIPTRn Address Map

9.5.1.23 ICDICFR_CPU

- Base Address: 0x1049_0000
- Address = Base Address + 0x0C00, Reset Value = 0xAAAA_AAAA (ICDICFR0_CPU0)
- Address = Base Address + 0x0C04, Reset Value = 0x7DD5_5FFF (ICDICFR1_CPU0)
- Address = Base Address + 0x0C08, Reset Value = 0x5555_5555 (ICDICFR2_CPU0)
- Address = Base Address + 0x0C0C, Reset Value = 0x5555_5555 (ICDICFR3_CPU0)
- Address = Base Address + 0x0C10, Reset Value = 0x5555_5555 (ICDICFR4_CPU0)
- Address = Base Address + 0x0C14, Reset Value = 0x5555_5555 (ICDICFR5_CPU0)
- Address = Base Address + 0x0C18, Reset Value = 0x5555_5555 (ICDICFR6_CPU0)
- Address = Base Address + 0x0C1C, Reset Value = 0x5555_5555 (ICDICFR7_CPU0)
- Address = Base Address + 0x0C20, Reset Value = 0x5555_5555 (ICDICFR8_CPU0)
- Address = Base Address + 0x0C24, Reset Value = 0x5555_5555 (ICDICFR9_CPU0)
- Address = Base Address + 0x4C00, Reset Value = 0xAAAA_AAAA (ICDICFR0_CPU1)
- Address = Base Address + 0x8C00, Reset Value = 0xAAAA_AAAA (ICDICFR0_CPU2)
- Address = Base Address + 0xCC00, Reset Value = 0xAAAA_AAAA (ICDICFR0_CPU3)

Name	Bit	Type	Description	Reset Value
Int_config, filed F	[2F + 1:2F]	R	For Int_config[1], the most significant bit, bit[2F + 1], the encoding is: 0 = Corresponding interrupt is level-sensitive. 1 = Corresponding interrupt is edge-triggered. Int_config[0], the least significant bit, bit[2F], is Reserved. For SGIs and PPIs: Int_config[1] Not programmable	0x01

The ICDICFRs provide a 2-bit Int_config field for each interrupt supported by the GIC. This field identifies whether the corresponding interrupt is edge-triggered or level-sensitive. For SGIs and PPIs, Int_config fields are Read-only. This means ICDICFR0 and ICDICFR1 are Read-only. These registers are available in all configurations of GIC. If GIC implements the Security Extensions, these registers are Common.

Figure 9-15 illustrates the ICDICFRn bit assignments.

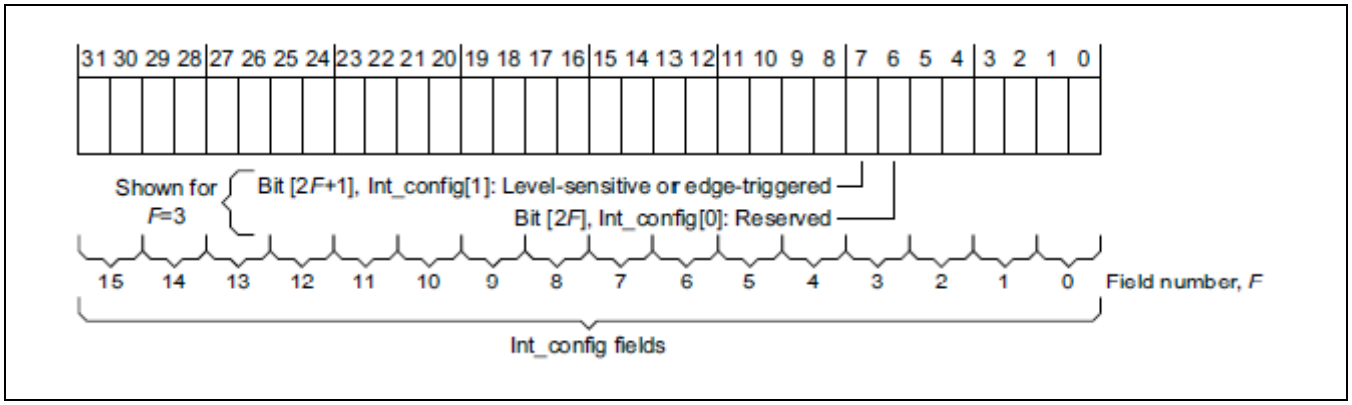


Figure 9-15 ICDICFRn Bit Assignments

Figure 9-16 illustrates the ICDICFRn address map.

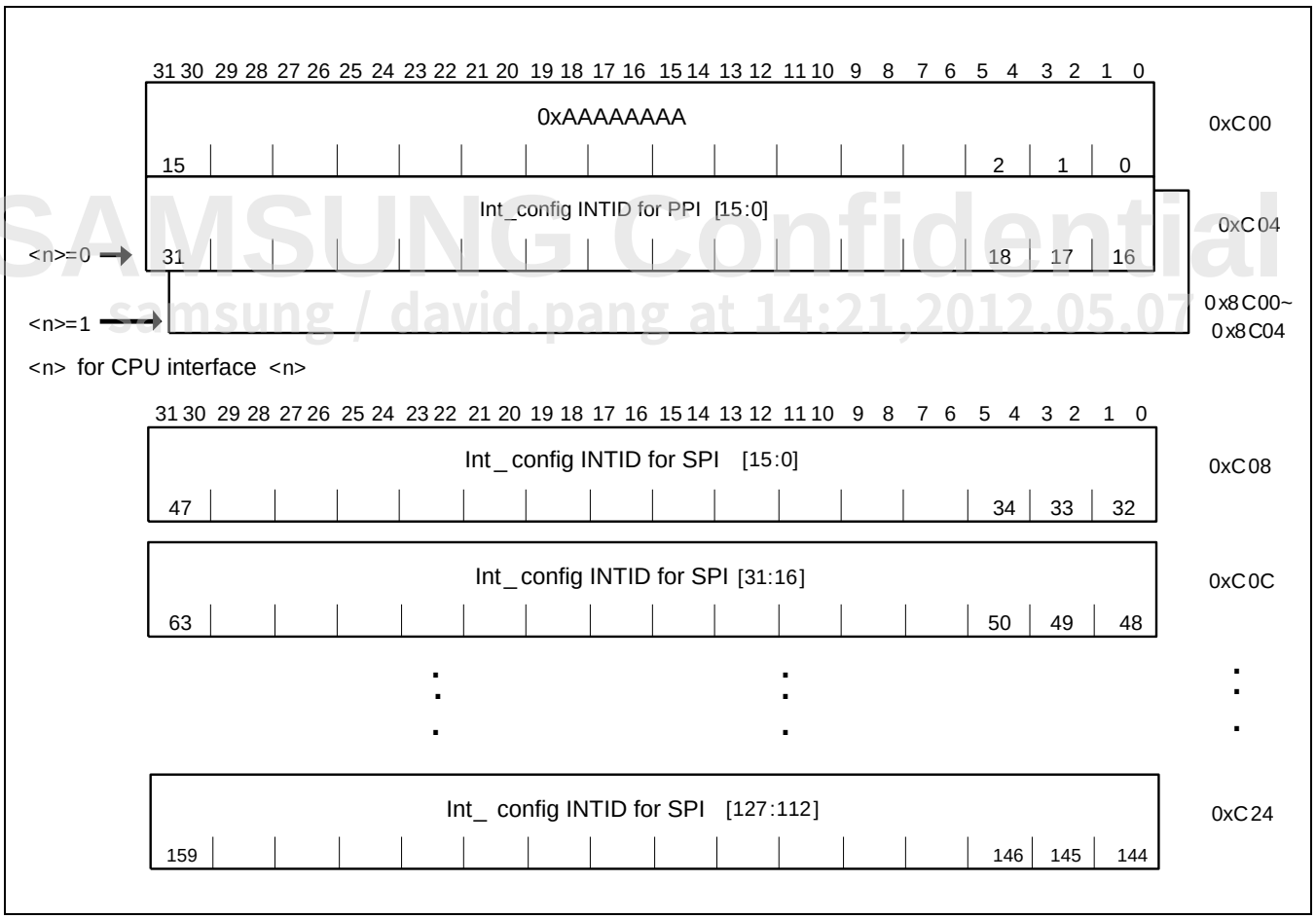


Figure 9-16 ICDICFRn Address Map

9.5.1.24 PPI_STATUS_CPU

- Base Address: 0x1049_0000
- Address = Base Address + 0x0D00, Reset Value = 0x0000_0000 (PPI_STATUS_CPU0)
- Address = Base Address + 0x4D00, Reset Value = 0x0000_0000 (PPI_STATUS_CPU1)
- Address = Base Address + 0x8D00, Reset Value = 0x0000_0000 (PPI_STATUS_CPU2)
- Address = Base Address + 0xCD00, Reset Value = 0x0000_0000 (PPI_STATUS_CPU3)

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0
ppi_status	[15:0]	R	Returns the status of the ppi_c<n>[15:0] inputs on the Distributor: Bit[X] = 0 ppi_c<n>[x] is LOW Bit[X] = 1 ppi_c<n>[x] is HIGH.	0x0

Each bit provides the status of the ppi_c<n>[15:0] inputs for CPU Interface <n>. These registers are only accessible to processors in secure state.

9.5.1.25 SPI_STATUS

- Base Address: 0x1049_0000
- Address = Base Address + 0x0D04, Reset Value = 0x0000_0000 (SPI_STATUS0)
- Address = Base Address + 0x0D08, Reset Value = 0x0000_0000 (SPI_STATUS1)
- Address = Base Address + 0x0D0C, Reset Value = 0x0000_0000 (SPI_STATUS2)
- Address = Base Address + 0x0D10, Reset Value = 0x0000_0000 (SPI_STATUS3)

Name	Bit	Type	Description	Reset Value
spi_status	[31:0]	R	Returns the status of the spi[127:0] inputs on the Distributor: Bit[x] = 0 spi[x] is LOW Bit[x] = 1 spi[x] is HIGH.	0x0

Each bit provides the status of the SPI[127:0] inputs. This register is only accessible to processors in Secure state.

NOTE: These bits return the actual status of the spi signals. The Pending Set Register (ICDISPR) and Pending Clear Register (ICDICPR) also provide the spi status. As you can write to these registers, they do not contain the actual status of the SPI signals.

Figure 9-17 illustrates the SPI_STATUSn address map.

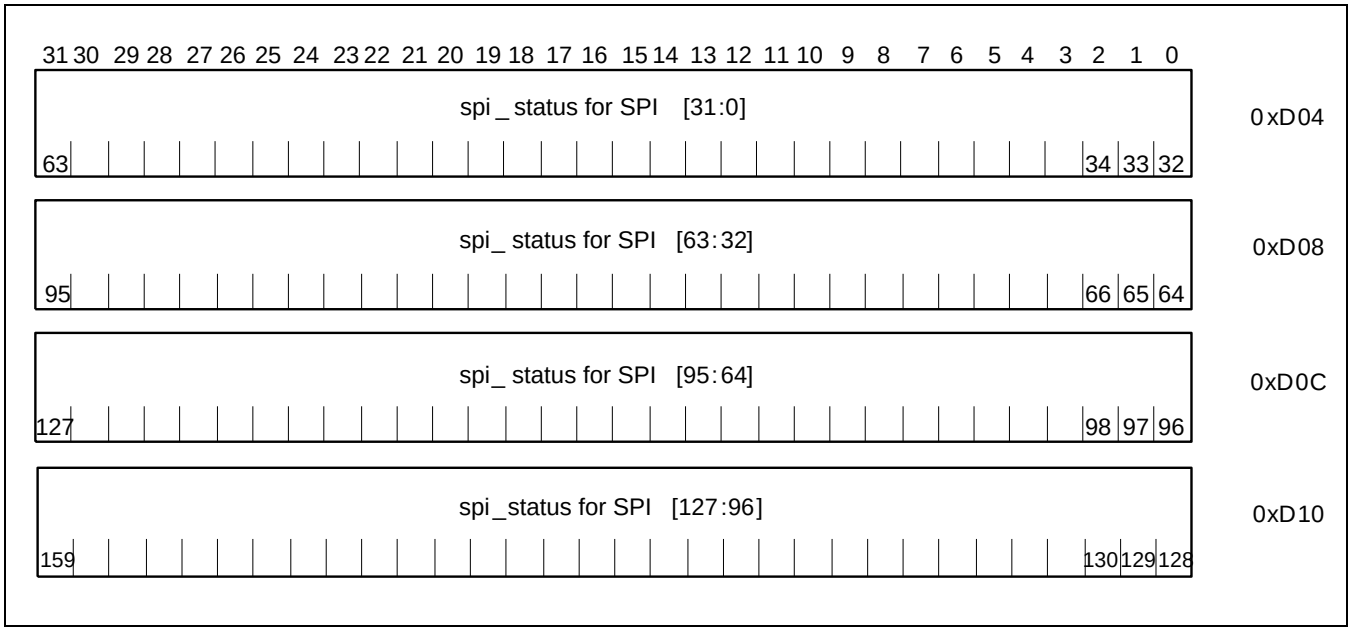


Figure 9-17 SPI_STATUSn Address Map

9.5.1.26 ICDSGIR

- Base Address: 0x1049_0000
- Address = Base Address + 0x0F00, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	–
TargetListFilter	[25:24]	W	0b00: Sends the interrupt to the CPU interfaces specified in the CPUTargetList field. 0b01: Sends the interrupt to all CPU interfaces except the CPU interface that requested the interrupt. 0b10: Sends the interrupt only to the CPU interface that requests the interrupt. 0b11: Reserved.	–
CPUTargetList	[23:16]	W	When TargetList Filter = 0b00, it defines the CPU interfaces the Distributor must send the interrupt to. Each bit of CPUTargetList[7:0] refers to the corresponding CPU interface, for example CPUTargetList[0] corresponds to CPU interface 0. Setting a bit to 1 sends the interrupt to the corresponding interface.	–
SATT	[15]	W	This field is writable only by using a Secure access. Specifies the required security value of the SGI: 0 = Sends the SGI specified in the SGIINTID field to a specified CPU interface only if the SGI is configured as Secure on that interface. 1 = Sends the specified SGI in the SGIINTID field to specified CPU interfaces only if the SGI is configured as Non-secure on that interface.	–
RSVD	[14:4]	–	Reserved	–
SGINTID	[3:0]	W	The Interrupt ID of the SGI to send to the specified CPU interfaces. The value of this field is the Interrupt ID, in the range 0-15. For example, a value of 0b0011 specifies Interrupt ID 3.	–

This register controls the generation of SGIs. This register is available in all configurations of the GIC. When the GIC implements the Security Extensions, this register is Common.

NOTE: When TargetListFilter is 0b00, if the CPUTargetList field is 0x00, the Distributor does not send the interrupt to any CPU interface.

9.5.1.26.1 SGI Generation when the GIC Implements the Security Extensions

When the GIC implements the Security Extensions, whether an SGI is sent to a processor specified in the write to the ICDSGIR depends on:

- The security status of the write to the ICDSGIR
- The value of the ICDSGIR.SATT bit for a Secure write to the ICDSGIR
- Whether the specified SGI is configured as Secure or Non-secure on the targeted processor.

ICDISR0 holds the security states of the SGIs. Refer to Interrupt Security Registers (ICDISRn) for more information. In a multiprocessor system, ICDISR0_CPU0 and ICDISR0_CPU1 are for each connected processor. Therefore, the system configures the security of each SGI independently for each processor.

A single write to the ICDSGIR can target multiple processors. For each targeted processor, the Distributor determines whether to send the SGI to the processor.

Table 9-11 illustrates the truth table for sending SGI to a target processor.

Table 9-11 Truth Table for Sending a SGI to a Target Processor

Status of ICDSGIR Write	SATT Value	Status of SGI on Target Processor	Send SGI?
Secure	0	Secure	Yes
		Non-secure	No
	1	Secure	No
		Non-secure	Yes
Non-secure	X	Secure	No
		Non-secure	Yes

10 Interrupt Combiner

10.1 Overview

Interrupt controller in Exynos 4412 SCP consists of:

- PrimeCell generic interrupt controller (PL390)
- Interrupt combiner

A few interrupt sources are grouped in Exynos 4412 SCP. Interrupt combiner combines several interrupt sources as a group. Several interrupt requests in a group make a group interrupt request and a single request signal. As a result, the interrupt input sources of PrimeCell generic interrupt controller consists of the group interrupt requests from the interrupt combiner and uncombined interrupt sources.

10.1.1 Features

The features of Interrupt Combiner are:

- 116 interrupt source inputs.
- 18 group interrupt outputs.
- Enables or masks each interrupt source in a group.
- Provides the status of interrupt source in a group before interrupt masking.
- Provides the status of interrupt source in a group after interrupt masking.
- Provides the status of group interrupt output after interrupt masking and combining

10.1.2 Block Diagram

There is a interrupt combiner in Exynos 4412 SCP. The interrupt combiner combines a few interrupts source into 18 group interrupt request outputs. The inputs of the GIC unit outside the ARM Core unit connect to the group interrupt request outputs.

Figure 10-1 illustrates the block diagram of interrupt combiner.

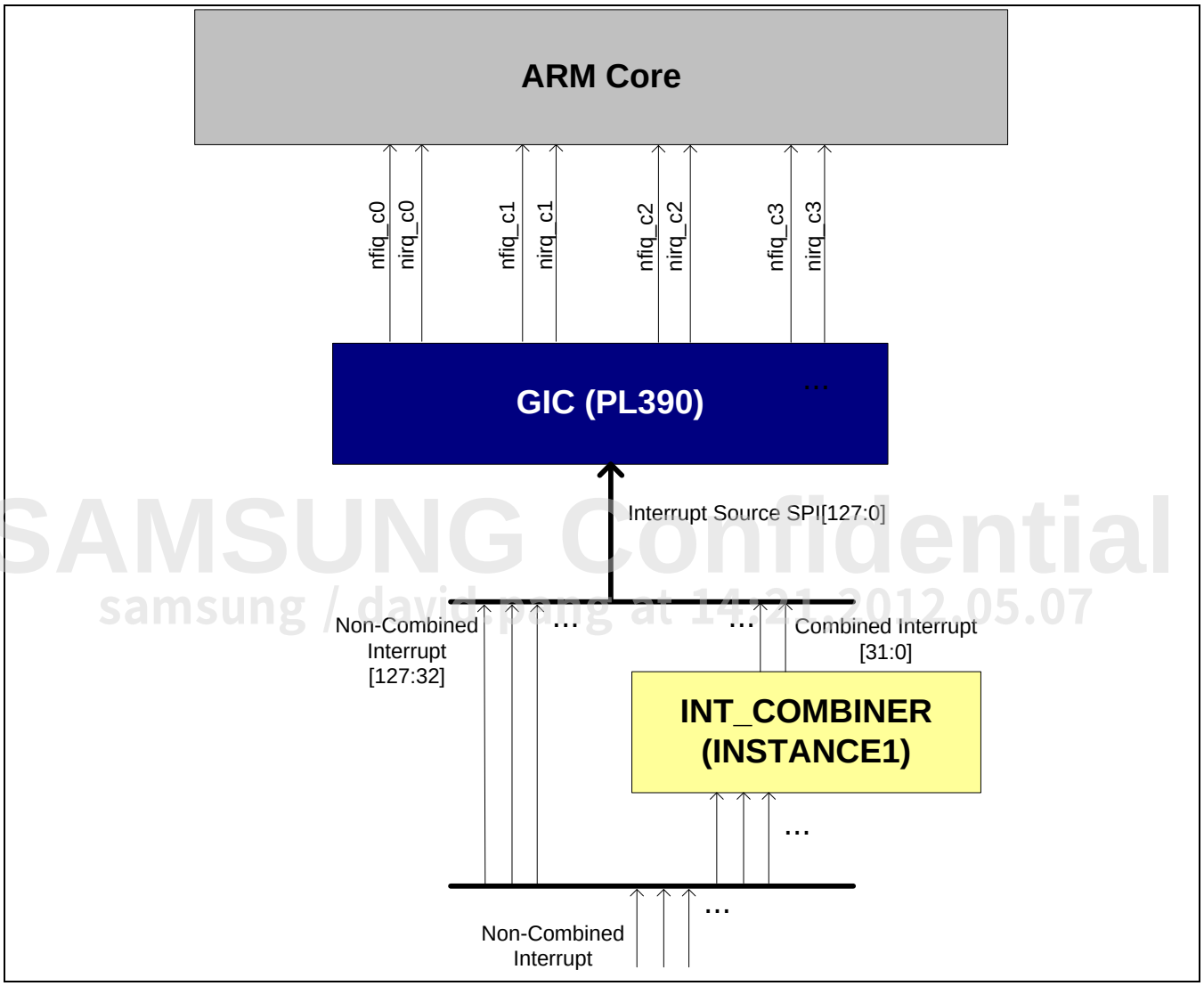


Figure 10-1 Block Diagram of Interrupt Combiner

Figure 10-2 illustrates the internal operation of interrupt combiner.

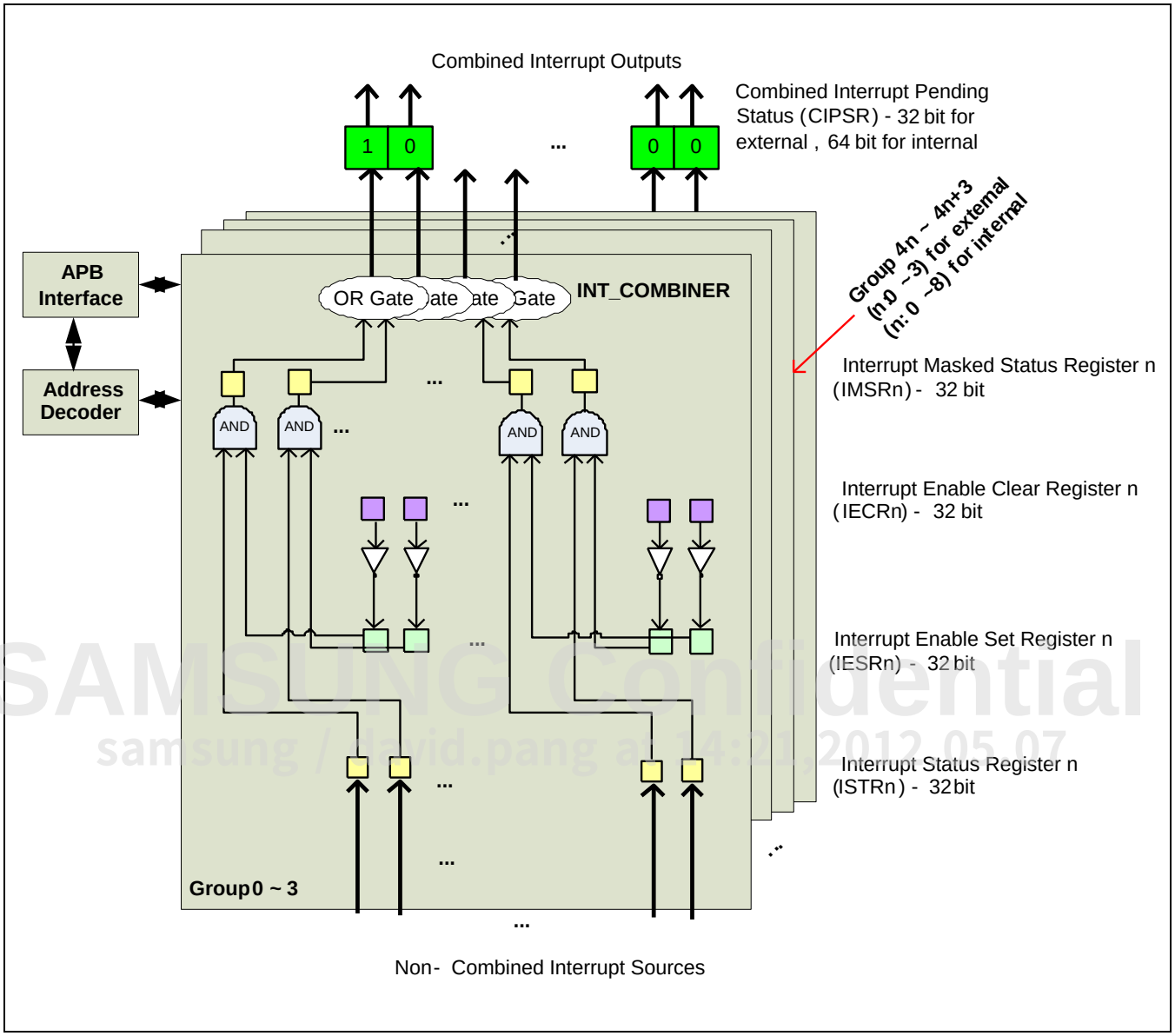


Figure 10-2 Internal Operation of Interrupt Combiner

10.2 Interrupt Sources

This section includes:

- Interrupt Combiner

10.2.1 Interrupt Combiner

Table 10-1 describes the interrupt groups of interrupt combiner.

Table 10-1 Interrupt Groups of Interrupt Combiner

Combiner Group ID	Combined Interrupt Source Name	Bit	Interrupt Source	Source Block
INTG0	MDNIE_LCD0	[3]	MDNIE_LCD0[3]	MDNIE_LCD0
		[2]	MDNIE_LCD0[2]	
		[1]	MDNIE_LCD0[1]	
		[0]	MDNIE_LCD0[0]	
INTG1	TZASC	[3]	TZASC1[1]	TZASC
		[2]	TZASC1[0]	
		[1]	TZASC0[1]	
		[0]	TZASC0[0]	
INTG2	PARITYFAIL0/ F4DCTI0/PMU0	[6]	PARRINTR	CPU0
		[5]	PARRDINTR	
		[4]	TMU	
		[3]	nCTIIRQ[0]	
		[2]	PMUIRQ[0]	
		[1]	PARITYFAILSCU[0]	
		[0]	PARITYFAIL0	
INTG3	PARITYFAIL1/ F4DCTI1/PMU1	[6]	nCTIIRQ_ISP	CPU1
		[5]	PMUIRQ_ISP	
		[4]	TMU	
		[3]	nCTIIRQ[1]	
		[2]	PMUIR[1]	
		[1]	PARITYFAILSC[1]	
		[0]	PARITYFAIL1	
INTG4	SYSMMU[7:0]	[7]	SYSMMU_2D[0]	System MMU
		[6]	SYSMMU_JPEG[0]	
		[5]	SYSMMU_FIMC3[0]	
		[4]	SYSMMU_FIMC2[0]	
		[3]	SYSMMU_FIMC1[0]	
		[2]	SYSMMU_FIMC0[0]	

Combiner Group ID	Combined Interrupt Source Name	Bit	Interrupt Source	Source Block
		[1]	SYSMMU_SSS[0]	
		[0]	SYSMMU_MDMA[0]	
INTG5	SYSMMU[15:8]	[7]	Reserved	System MMU
		[6]	SYSMMU_MFC_M1[0]	
		[5]	SYSMMU_MFC_M0[0]	
		[4]	SYSMMU_TV_M0[0]	
		[3]	Reserved	
		[2]	SYSMMU_LCD0_M0[0]	
		[1]	SYSMMU_GPS[0]	
		[0]	SYSMMU_ROTATOR[0]	
INTG6	SYSMMU[23:16]	[7]	SYSMMU_2D[1]	System MMU
		[6]	SYSMMU_JPEG[1]	
		[5]	SYSMMU_FIMC3[1]	
		[4]	SYSMMU_FIMC2[1]	
		[3]	SYSMMU_FIMC1[1]	
		[2]	SYSMMU_FIMC0[1]	
		[1]	SYSMMU_SSS[1]	
		[0]	SYSMMU_MDMA[1]	
INTG7	SYSMMU[31:24]	[7]	Reserved	System MMU
		[6]	SYSMMU_MFC_M1[1]	
		[5]	SYSMMU_MFC_M0[1]	
		[4]	SYSMMU_TV_M0[1]	
		[3]	Reserved	
		[2]	SYSMMU_LCD0_M0[1]	
		[1]	SYSMMU_GPS[1]	
		[0]	SYSMMU_ROTATOR[1]	
INTG8	PPMU [7:0]	[7]	PPMU_IMAGE_M0	PPMU
		[6]	PPMU_CAMIF_M0	
		[5]	PPMU_D_RIGHT_M0	
		[4]	PPMU_D_LEFT_M0	
		[3]	PPMU_ACP0_M0	
		[2]	PPMU_XIU_R_S1	
		[1]	PPMU_XIU_R	
		[0]	PPMU_XIU_L	
INTG9	PPMU [14:8]	[6]	PPMU_MFC_M1	PPMU
		[5]	PPMU_MFC_M0	

Combiner Group ID	Combined Interrupt Source Name	Bit	Interrupt Source	Source Block
		[4]	PPMU_3D	
		[3]	PPMU_TV_M0	
		[2]	PPMU_FILE_D_M0	
		[1]	PPMU_ISP_MX	
		[0]	PPMU_LCD0	
INTG10	DMC1/DMC0/MIU/ L2CACHE	[7]	DMC1_PPC_PEREV_M	DMC1
		[6]	DMC1_PPC_PEREV_A	
		[5]	DMC0_PPC_PEREV_M	DMC0
		[4]	DMC0_PPC_PEREV_A	
		[3]	ADC	General ADC
		[2]	L2CACHE	L2 Cache
		[1]	RP_TIMER	RP
		[0]	GPIO_AUDIO	Audio_SS
INTG11	LCD0	[3]	LCD0[3]	LCD0
		[2]	LCD0[2]	
		[1]	LCD0[1]	
		[0]	LCD0[0]	
INTG12	MCT/MIPI/UART4	[7]	G3	MCT
		[6]	G2	
		[5]	G1	
		[4]	G0	
		[1]	MIPI_HSI	MIPI
		[0]	UART4	UART
INTG13	CPU	[5]	CPU_nIRQOUT[0]	CPU
		[4]	Reserved	
		[3]	Reserved	
		[2]	Reserved	
		[1]	Reserved	
		[0]	Reserved	
INTG14	CPU	[6]	CPU_nIRQOUT[1]	CPU
		[5]	Reserved	
		[4]	Reserved	
		[3]	Reserved	
		[2]	Reserved	
		[1]	Reserved	
		[0]	Reserved	

Combiner Group ID	Combined Interrupt Source Name	Bit	Interrupt Source	Source Block
INTG15	BUS_ERROR/ SCUEVABORT	[7]	DECERRINTR	F4D
		[6]	SLVERRINTR	
		[5]	ERRRDINTR	
		[4]	ERRRTINTR	
		[3]	ERRWDINTR	
		[2]	ERRWTINTR	
		[1]	ECNTRINTR	
		[0]	SCUEVABORT	
INTG16	ISP	[7]	L3_IRQ	ISP
		[6]	Reserved	
		[5]	SYSMMU_ISP_CX[0]	
		[4]	SYSMMU_FIMC_FD[0]	
		[3]	SYSMMU_FIMC_DRC[0]	
		[2]	SYSMMU_FIMC_ISP[0]	
		[1]	SYSMMU_FIMC_LITE0[0]	
		[0]	SYSMMU_FIMC_LITE0[0]	
INTG17	ISP	[7]	L2_IRQ	ISP
		[6]	Reserved	
		[5]	SYSMMU_ISP_CX[1]	
		[4]	SYSMMU_FIMC_FD[1]	
		[3]	SYSMMU_FIMC_DRC[1]	
		[2]	SYSMMU_FIMC_ISP[1]	
		[1]	SYSMMU_FIMC_LITE0[1]	
		[0]	SYSMMU_FIMC_LITE0[1]	
INTG18	PARITYFAIL2/ F4DCTI2/ PMU2	[6]	CPU_nIRQOUT[2]	CPU
		[5]	PARITYFAILSCU[2]	
		[4]	PARITYFAIL2	
		[3]	nCTIIRQ[2]	
		[2]	PMUIRQ[2]	
		[1]	Reserved	
		[0]	L1_IRQ	
INTG19	PARITYFAIL3/ F4DCTI3/ PMU3	[6]	CPU_nIRQOUT[3]	CPU
		[5]	PARITYFAILSCU[3]	
		[4]	PARITYFAIL3	
		[3]	nCTIIRQ[3]	
		[2]	PMUIRQ[3]	

Combiner Group ID	Combined Interrupt Source Name	Bit	Interrupt Source	Source Block
		[1]	Reserved	
		[0]	L0_IRQ	

10.3 Functional Description

An interrupt enable bit controls an interrupt source in an interrupt group. IESRn and IECRn registers control the interrupt enable bits. IESRn register can toggle an interrupt bit to "1". If you write "1" to a bit position on IESRn, then it sets the corresponding bit on the interrupt enable bit to "1". However, IECRn register can toggle an interrupt enable bit to "0". If you write "1" to a bit position on IECRn, then it clears the corresponding bit on the interrupt enable bits to "0". This feature will make it easy to address resource sharing issues in a multi-processor system.

There are several interrupt sources in an interrupt group. If an interrupt enable bit is "0", then it masks the corresponding interrupt. All the interrupt sources in an interrupt group, including the masked interrupt sources, are ORed to form a combined interrupt request signal. The interrupt combiner connects the combined group interrupt request output to an input of a GIC.

You can show each interrupt source status before an interrupt-enable bit masks it by reading ISTRn register. You can show the combined group interrupt request output signal by reading CIPSR0 register.

10.4 Register Description

10.4.1 Register Map Summary

- Base Address: 0x1044_0000

Register	Offset	Description	Reset Value
Interrupt Combiner			
IESR0	0x0000	Interrupt enable set register for group 0 to 3	0x00000000
IECR0	0x0004	Interrupt enable clear register for group 0 to 3	0x00000000
ISTR0	0x0008	Interrupt status register for group 0 to 3	Undefined
IMSR0	0x000C	Interrupt masked status register for group 0 to 3	Undefined
IESR1	0x0010	Interrupt enable set register for group 4 to 7	0x00000000
IECR1	0x0014	Interrupt enable clear register for group 4 to 7	0x00000000
ISTR1	0x0018	Interrupt status register for group 4 to 7	Undefined
IMSR1	0x001C	Interrupt masked status register for group 4 to 7	Undefined
IESR2	0x0020	Interrupt enable set register for group 8 to 11	0x00000000
IECR2	0x0024	Interrupt enable clear register for group 8 to 11	0x00000000
ISTR2	0x0028	Interrupt status register for group 8 to 11	Undefined
IMSR2	0x002C	Interrupt masked status register for group 8 to 11	Undefined
IESR3	0x0030	Interrupt enable set register for group 12 to 15	0x00000000
IECR3	0x0034	Interrupt enable clear register for group 12 to 15	0x00000000
ISTR3	0x0038	Interrupt masked status register for group 12 to 15	Undefined
IMSR3	0x003C	Interrupt status register for group 16 to 17	Undefined
IESR4	0x0040	Interrupt enable set register for group 16 to 17	0x00000000
IECR4	0x0044	Interrupt enable clear register for group 16 to 17	0x00000000
ISTR4	0x0048	Interrupt masked status register for group 16 to 17	Undefined
IMSR4	0x004C	Interrupt status register for group 16 to 17	Undefined
CIPSR0	0x0100	Combined interrupt pending status0	Undefined

10.4.2 Interrupt Combiner

10.4.2.1 IESR0

- Base Address: 0x1044_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
nCTIIRQ_ISP	[30]	RW	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1" Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
PMUIRQ_ISP	[29]	RW		0
TMU	[28]	RW		0
nCTIIRQ[1]	[27]	RW		0
PMUIR[1]	[26]	RW		0
PARITYFAILSC[1]	[25]	RW		0
PARITYFAIL1	[24]	RW		0
RSVD	[23]	–	Reserved	0x0
PARRINTR	[22]	RW	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
PARRDINTR	[21]	RW		0
TMU	[20]	RW		0
nCTIIRQ[0]	[19]	RW		0
PMUIRQ[0]	[18]	RW		0
PARITYFAILSCU[0]	[17]	RW		0
PARITYFAIL0	[16]	RW		0
RSVD	[15:12]	–	Reserved	0x0
TZASC1[1]	[11]	RW	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit 0 = Masks. 1 = Enables.	0
TZASC1[0]	[10]	RW		0
TZASC0[1]	[9]	RW		0
TZASC0[0]	[8]	RW		0
RSVD	[7:4]	–	Reserved	0x0
MDNIE_LCD0[3]	[3]	RW	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
MDNIE_LCD0[2]	[2]	RW		0
MDNIE_LCD0[1]	[1]	RW		0
MDNIE_LCD0[0]	[0]	RW		0

10.4.2.2 IECR0

- Base Address: 0x1044_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
nCTIIRQ_ISP	[30]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
PMUIRQ_ISP	[29]	RW		0
TMU	[28]	RW		0
nCTIIRQ[1]	[27]	RW		0
PMUIR[1]	[26]	RW		0
PARITYFAILSC[1]	[25]	RW		0
PARITYFAIL1	[24]	RW		0
RSVD	[23]	–	Reserved	0x0
PARRINTR	[22]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
PARRDINTR	[21]	RW		0
TMU	[20]	RW		0
nCTIIRQ[0]	[19]	RW		0
PMUIRQ[0]	[18]	RW		0
PARITYFAILSCU[0]	[17]	RW		0
PARITYFAIL0	[16]	RW		0
RSVD	[15:12]	–	Reserved	0x0
TZASC1[1]	[11]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
TZASC1[0]	[10]	RW		0
TZASC0[1]	[9]	RW		0
TZASC0[0]	[8]	RW		0
RSVD	[7:4]	–	Reserved	0x0
MDNIE_LCD0[3]	[3]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
MDNIE_LCD0[2]	[2]	RW		0
MDNIE_LCD0[1]	[1]	RW		0
MDNIE_LCD0[0]	[0]	RW		0

10.4.2.3 ISTR0

- Base Address: 0x1044_0000
- Address = Base Address + 0x0008, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
nCTIIRQ_ISP	[30]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
PMUIRQ_ISP	[29]	R		–
TMU	[28]	R		–
nCTIIRQ[1]	[27]	R		–
PMUIR[1]	[26]	R		–
PARITYFAILSC[1]	[25]	R		–
PARITYFAIL1	[24]	R		–
RSVD	[23]	–	Reserved	0x0
PARRINTR	[22]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
PARRDINTR	[21]	R		–
TMU	[20]	R		–
nCTIIRQ[0]	[19]	R		–
PMUIRQ[0]	[18]	R		–
PARITYFAILSCU[0]	[17]	R		–
PARITYFAIL0	[16]	R		–
RSVD	[15:12]	–	Reserved	0x0
TZASC1[1]	[11]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
TZASC1[0]	[10]	R		–
TZASC0[1]	[9]	R		–
TZASC0[0]	[8]	R		–
RSVD	[7:4]	–	Reserved	0x0
MDNIE_LCD0[3]	[3]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
MDNIE_LCD0[2]	[2]	R		–
MDNIE_LCD0[1]	[1]	R		–
MDNIE_LCD0[0]	[0]	R		–

10.4.2.4 IMSR0

- Base Address: 0x1044_0000
- Address = Base Address + 0x000C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
nCTIIRQ_ISP	[30]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
PMUIRQ_ISP	[29]	R		–
TMU	[28]	R		–
nCTIIRQ[1]	[27]	R		–
PMUIR[1]	[26]	R		–
PARITYFAILSC[1]	[25]	R		–
PARITYFAIL1	[24]	R		–
RSVD	[23]	–	Reserved	0x0
PARRINTR	[22]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
PARRDINTR	[21]	R		–
TMU	[20]	R		–
nCTIIRQ[0]	[19]	R		–
PMUIRQ[0]	[18]	R		–
PARITYFAILSCU[0]	[17]	R		–
PARITYFAIL0	[16]	R		–
RSVD	[15:12]	–	Reserved	0x0
TZASC1[1]	[11]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
TZASC1[0]	[10]	R		–
TZASC0[1]	[9]	R		–
TZASC0[0]	[8]	R		–
RSVD	[7:4]	–	Reserved	0x0
MDNIE_LCD0[3]	[3]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
MDNIE_LCD0[2]	[2]	R		–
MDNIE_LCD0[1]	[1]	R		–
MDNIE_LCD0[0]	[0]	R		–

10.4.2.5 IESR1

- Base Address: 0x1044_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
SYSMMU_MFC_M1[1]	[30]	RW		0
SYSMMU_MFC_M0[1]	[29]	RW		0
SYSMMU_TV_M0[1]	[28]	RW		0
RSVD	[27]	–		0
SYSMMU_LCD0_M0[1]	[26]	RW		0
SYSMMU_GPS[1]	[25]	RW		0
SYSMMU_ROTATOR[1]	[24]	RW		0
SYSMMU_2D[1]	[23]	RW	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
SYSMMU_JPEG[1]	[22]	RW		0
SYSMMU_FIMC3[1]	[21]	RW		0
SYSMMU_FIMC2[1]	[20]	RW		0
SYSMMU_FIMC1[1]	[19]	RW		0
SYSMMU_FIMC0[1]	[18]	RW		0
SYSMMU_SSS[1]	[17]	RW		0
SYSMMU_MDMA[1]	[16]	RW		0
RSVD	[15]	–	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
SYSMMU_MFC_M1[0]	[14]	RW		0
SYSMMU_MFC_M0[0]	[13]	RW		0
SYSMMU_TV_M0[0]	[12]	RW		0
RSVD	[11]	–		0
SYSMMU_LCD0_M0[0]	[10]	RW		0
SYSMMU_GPS[0]	[9]	RW		0
SYSMMU_ROTATOR[0]	[8]	RW		0
SYSMMU_2D[0]	[7]	RW	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
SYSMMU_JPEG[0]	[6]	RW		0
SYSMMU_FIMC3[0]	[5]	RW		0
SYSMMU_FIMC2[0]	[4]	RW		0
SYSMMU_FIMC1[0]	[3]	RW		0
SYSMMU_FIMC0[0]	[2]	RW		0
SYSMMU_SSS[0]	[1]	RW		0
SYSMMU_MDMA[0]	[0]	RW		0

10.4.2.6 IECR1

- Base Address: 0x1044_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
SYSMMU_MFC_M1[1]	[30]	RW		0
SYSMMU_MFC_M0[1]	[29]	RW		0
SYSMMU_TV_M0[1]	[28]	RW		0
RSVD	[27]	–		0
SYSMMU_LCD0_M0[1]	[26]	RW		0
SYSMMU_GPS[1]	[25]	RW		0
SYSMMU_ROTATOR[1]	[24]	RW		0
SYSMMU_2D[1]	[23]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
SYSMMU_JPEG[1]	[22]	RW		0
SYSMMU_FIMC3[1]	[21]	RW		0
SYSMMU_FIMC2[1]	[20]	RW		0
SYSMMU_FIMC1[1]	[19]	RW		0
SYSMMU_FIMC0[1]	[18]	RW		0
SYSMMU_SSS[1]	[17]	RW		0
SYSMMU_MDMA[1]	[16]	RW		0
RSVD	[15]	–	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
SYSMMU_MFC_M1[0]	[14]	RW		0
SYSMMU_MFC_M0[0]	[13]	RW		0
SYSMMU_TV_M0[0]	[12]	RW		0
RSVD	[11]	–		0
SYSMMU_LCD0_M0[0]	[10]	RW		0
SYSMMU_GPS[0]	[9]	RW		0
SYSMMU_ROTATOR[0]	[8]	RW		0
SYSMMU_2D[0]	[7]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
SYSMMU_JPEG[0]	[6]	RW		0
SYSMMU_FIMC3[0]	[5]	RW		0
SYSMMU_FIMC2[0]	[4]	RW		0
SYSMMU_FIMC1[0]	[3]	RW		0
SYSMMU_FIMC0[0]	[2]	RW		0
SYSMMU_SSS[0]	[1]	RW		0
SYSMMU_MDMA[0]	[0]	RW		0

10.4.2.7 ISTR1

- Base Address: 0x1044_0000
- Address = Base Address + 0x0018, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
SYSMMU_MFC_M1[1]	[30]	R		–
SYSMMU_MFC_M0[1]	[29]	R		–
SYSMMU_TV_M0[1]	[28]	R		–
RSVD	[27]	–		–
SYSMMU_LCD0_M0[1]	[26]	R		–
SYSMMU_GPS[1]	[25]	R		–
SYSMMU_ROTATOR[1]	[24]	R		–
SYSMMU_2D[1]	[23]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
SYSMMU_JPEG[1]	[22]	R		–
SYSMMU_FIMC3[1]	[21]	R		–
SYSMMU_FIMC2[1]	[20]	R		–
SYSMMU_FIMC1[1]	[19]	R		–
SYSMMU_FIMC0[1]	[18]	R		–
SYSMMU_SSS[1]	[17]	R		–
SYSMMU_MDMA[1]	[16]	R		–
RSVD	[15]	–	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
SYSMMU_MFC_M1[0]	[14]	R		–
SYSMMU_MFC_M0[0]	[13]	R		–
SYSMMU_TV_M0[0]	[12]	R		–
RSVD	[11]	–		–
SYSMMU_LCD0_M0[0]	[10]	R		–
SYSMMU_GPS[0]	[9]	R		–
SYSMMU_ROTATOR[0]	[8]	R		–
SYSMMU_2D[0]	[7]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
SYSMMU_JPEG[0]	[6]	R		–
SYSMMU_FIMC3[0]	[5]	R		–
SYSMMU_FIMC2[0]	[4]	R		–
SYSMMU_FIMC1[0]	[3]	R		–
SYSMMU_FIMC0[0]	[2]	R		–
SYSMMU_SSS[0]	[1]	R		–
SYSMMU_MDMA[0]	[0]	R		–

10.4.2.8 IMSR1

- Base Address: 0x1044_0000
- Address = Base Address + 0x001C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads out as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
SYSMMU_MFC_M1[1]	[30]	R		–
SYSMMU_MFC_M0[1]	[29]	R		–
SYSMMU_TV_M0[1]	[28]	R		–
RSVD	[27]	–		–
SYSMMU_LCD0_M0[1]	[26]	R		–
SYSMMU_GPS[1]	[25]	R		–
SYSMMU_ROTATOR[1]	[24]	R		–
SYSMMU_2D[1]	[23]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads out as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
SYSMMU_JPEG[1]	[22]	R		–
SYSMMU_FIMC3[1]	[21]	R		–
SYSMMU_FIMC2[1]	[20]	R		–
SYSMMU_FIMC1[1]	[19]	R		–
SYSMMU_FIMC0[1]	[18]	R		–
SYSMMU_SSS[1]	[17]	R		–
SYSMMU_MDMA[1]	[16]	R		–
RSVD	[15]	–	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads out as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
SYSMMU_MFC_M1[0]	[14]	R		–
SYSMMU_MFC_M0[0]	[13]	R		–
SYSMMU_TV_M0[0]	[12]	R		–
RSVD	[11]	–		–
SYSMMU_LCD0_M0[0]	[10]	R		–
SYSMMU_GPS[0]	[9]	R		–
SYSMMU_ROTATOR[0]	[8]	R		–
SYSMMU_2D[0]	[7]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads out as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
SYSMMU_JPEG[0]	[6]	R		–
SYSMMU_FIMC3[0]	[5]	R		–
SYSMMU_FIMC2[0]	[4]	R		–
SYSMMU_FIMC1[0]	[3]	R		–
SYSMMU_FIMC0[0]	[2]	R		–
SYSMMU_SSS[0]	[1]	R		–
SYSMMU_MDMA[0]	[0]	R		–

10.4.2.9 IESR2

- Base Address: 0x1044_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
LCD0[3]	[27]	RW	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
LCD0[2]	[26]	RW		0
LCD0[1]	[25]	RW		0
LCD0[0]	[24]	RW		0
DMC1_PPC_PEREV_M	[23]	RW	Set the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
DMC1_PPC_PEREV_A	[22]	RW		0
DMC0_PPC_PEREV_M	[21]	RW		0
DMC0_PPC_PEREV_A	[20]	RW		0
ADC	[19]	RW		0
L2CACHE	[18]	RW		0
RP_TIMER	[17]	RW		0
GPIO_AUDIO	[16]	–		0
RSVD	[15]	–	Reserved	0
PPMU_MFC_M1	[14]	RW	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 =Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
PPMU_MFC_M0	[13]	RW		0
PPMU_3D	[12]	RW		0
PPMU_TV_M0	[11]	RW		0
PPMU_FILE_D_M0	[10]	RW		0
PPMU_ISP_MX	[9]	RW		0
PPMU_LCD0	[8]	RW		0
PPMU_IMAGE_M0	[7]	RW	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
PPMU_CAMIF_M0	[6]	RW		0
PPMU_D_RIGHT_M0	[5]	RW		0
PPMU_D_LEFT_M0	[4]	RW		0
PPMU_ACP0_M0	[3]	RW		0
PPMU_XIU_R_S1	[2]	RW		0
PPMU_XIU_R	[1]	RW		0
PPMU_XIU_L	[0]	RW		0

10.4.2.10 IECR2

- Base Address: 0x1044_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
LCD0[3]	[27]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
LCD0[2]	[26]	RW		0
LCD0[1]	[25]	RW		0
LCD0[0]	[24]	RW		0
DMC1_PPC_PEREV_M	[23]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
DMC1_PPC_PEREV_A	[22]	RW		0
DMC0_PPC_PEREV_M	[21]	RW		0
DMC0_PPC_PEREV_A	[20]	RW		0
ADC	[19]	RW		0
L2CACHE	[18]	RW		0
RP_TIMER	[17]	RW		0
GPIO_AUDIO	[16]	RW		0
RSVD	[15]	–	Reserved	0
PPMU_MFC_M1	[14]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
PPMU_MFC_M0	[13]	RW		0
PPMU_3D	[12]	RW		0
PPMU_TV_M0	[11]	RW		0
PPMU_FILE_D_M0	[10]	RW		0
PPMU_ISP_MX	[9]	RW		0
PPMU_LCD0	[8]	RW		0
PPMU_IMAGE_M0	[7]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
PPMU_CAMIF_M0	[6]	RW		0
PPMU_D_RIGHT_M0	[5]	RW		0
PPMU_D_LEFT_M0	[4]	RW		0
PPMU_ACP0_M0	[3]	RW		0
PPMU_XIU_R_S1	[2]	RW		0
PPMU_XIU_R	[1]	RW		0
PPMU_XIU_L	[0]	RW		0

10.4.2.11 ISTR2

- Base Address: 0x1044_0000
- Address = Base Address + 0x0028, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
LCD0[3]	[27]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
LCD0[2]	[26]	R		–
LCD0[1]	[25]	R		–
LCD0[0]	[24]	R		–
DMC1_PPC_PEREV_M	[23]	R	Interrupts pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
DMC1_PPC_PEREV_A	[22]	R		–
DMC0_PPC_PEREV_M	[21]	R		–
DMC0_PPC_PEREV_A	[20]	R		–
ADC	[19]	R		–
L2CACHE	[18]	R		–
RP_TIMER	[17]	R		–
GPIO_AUDIO	[16]	R		0
RSVD	[15]	–	Reserved	0
PPMU_MFC_M1	[14]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
PPMU_MFC_M0	[13]	R		–
PPMU_3D	[12]	R		–
PPMU_TV_M0	[11]	R		–
PPMU_FILE_D_M0	[10]	R		–
PPMU_ISP_MX	[9]	R		–
PPMU_LCD0	[8]	R		–
PPMU_IMAGE_M0	[7]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
PPMU_CAMIF_M0	[6]	R		–
PPMU_D_RIGHT_M0	[5]	R		–
PPMU_D_LEFT_M0	[4]	R		–
PPMU_ACP0_M0	[3]	R		–
PPMU_XIU_R_S1	[2]	R		–
PPMU_XIU_R	[1]	R		–
PPMU_XIU_L	[0]	R		–

10.4.2.12 IMSR2

- Base Address: 0x1044_0000
- Address = Base Address + 0x002C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0x0
LCD0[3]	[27]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads out as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
LCD0[2]	[26]	R		–
LCD0[1]	[25]	R		–
LCD0[0]	[24]	R		–
DMC1_PPC_PEREV_M	[23]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
DMC1_PPC_PEREV_A	[22]	R		–
DMC0_PPC_PEREV_M	[21]	R		–
DMC0_PPC_PEREV_A	[20]	R		–
ADC	[19]	R		–
L2CACHE	[18]	R		–
RP_TIMER	[17]	R		–
GPIO_AUDIO	[16]	R		0
RSVD	[15]	–	Reserved	0
PPMU_MFC_M1	[14]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
PPMU_MFC_M0	[13]	R		–
PPMU_3D	[12]	R		–
PPMU_TV_M0	[11]	R		–
PPMU_FILE_D_M0	[10]	R		–
PPMU_ISP_MX	[9]	R		–
PPMU_LCD0	[8]	R		–
PPMU_IMAGE_M0	[7]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
PPMU_CAMIF_M0	[6]	R		–
PPMU_D_RIGHT_M0	[5]	R		–
PPMU_D_LEFT_M0	[4]	R		–
PPMU_ACP0_M0	[3]	R		–
PPMU_XIU_R_S1	[2]	R		–
PPMU_XIU_R	[1]	R		–
PPMU_XIU_L	[0]	R		–

10.4.2.13 IESR3

- Base Address: 0x1044_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DECERRINTR	[31]	RW	Sets the corresponding interrupt enable bit to "1 If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
SLVERRINTR	[30]	RW		0
ERRRDINTR	[29]	RW		0
ERRRTINTR	[28]	RW		0
ERRWDINTR	[27]	RW		0
ERRWTINTR	[26]	RW		0
ECNTRINTR	[25]	RW		0
SCUEVABORT	[24]	RW		0
RSVD	[23]	–	Sets the corresponding interrupt enable bit to "1 If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
CPU_nIRQOUT_1	[22]	RW		0
RSVD	[21]	–		0
RSVD	[20]	–		0
RSVD	[19]	–		0
RSVD	[18]	–		0
RSVD	[17]	–		0
RSVD	[16]	–		0
RSVD	[15:14]	–	Sets the corresponding interrupt enable bit to "1 If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	00
CPU_nIRQOUT_0	[13]	RW		0
RSVD	[12]	–		0
RSVD	[11]	–		0
RSVD	[10]	–		0
RSVD	[9]	–		0
RSVD	[8]	–		0
MCT_G3	[7]	RW	Sets the corresponding interrupt enable bit to "1 If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
MCT_G2	[6]	RW		0
MCT_G1	[5]	RW		0
MCT_G0	[4]	RW		0
RSVD	[3:2]	–		0x0
MIPI_HSI]	[1]	RW		0
UART4	[0]	RW		0

10.4.2.14 IECR3

- Base Address: 0x1044_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DECERRINTR	[31]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
SLVERRINTR	[30]	RW		0
ERRRDINTR	[29]	RW		0
ERRRTINTR	[28]	RW		0
ERRWDINTR	[27]	RW		0
ERRWTINTR	[26]	RW		0
ECNTRINTR	[25]	RW		0
SCUEVABORT	[24]	RW		0
RSVD	[23]	–	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
CPU_nIRQOUT_1	[22]	RW		0
RSVD	[21]	–		0
RSVD	[20]	–		0
RSVD	[19]	–		0
RSVD	[18]	–		0
RSVD	[17]	–		0
RSVD	[16]	–		0
RSVD	[15:14]	–	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	00
CPU_nIRQOUT_0	[13]	RW		0
RSVD	[12]	–		0
RSVD	[11]	–		0
RSVD	[10]	–		0
RSVD	[9]	–		0
RSVD	[8]	–		0
MCT_G3	[7]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
MCT_G2	[6]	RW		0
MCT_G1	[5]	RW		0
MCT_G0	[4]	RW		0
RSVD	[3:2]	RW		00
MIPI_HSI]	[1]	RW		0
UART4	[0]	RW		0
LCD1[0]	[0]	RW		0

10.4.2.15 ISTR3

- Base Address: 0x1044_0000
- Address = Base Address + 0x0038, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
DECERRINTR	[31]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
SLVERRINTR	[30]	R		–
ERRRDINTR	[29]	R		–
ERRRTINTR	[28]	R		–
ERRWDINTR	[27]	R		–
ERRWTINTR	[26]	R		–
ECNTRINTR	[25]	R		–
SCUEVABORT	[24]	R		–
RSVD	[23]	–	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	0
CPU_nIRQOUT_1	[22]	R		–
RSVD	[21]	–		–
RSVD	[20]	–		–
RSVD	[19]	–		–
RSVD	[18]	–		–
RSVD	[17]	–		–
RSVD	[16]	–		–
RSVD	[15:14]	–	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	00
CPU_nIRQOUT_0	[13]	R		–
RSVD	[12]	–		–
RSVD	[11]	–		–
RSVD	[10]	–		–
RSVD	[9]	–		–
RSVD	[8]	–		–
MCT_G3	[7]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	0x0
MCT_G2	[6]	R		
MCT_G1	[5]	R		
MCT_G0	[4]	R		
RSVD	[3:2]	–		–
MIPI_HSI]	[1]	R		–
UART4	[0]	R		–
LCD1[0]	[0]	R		–

10.4.2.16 IMSR3

- Base Address: 0x1044_0000
- Address = Base Address + 0x003C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
DECERRINTR	[31]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	–
SLVERRINTR	[30]	R		–
ERRRDINTR	[29]	R		–
ERRRTINTR	[28]	R		–
ERRWDINTR	[27]	R		–
ERRWTINTR	[26]	R		–
ECNTRINTR	[25]	R		–
SCUEVABORT	[24]	R		–
RSVD	[23]	–	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	0
CPU_nIRQOUT_1	[22]	R		–
RSVD	[21]	–		–
RSVD	[20]	–		–
RSVD	[19]	–		–
RSVD	[18]	–		–
RSVD	[17]	–		–
RSVD	[16]	–		–
RSVD	[15:14]	–	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	00
CPU_nIRQOUT_0	[13]	R		–
RSVD	[12]	–		–
RSVD	[11]	–		–
RSVD	[10]	–		–
RSVD	[9]	–		–
RSVD	[8]	–		–
MCT_G3	[7]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	-
MCT_G2	[6]	R		-
MCT_G1	[5]	R		-
MCT_G0	[4]	R		-
RSVD	[3:2]	-		00
MIPI_HSI]	[1]	R		–
UART4	[0]	R		–
LCD1[0]	[0]	R		–

10.4.2.17 IESR4

- Base Address: 0x1044_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
CPU_nIRQOUT_3	[30]	RW		0
PARITYFAILSCU3	[29]	RW		0
PARITYFAIL3	[28]	RW		0
CPU_nCTIIRQ_3	[27]	RW		0
CPU_PMUIRQ_3	[26]	RW		0
RSVD	[25]	RW		0
MCT_L0	[24]	RW		0
RSVD	[23]	–	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
CPU_nIRQOUT_2	[22]	RW		0
PARITYFAILSCU2	[21]	RW		0
PARITYFAIL2	[20]	RW		0
CPU_nCTIIRQ_2	[19]	RW		0
CPU_PMUIRQ_2	[18]	RW		0
RSVD	[17]	RW		0
MCT_L1	[16]	RW		0
MCT_L2	[15]	RW	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
RSVD	[14]	–		0
SYSMMU_ISP_CX[1]	[13]	RW		0
SYSMMU_FIMC_FD[1]	[12]	RW		0
SYSMMU_FIMC_DRC[1]	[11]	RW		0
SYSMMU_FIMC_ISP[1]	[10]	RW		0
SYSMMU_FIMC_LITE0[1]	[9]	RW		0
SYSMMU_FIMC_LITE0[1]	[8]	RW		0
MCT_L3	[7]	RW	Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
RSVD	[6]	–		0
SYSMMU_ISP_CX[0]	[5]	RW		0
SYSMMU_FIMC_FD[0]	[4]	RW		0
SYSMMU_FIMC_DRC[0]	[3]	RW		0
SYSMMU_FIMC_ISP[0]	[2]	RW		0
SYSMMU_FIMC_LITE0[0]	[1]	RW		0
SYSMMU_FIMC_LITE0[0]	[0]	RW		0

10.4.2.18 IECR4

- Base Address: 0x1044_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
CPU_nIRQOUT_3	[30]	RW		0
PARITYFAILSCU3	[29]	RW		0
PARITYFAIL3	[28]	RW		0
CPU_nCTIIRQ_3	[27]	RW		0
CPU_PMUIRQ_3	[26]	RW		0
RSVD	[25]	RW		0
MCT_L0	[24]	RW		0
RSVD	[23]	–	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
CPU_nIRQOUT_2	[22]	RW		0
PARITYFAILSCU2	[21]	RW		0
PARITYFAIL2	[20]	RW		0
CPU_nCTIIRQ_2	[19]	RW		0
CPU_PMUIRQ_2	[18]	RW		0
RSVD	[17]	RW		0
MCT_L1	[16]	RW		0
MCT_L2	[15]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
RSVD	[14]	–		0
SYSMMU_ISP_CX[1]	[13]	RW		0
SYSMMU_FIMC_FD[1]	[12]	RW		0
SYSMMU_FIMC_DRC[1]	[11]	RW		0
SYSMMU_FIMC_ISP[1]	[10]	RW		0
SYSMMU_FIMC_LITE0[1]	[9]	RW		0
SYSMMU_FIMC_LITE0[1]	[8]	RW		0
MCT_L3	[7]	RW	Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables.	0
RSVD	[6]	–		0
SYSMMU_ISP_CX[0]	[5]	RW		0
SYSMMU_FIMC_FD[0]	[4]	RW		0
SYSMMU_FIMC_DRC[0]	[3]	RW		0
SYSMMU_FIMC_ISP[0]	[2]	RW		0
SYSMMU_FIMC_LITE0[0]	[1]	RW		0
SYSMMU_FIMC_LITE0[0]	[0]	RW		0

10.4.2.19 ISTR4

- Base Address: 0x1044_0000
- Address = Base Address + 0x0048, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	0
CPU_nIRQOUT_3	[30]	R		0
PARITYFAILSCU3	[29]	R		0
PARITYFAIL3	[28]	R		0
CPU_nCTIIRQ_3	[27]	R		0
CPU_PMUIRQ_3	[26]	R		0
RSVD	[25]	R		0
MCT_L0	[24]	R		0
RSVD	[23]	–	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	0
CPU_nIRQOUT_2	[22]	R		0
PARITYFAILSCU2	[21]	R		0
PARITYFAIL2	[20]	R		0
CPU_nCTIIRQ_2	[19]	R		0
CPU_PMUIRQ_2	[18]	R		0
RSVD	[17]	R		0
MCT_L1	[16]	R		0
MCT_L2	[15]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	0
RSVD	[14]	–		0
SYSMMU_ISP_CX[1]	[13]	R		0
SYSMMU_FIMC_FD[1]	[12]	R		0
SYSMMU_FIMC_DRC[1]	[11]	R		0
SYSMMU_FIMC_ISP[1]	[10]	R		0
SYSMMU_FIMC_LITE0[1]	[9]	R		0
SYSMMU_FIMC_LITE0[1]	[8]	R		0
MCT_L3	[7]	R	Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending.	0
RSVD	[6]	–		0
SYSMMU_ISP_CX[0]	[5]	R		0
SYSMMU_FIMC_FD[0]	[4]	R		0
SYSMMU_FIMC_DRC[0]	[3]	R		0
SYSMMU_FIMC_ISP[0]	[2]	R		0
SYSMMU_FIMC_LITE0[0]	[1]	R		0
SYSMMU_FIMC_LITE0[0]	[0]	R		0

10.4.2.20 IMSR4

- Base Address: 0x1044_0000
- Address = Base Address + 0x004C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	0
CPU_nIRQOUT_3	[30]	R		0
PARITYFAILSCU3	[29]	R		0
PARITYFAIL3	[28]	R		0
CPU_nCTIIRQ_3	[27]	R		0
CPU_PMUIRQ_3	[26]	R		0
RSVD	[25]	R		0
MCT_L0	[24]	R		0
RSVD	[23]	–	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	0
CPU_nIRQOUT_2	[22]	R		0
PARITYFAILSCU2	[21]	R		0
PARITYFAIL2	[20]	R		0
CPU_nCTIIRQ_2	[19]	R		0
CPU_PMUIRQ_2	[18]	R		0
RSVD	[17]	R		0
MCT_L1	[16]	R		0
MCT_L2	[15]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	0
RSVD	[14]	–		0
SYSMMU_ISP_CX[1]	[13]	R		0
SYSMMU_FIMC_FD[1]	[12]	R		0
SYSMMU_FIMC_DRC[1]	[11]	R		0
SYSMMU_FIMC_ISP[1]	[10]	R		0
SYSMMU_FIMC_LITE0[1]	[9]	R		0
SYSMMU_FIMC_LITE0[1]	[8]	R		0
MCT_L3	[7]	R	Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending.	0
RSVD	[6]	–		0
SYSMMU_ISP_CX[0]	[5]	R		0
SYSMMU_FIMC_FD[0]	[4]	R		0
SYSMMU_FIMC_DRC[0]	[3]	R		0
SYSMMU_FIMC_ISP[0]	[2]	R		0
SYSMMU_FIMC_LITE0[0]	[1]	R		0
SYSMMU_FIMC_LITE0[0]	[0]	R		0

10.4.2.21 CIPSR0

- Base Address: 0x1044_0000
- Address = Base Address + 0x0100, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
INTG31	[31]	–	Reserved	–
INTG30	[30]	–		–
INTG29	[29]	–		–
INTG28	[28]	–		–
INTG27	[27]	–		–
INTG26	[26]	–		–
INTG25	[25]	–		–
INTG24	[24]	–		–
INTG23	[23]	–		–
INTG22	[22]	–		–
INTG21	[21]	–		–
INTG20	[20]	–		–
INTG19	[19]	R	Combined interrupt pending status. 0 = The combined interrupt is not pending. 1 = The combined interrupt is pending. This means the corresponding interrupt request to the GIC is asserted.	–
INTG18	[18]	R		–
INTG17	[17]	R		–
INTG16	[16]	R		–
INTG15	[15]	R		–
INTG14	[14]	R		–
INTG13	[13]	R		–
INTG12	[12]	R		–
INTG11	[11]	R		–
INTG10	[10]	R		–
INTG9	[9]	R		–
INTG8	[8]	R		–
INTG7	[7]	R		–
INTG6	[6]	R		–
INTG5	[5]	R		–
INTG4	[4]	R		–
INTG3	[3]	R		–
INTG2	[2]	R		–
INTG1	[1]	R		–
INTG0	[0]	R		–

11 Direct Memory Access Controller (DMAC)

This chapter includes:

- Overview of DMA Controller
- Register description
- Instruction

11.1 Overview of DMA Controller

The two Direct Memory Access (DMA) tops that Exynos 4412 SCP supports:

- Memory-to-Memory (M2M) transfer (DMA_mem)
- Peripheral-to-memory transfer and vice-versa (DMA_peri)

The DMA_mem consists of one PL330 (DMA) and some logics. DMA_peri consists of two PL330s (DMA0 and DMA1) and dma_map.

Figure 11-1 illustrates the two DMA tops.

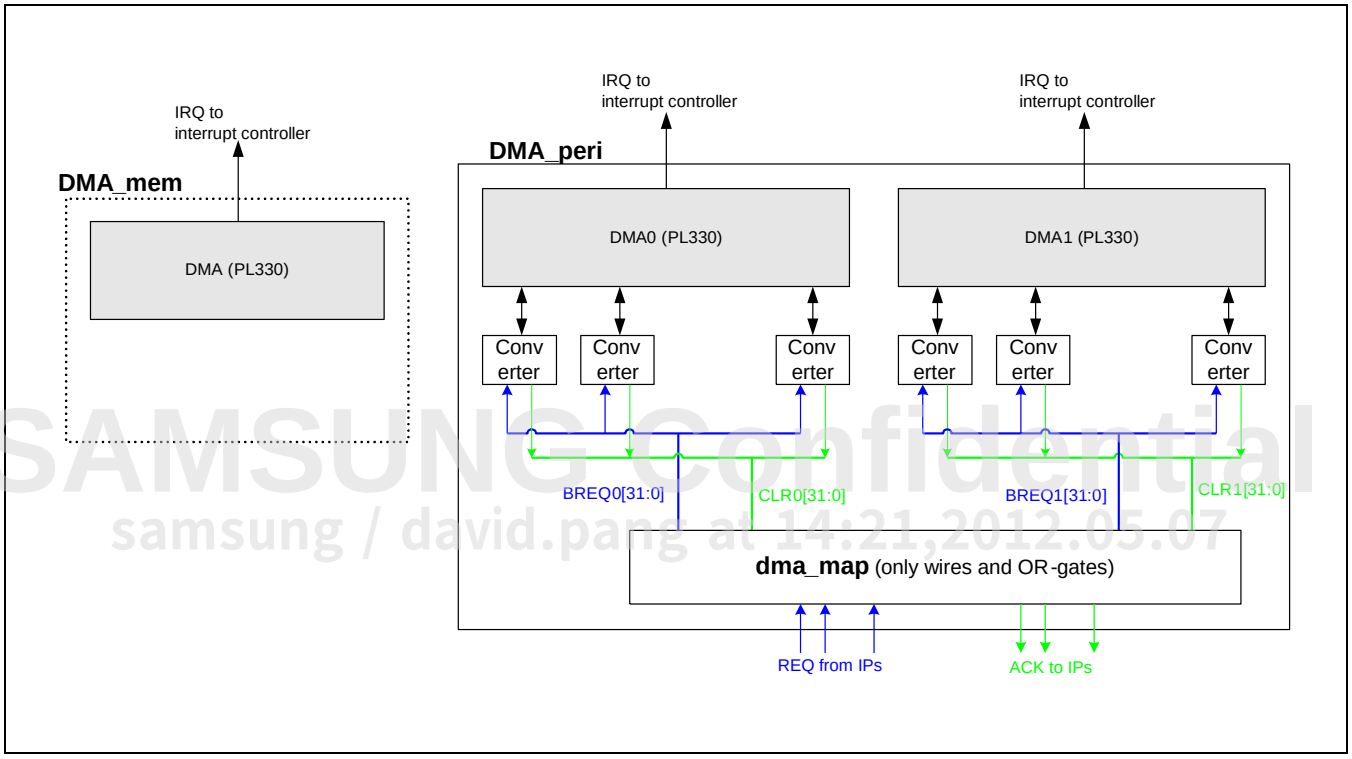


Figure 11-1 Two DMA Tops

You should set all the peripherals as non-secure in TrustZone Protection Controller (TZPC) module since DMA_peri operates only as non-secure.

The attributes that the DMA_mem DMA Controllers have:

- DMA_mem accesses memory through the AXI_IMGX bus and is located in IMG block.
- DMA_mem supports only the secured AXI transaction.

11.1.1 Features of DMA Controller

Table 11-1 describes the features of DMA Controller. Refer to this table for DMA and for writing DMA assembly code.

Table 11-1 Features of DMA Controller

Key Features	DMA_mem	DMA_peri
Supports data size	Up to double word (64-bit)	Up to word (32-bit)
Supports burst size	Up to 16 burst	Up to 16 burst
Supports channel	8 channels at the same time	16 channels at the same time

Each DMA module has 32 interrupt sources. However, you should send only one interrupt to Interrupt Controller.

Table 11-2 describes the DMA request mapping.

Table 11-2 DMA Request Mapping Table

Module	No.	
Peri DMA1	31	Reserved
	30	Reserved
	29	MIPI_HSI7
	28	MIPI_HSI6
	27	SPDIF
	26	Siimbus0AUX_TX
	25	Siimbus0AUX_RX
	24	Siimbus5_RX
	23	Siimbus5_RX
	22	Siimbus3_TX
	21	Siimbus3_RX
	20	Slimbus1_TX
	19	Slimbus1_RX
	18	UART3_TX
	17	UART3_RX
	16	UART1_TX
	15	UART1_RX
	14	UART0_TX
	13	UART0_RX
	12	I2S1_TX
	11	I2S1_RX
	10	I2S0_TX
	9	I2S0_RX

Module	No.	
	8	I2S0S_TX
	7	SPI1_TX
	6	SPI1_RX
	5	MIPI_HSI3
	4	MIPI_HSI2
	3	PCM1_TX
	2	PCM1_RX
	1	PCM0_TX
	0	PCM0_RX
Peri DMA0	31	MIPI_HSI5
	30	MIPI_HSI4
	29	AC_PCMout
	28	AC_PCMin
	27	AC_MICin
	26	SlimBUS4_TX
	25	SlimBUS4_RX
	24	SlimBUS2_TX
	23	SlimBUS2_RX
	22	SlimBUS0_TX
	21	SlimBUS0_RX
	20	UART4_TX
	19	UART4_RX
	18	UART2_TX
	17	UART2_RX
	16	UART0_TX
	15	UART0_RX
	14	I2S2_TX
	13	I2S2_RX
	12	I2S0_TX
	11	I2S0_RX
	10	I2S0S_TX
	9	SPI2_TX
	8	SPI2_RX
	7	SPI0_TX
	6	SPI0_RX
	5	MIPI_HSI1
	4	MIPI_HSI0

Module	No.	
	3	PCM2_TX
	2	PCM2_RX
	1	PCM0_TX
	0	PCM0_RX
DMA_mem	–	–

Caution: When you enable PDMA0 or PDMA1, verify the CLKGATE status.

11.2 Register Description

Most of the Special Function Registers (SFRs) are read-only. The main role of SFR is to verify the PL330 status. There are many SFRs for PL330. This section describes only the Exynos 4412 SCP-specific SFRs. Refer to Chapter 3, "PL330 TRM," for more information.

11.2.1 Register Map Summary

- Base Address: 0x1284_0000 (MDMA)

Register	Offset	Description	Reset Value
DS	0x0000	Specifies the DMA status register. Refer to page 3-11 of "PL330 TRM" for more information.	0x200
DPC	0x0004	Specifies the DMA program counter register. Refer to page 3-13 of "PL330 TRM" for more information.	0x0
RSVD	0x0008 to 0x001C	Reserved	Undefined
INTEN	0x0020	Specifies the interrupt enable register. Refer to page 3-13 of "PL330 TRM" for more information.	0x0
ES	0x0024	Specifies the event status register. Refer to page 3-14 of "PL330 TRM" for more information.	0x0
INTSTATUS	0x0028	Specifies the interrupt status register. Refer to page 3-16 of "PL330 TRM" for more information.	0x0
INTCLR	0x002C	Specifies the interrupt clear register. Refer to page 3-17 of "PL330 TRM" for more information.	0x0
FSM	0x0030	Specifies the fault status DMA manager register. Refer to page 3-18 of "PL330 TRM" for more information.	0x0
FSC	0x0034	Specifies the fault status DMA channel register. Refer to page 3-19 of "PL330 TRM" for more information.	0x0
FTM	0x0038	Specifies the fault type DMA manager register. Refer to page 3-20 of "PL330 TRM" for more information.	0x0
RSVD	0x003C	Reserved	Undefined
FTC0	0x0040	Specifies the fault type for DMA channel 0.	0x0
FTC1	0x0044	Specifies the fault type for DMA channel 1.	0x0
FTC2	0x0048	Specifies the fault type for DMA channel 2.	0x0
FTC3	0x004C	Specifies the fault type for DMA channel 3.	0x0
FTC4	0x0050	Specifies the fault type for DMA channel 4.	0x0
FTC5	0x0054	Specifies the fault type for DMA channel 5.	0x0
FTC6	0x0058	Specifies the fault type for DMA channel 6.	0x0
FTC7	0x005C	Specifies the fault type for DMA channel 7.	0x0
RSVD	0x0060 to 0x00FC	Reserved	Undefined
CS0	0x0100	Specifies the channel status for DMA channel 0.	0x0
CS1	0x0108	Specifies the channel status for DMA channel 1.	0x0

Register	Offset	Description	Reset Value
CS2	0x0110	Specifies the channel status for DMA channel 2.	0x0
CS3	0x0118	Specifies the channel status for DMA channel 3.	0x0
CS4	0x0120	Specifies the channel status for DMA channel 4.	0x0
CS5	0x0128	Specifies the channel status for DMA channel 5.	0x0
CS6	0x0130	Specifies the channel status for DMA channel 6.	0x0
CS7	0x0138	Specifies the channel status for DMA channel 7.	0x0
CPC0	0x0104	Specifies the channel PC for DMA channel 0.	0x0
CPC1	0x010C	Specifies the channel PC for DMA channel 1.	0x0
CPC2	0x0114	Specifies the channel PC for DMA channel 2.	0x0
CPC3	0x011C	Specifies the channel PC for DMA channel 3.	0x0
CPC4	0x0124	Specifies the channel PC for DMA channel 4.	0x0
CPC5	0x012C	Specifies the channel PC for DMA channel 5.	0x0
CPC6	0x0134	Specifies the channel PC for DMA channel 6.	0x0
CPC7	0x013C	Specifies the channel PC for DMA channel 7.	0x0
RSVD	0x0140 to 0x03FC	Reserved	Undefined
SA_0	0x0400	Specifies the source address for DMA channel 0.	0x0
SA_1	0x0420	Specifies the source address for DMA channel 1.	0x0
SA_2	0x0440	Specifies the source address for DMA channel 2.	0x0
SA_3	0x0460	Specifies the source address for DMA channel 3.	0x0
SA_4	0x0480	Specifies the source address for DMA channel 4.	0x0
SA_5	0x04A0	Specifies the source address for DMA channel 5.	0x0
SA_6	0x04C0	Specifies the source address for DMA channel 6.	0x0
SA_7	0x04E0	Specifies the source address for DMA channel 7.	0x0
DA_0	0x0404	Specifies the destination address for DMA channel 0.	0x0
DA_1	0x0424	Specifies the destination address for DMA channel 1.	0x0
DA_2	0x0444	Specifies the destination address for DMA channel 2.	0x0
DA_3	0x0464	Specifies the destination address for DMA channel 3.	0x0
DA_4	0x0484	Specifies the destination address for DMA channel 4.	0x0
DA_5	0x04A4	Specifies the destination address for DMA channel 5.	0x0
DA_6	0x04C4	Specifies the destination address for DMA channel 6.	0x0
DA_7	0x04E4	Specifies the destination address for DMA channel 7.	0x0
CC_0	0x0408	Specifies the channel control for DMA channel 0.	0x00800200
CC_1	0x0428	Specifies the channel control for DMA channel 1.	0x00800200
CC_2	0x0448	Specifies the channel control for DMA channel 2.	0x00800200
CC_3	0x0468	Specifies the channel control for DMA channel 3.	0x00800200
CC_4	0x0488	Specifies the channel control for DMA channel 4.	0x00800200

Register	Offset	Description	Reset Value
CC_5	0x04A8	Specifies the channel control for DMA channel 5.	0x00800200
CC_6	0x04C8	Specifies the channel control for DMA channel 6.	0x00800200
CC_7	0x04E8	Specifies the channel control for DMA channel 7.	0x00800200
LC0_0	0x040C	Specifies the loop counter 0 for DMA channel 0.	0x0
LC0_1	0x042C	Specifies the loop counter 0 for DMA channel 1.	0x0
LC0_2	0x044C	Specifies the loop counter 0 for DMA channel 2.	0x0
LC0_3	0x046C	Specifies the loop counter 0 for DMA channel 3.	0x0
LC0_4	0x048C	Specifies the loop counter 0 for DMA channel 4.	0x0
LC0_5	0x04AC	Specifies the loop counter 0 for DMA channel 5.	0x0
LC0_6	0x04CC	Specifies the loop counter 0 for DMA channel 6.	0x0
LC0_7	0x04EC	Specifies the loop counter 0 for DMA channel 7.	0x0
LC1_0	0x0410	Specifies the loop counter 1 for DMA channel 0.	0x0
LC1_1	0x0430	Specifies the loop counter 1 for DMA channel 1.	0x0
LC1_2	0x0450	Specifies the loop counter 1 for DMA channel 2.	0x0
LC1_3	0x0470	Specifies the loop counter 1 for DMA channel 3.	0x0
LC1_4	0x0490	Specifies the loop counter 1 for DMA channel 4.	0x0
LC1_5	0x04B0	Specifies the loop counter 1 for DMA channel 5.	0x0
LC1_6	0x04D0	Specifies the loop counter 1 for DMA channel 6.	0x0
LC1_7	0x04F0	Specifies the loop counter 1 for DMA channel 7.	0x0
RSVD	0x0414 to 0x041C	Reserved	Undefined
RSVD	0x0434 to 0x043C	Reserved	Undefined
RSVD	0x0454 to 0x045C	Reserved	Undefined
RSVD	0x0474 to 0x047C	Reserved	Undefined
RSVD	0x0494 to 0x049C	Reserved	Undefined
RSVD	0x04B4 to 0x04BC	Reserved	Undefined
RSVD	0x04D4 to 0x04DC	Reserved	Undefined
RSVD	0x04F4 to 0x0CFC	Reserved	Undefined
DBGSTATUS	0x0D00	Specifies the debug status register. Refer to page 3-37 of "PL330 TRM" for more information.	0x0
DBGCMD	0x0D04	Specifies the debug command register. Refer to page 3-37 of "PL330 TRM" for more information.	Undefined

Register	Offset	Description	Reset Value
DBGINST0	0x0D08	Specifies the debug instruction-0 register. Refer to page 3-38 of "PL330 TRM" for more information.	Undefined
DBGINST1	0x0D0C	Specifies the debug instruction-1 register. Refer to page 3-39 of "PL330 TRM" for more information.	Undefined
CR0	0x0E00	Specifies the configuration register 0. Refer to page 3-40 of "PL330 TRM" for more information.	0x003E_0075
CR1	0x0E04	Specifies the configuration register 1. Refer to page 3-42 of "PL330 TRM" for more information.	0x0000_0075
CR2	0x0E08	Specifies the configuration register 2. Refer to page 3-43 of "PL330 TRM" for more information.	0x0
CR3	0x0E0C	Specifies the configuration register 3. Refer to page 3-44 of "PL330 TRM" for more information.	0x0
CR4	0x0E10	Specifies the configuration register 4. Refer to page 3-45 of "PL330 TRM" for more information.	0x0000_0001
CRDn	0x0E14	Specifies the configuration register Dn. Refer to page 3-46 of "PL330 TRM" for more information.	0x03F7_3733
periph_id_n	0x0FE0 to 0x0FEC	Specifies the peripheral identification registers 0-3. Refer to page 3-48 of "PL330 TRM" for more information.	Configuration-dependent
pcell_id_n	0x0FF0 to 0x0FFC	Specifies the primecell identification registers 0-3. Refer to page 3-50 of "PL330 TRM" for more information.	Configuration-dependent

- Base Address: 0x1268_000, 0x1269_0000 (PDMA0/PDMA1)

Register	Offset	Description	Reset Value
DS	0x0000	Specifies the DMA status register. Refer to page 3-11 of "PL330 TRM" for more information.	0x00000200
DPC	0x0004	Specifies the DMA program counter register. Refer to page 3-13 of "PL330 TRM" for more information.	0x0
RSVD	0x0008 to 0x001C	Reserved	Undefined
INTEN	0x0020	Specifies the interrupt enable register. Refer to page 3-13 of "PL330 TRM" for more information.	0x0
ES	0x0024	Specifies the event status register. Refer to page 3-14 of "PL330 TRM" for more information.	0x0
INTSTATUS	0x0028	Specifies the interrupt status register. Refer to page 3-16 of "PL330 TRM" for more information.	0x0
INTCLR	0x002C	Specifies the interrupt clear register. Refer to page 3-17 of "PL330 TRM" for more information.	0x0
FSM	0x0030	Specifies the fault status DMA manager register. Refer to page 3-18 of "PL330 TRM" for more information.	0x0
FSC	0x0034	Specifies the fault status DMA channel register. Refer to page 3-19 of "PL330 TRM" for more information.	0x0
FTM	0x0038	Specifies the fault type DMA manager register. Refer to page 3-20 of "PL330 TRM" for more information.	0x0
RSVD	0x003C	Reserved	Undefined
FTC0	0x0040	Specifies the fault type for DMA channel 0.	0x0
FTC1	0x0044	Specifies the fault type for DMA channel 1.	0x0
FTC2	0x0048	Specifies the fault type for DMA channel 2.	0x0
FTC3	0x004C	Specifies the fault type for DMA channel 3.	0x0
FTC4	0x0050	Specifies the fault type for DMA channel 4.	0x0
FTC5	0x0054	Specifies the fault type for DMA channel 5.	0x0
FTC6	0x0058	Specifies the fault type for DMA channel 6.	0x0
FTC7	0x005C	Specifies the fault type for DMA channel 7.	0x0
RSVD	0x0060 to 0x00FC	Reserved	Undefined
CS0	0x0100	Specifies the channel status for DMA channel 0.	0x0
CS1	0x0108	Specifies the channel status for DMA channel 1.	0x0
CS2	0x0110	Specifies the channel status for DMA channel 2.	0x0
CS3	0x0118	Specifies the channel status for DMA channel 3.	0x0
CS4	0x0120	Specifies the channel status for DMA channel 4.	0x0
CS5	0x0128	Specifies the channel status for DMA channel 5.	0x0
CS6	0x0130	Specifies the channel status for DMA channel 6.	0x0
CS7	0x0138	Specifies the channel status for DMA channel 7.	0x0

Register	Offset	Description	Reset Value
CPC0	0x0104	Specifies the channel PC for DMA channel 0.	0x0
CPC1	0x010C	Specifies the channel PC for DMA channel 1.	0x0
CPC2	0x0114	Specifies the channel PC for DMA channel 2.	0x0
CPC3	0x011C	Specifies the channel PC for DMA channel 3.	0x0
CPC4	0x0124	Specifies the channel PC for DMA channel 4.	0x0
CPC5	0x012C	Specifies the channel PC for DMA channel 5.	0x0
CPC6	0x0134	Specifies the channel PC for DMA channel 6.	0x0
CPC7	0x013C	Specifies the channel PC for DMA channel 7.	0x0
RSVD	0x0140 to 0x03FC	Reserved	Undefined
SA_0	0x0400	Specifies the source address for DMA channel 0.	0x0
SA_1	0x0420	Specifies the source address for DMA channel 1.	0x0
SA_2	0x0440	Specifies the source address for DMA channel 2.	0x0
SA_3	0x0460	Specifies the source address for DMA channel 3.	0x0
SA_4	0x0480	Specifies the source address for DMA channel 4.	0x0
SA_5	0x04A0	Specifies the source address for DMA channel 5.	0x0
SA_6	0x04C0	Specifies the source address for DMA channel 6.	0x0
SA_7	0x04E0	Specifies the source address for DMA channel 7.	0x0
DA_0	0x0404	Specifies the destination address for DMA channel 0.	0x0
DA_1	0x0424	Specifies the destination address for DMA channel 1.	0x0
DA_2	0x0444	Specifies the destination address for DMA channel 2.	0x0
DA_3	0x0464	Specifies the destination address for DMA channel 3.	0x0
DA_4	0x0484	Specifies the destination address for DMA channel 4.	0x0
DA_5	0x04A4	Specifies the destination address for DMA channel 5.	0x0
DA_6	0x04C4	Specifies the destination address for DMA channel 6.	0x0
DA_7	0x04E4	Specifies the destination address for DMA channel 7.	0x0
CC_0	0x0408	Specifies the channel control for DMA channel 0.	0x0
CC_1	0x0428	Specifies the channel control for DMA channel 1.	0x0
CC_2	0x0448	Specifies the channel control for DMA channel 2.	0x0
CC_3	0x0468	Specifies the channel control for DMA channel 3.	0x0
CC_4	0x0488	Specifies the channel control for DMA channel 4.	0x0
CC_5	0x04A8	Specifies the channel control for DMA channel 5.	0x0
CC_6	0x04C8	Specifies the channel control for DMA channel 6.	0x0
CC_7	0x04E8	Specifies the channel control for DMA channel 7.	0x0
LC0_0	0x040C	Specifies the loop counter 0 for DMA channel 0.	0x0
LC0_1	0x042C	Specifies the loop counter 0 for DMA channel 1.	0x0
LC0_2	0x044C	Specifies the loop counter 0 for DMA channel 2.	0x0

Register	Offset	Description	Reset Value
LC0_3	0x046C	Specifies the loop counter 0 for DMA channel 3.	0x0
LC0_4	0x048C	Specifies the loop counter 0 for DMA channel 4.	0x0
LC0_5	0x04AC	Specifies the loop counter 0 for DMA channel 5.	0x0
LC0_6	0x04CC	Specifies the loop counter 0 for DMA channel 6.	0x0
LC0_7	0x04EC	Specifies the loop counter 0 for DMA channel 7.	0x0
LC1_0	0x0410	Specifies the loop counter 1 for DMA channel 0.	0x0
LC1_1	0x0430	Specifies the loop counter 1 for DMA channel 1.	0x0
LC1_2	0x0450	Specifies the loop counter 1 for DMA channel 2.	0x0
LC1_3	0x0470	Specifies the loop counter 1 for DMA channel 3.	0x0
LC1_4	0x0490	Specifies the loop counter 1 for DMA channel 4.	0x0
LC1_5	0x04B0	Specifies the loop counter 1 for DMA channel 5.	0x0
LC1_6	0x04D0	Specifies the loop counter 1 for DMA channel 6.	0x0
LC1_7	0x04F0	Specifies the loop counter 1 for DMA channel 7.	0x0
RSVD	0x0414 to 0x041C	Reserved	Undefined
RSVD	0x0434 to 0x043C	Reserved	Undefined
RSVD	0x0454 to 0x045C	Reserved	Undefined
RSVD	0x0474 to 0x047C	Reserved	Undefined
RSVD	0x0494 to 0x049C	Reserved	Undefined
RSVD	0x04B4 to 0x04BC	Reserved	Undefined
RSVD	0x04D4 to 0x04DC	Reserved	Undefined
RSVD	0x04F4 to 0x0CFC	Reserved	Undefined
DBGSTATUS	0x0D00	Specifies the debug status register on page 3-37 of "TRM".	0x0
DBGCMD	0x0D04	Specifies the debug command register. Refer to page 3-37 of "PL330 TRM" for more information.	Undefined
DBGINST0	0x0D08	Specifies the debug instruction-0 register. Refer to page 3-38 of "PL330 TRM" for more information.	Undefined
DBGINST1	0x0D0C	Specifies the debug instruction-1 register. Refer to page 3-39 of "pl330 TRM" for more information.	Undefined
CR0	0x0E00	Specifies the configuration register 0. Refer to page 3-40 of "PL330 TRM" for more information.	0x003F_F075
CR1	0x0E04	Specifies the configuration register 1. Refer to page 3-42 of "PL330 TRM" for more information.	0x0000_0074

Register	Offset	Description	Reset Value
CR2	0x0E08	Specifies the configuration register 2. Refer to page 3-43 of "PL330 TRM" for more information.	0x0000_0000
CR3	0x0E0C	Specifies the configuration register 3. Refer to page 3-44 of "PL330 TRM" for more information.	0x0
CR4	0x0E10	Specifies the configuration register 4. Refer to page 3-45 of "PL330 TRM" for more information.	0xFFFF_FFFF
CRDn	0x0E14	Specifies the configuration register Dn. Refer to page 3-46 of "PL330 TRM" for more information.	0x01F7_3732
periph_id_n	0x0FE0 to 0x0FEC	Specifies the peripheral identification registers 0-3 Refer to page 3-48 of "PL330 TRM" for more information.	Configuration-dependent
pcell_id_n	0x0FF0 to 0x0FFC	Specifies the primecell identification registers 0-3. Refer to page 3-50 of "PL330 TRM" for more information.	Configuration-dependent

NOTE: The SFR description shows only the restricted and fixed part of some SFR. PL330 TRM shows detailed information of other parts and other SFRs.

11.2.1.1 CC

CC includes:

- Base Address: 0x1284_0000, 0x1268_0000, 0x1269_0000
- CC_0 Address = Base Address + 0x0408, Reset Value = 0x0080_0200
- CC_1 Address = Base Address + 0x0428, Reset Value = 0x0080_0200
- CC_2 Address = Base Address + 0x0448, Reset Value = 0x0080_0200
- CC_3 Address = Base Address + 0x0468, Reset Value = 0x0080_0200
- CC_4 Address = Base Address + 0x0488, Reset Value = 0x0080_0200
- CC_5 Address = Base Address + 0x04A8, Reset Value = 0x0080_0200
- CC_6 Address = Base Address + 0x04C8, Reset Value = 0x0080_0200
- CC_7 Address = Base Address + 0x04E8, Reset Value = 0x0080_0200

11.3 Instruction

Table 11-3 describes the instruction syntax summary.

Table 11-3 Instruction Syntax Summary

Mnemonic	Instruction	Thread Usage: M = DMA Manager C = DMA Channel		Description
DMAADDH	Add halfword	–	C	Refer DMAADDH on page 4-5 of "PL330 TRM".
DMAEND	End	M	C	Refer DMAEND on page 4-5 of "PL330 TRM".
DMAFLUSHP	Flush and notify Peripheral	–	C	Refer DMAFLUSHP on page 4-6 of "PL330 TRM".
DMAGO	Go	M	–	Refer DMAGO on page 4-6 of "PL330 TRM".
DMALD	Load	–	C	Refer DMALD[S|B] on page 4-8 of "PL330 TRM".
DMALDP	Load peripheral	–	C	Refer DMALDP<S|B> on page 4-9 "PL330 TRM".
DMALP	Loop	–	C	Refer DMALP on page 4-10 of "PL330 TRM".
DMALPEND	Loop end	–	C	Refer DMALPEND[S|B] on page 4-11 of "PL330 TRM".
DMALPFE	Loop forever	–	C	Refer DMALPFE on page 4-13 of "PL330 TRM".
DMAKILL	Kill	M	C	Refer DMAKILL on page 4-13 of "PL330 TRM".
DMAMOV	Move	–	C	Refer DMAMOV on page 4-14 of "PL330 TRM".
DMANOP	No operation	M	C	Refer DMANOP on page 4-16 of "PL330 TRM".
DMARMB	Read memory barrier	–	C	Refer DMARMB on page 4-16 of "PL330 TRM".
DMASEV	Send event	M	C	Refer DMASEV on page 4-17 of "PL330 TRM".
DMAST	Store	–	C	Refer DMAST[S|B] on page 4-17 of "PL330 TRM".
DMASTP	Store and notify Peripheral	–	C	Refer DMASTP<S|B> on page 4-19 of "PL330 TRM".
DMASTZ	Store zero	–	C	Refer DMASTZ on page 4-20 of "PL330 TRM".
DMAWFE	Wait for event	M	C	Refer DMAWFE on page 4-20 of "PL330 TRM".
DMAWFP	Wait for peripheral	–	C	Refer DMAWFP<S|B|P> on page 4-21 of "PL330 TRM".
DMAWMB	Write Memory Barrier	–	C	See DMAWMB on page 4-22 of "PL330 TRM".

Each PL330 has a manager thread and eight channel threads. A manager thread controls the overall operation of Direct Memory Access Controller (DMAC) that includes initiating and killing the channel. The channel thread operates the DMA.

11.3.1 Key Instruction

To run the channel thread, write the assembly code.

The DMAMOV section lists the description of key instruction. Refer to Chapter 4, "PL330 TRM" for the entire instruction set.

11.3.1.1 DMAMOV

"Move" instructs the DMAC to move 32 bits immediately into:

- Source Address REG (SAR)
- Destination Address REG (DAR)
- Channel Control REG (CCR)

SAR

- Example: DMAMOV SAR, 0x24000000
  - 0x2400_0000 is the source address of DMA operation.

DAR

- Example: DMAMOV DAR, 0x24001000
  - 0x2400_1000 is destination address of DMA operation.

CCR

- Example: DMAMOV CCR, SB2 SS32 SP0 DB2 DS32 DP0
  - Source: Burst length is 2, 32-bit data width.
  - Destination: Burst length is 2, 32-bit data width.
  - SP0 and DP0 refer to normal and secure, respectively.
  - SP2 and DP2 refers to normal and non-secure, respectively
- Refer to pages 4-25 to 4-26 in Chapter 4, "PL330 TRM" to know about the exact DMA settings, such as burst length, bit-width, address increment, and so on.

11.3.1.2 DMALD and DMALDP

"Load" instructs the DMAC to perform DMA load by using AXI transactions. DAR specifies source and CCR configures AXI transaction. For example, when you define CCR as 32-bit and burst length as 2, the DMALD generates a bus transaction of 32-bit and burst length 2. DMALDP notifies the peripheral when data transfer is complete.

11.3.1.3 DMAST and DMASTP

"Store" instructs the DMAC to transfer data from FIFO to a location. DAR specifies destination and CCR configures AXI transaction. For example, when you define CCR as 32-bit and burst length as 2, the DMAST generates a bus transaction of 32-bit and burst length 2. DMASTP notifies the peripheral when the data transfer is complete.

11.3.1.4 DMASTZ

"Store Zero" instructs the DMAC to store zeros by using AXI transactions. DAR specifies destination and CCR configures AXI transaction. For example, when you define CCR as 32-bit and burst length as 2, the DMASTZ generates a bus transaction of 32-bit and burst length 2 with zeros at data bus.

11.3.1.5 DMALP and DMALPEND

"DMALP lc0, 4[code] to DMALPEND lc0" loops (iterates) the "[code]" 4 times. There are two loop counters:

- lc0
- lc1

Use the nested loop by two loop counters.

11.3.1.6 DMAWFP

This is used for peripheral DMA. "Wait for Peripheral" instructs the DMAC to stop the execution of thread until the specified peripheral signals a DMA request for that DMA channel.

11.3.1.7 DMAFLUSHP

This is used for peripheral DMA. "Flush Peripheral" clears the state in DMA that describes the contents of the peripheral. DMAFLUSHP instruction also sends a message to the peripheral to resend its level status. This instruction asserts DMAACK. Place this instruction at that point where you need DMAACK.

11.3.1.8 DMAEND

This is used to instruct a channel to stop.

11.3.2 USAGE Model

PL330 requires its own binary.

You can describe the usage model to:

1. Load DMA binary into memory.
2. Use DMA debug SFRs to start the PL330 DMA controller.
   - Using debug SFRs:
     - Use DBGCMD, DBGINST0, and DBGINST1 (all write-only). Before writing the above three SFRs, verify whether DBGSTATUS is busy or not.
     - Refer to pages 3-37 to 3-40 in "PL330 TRM" for more information.
   - DBGINST0 and DBGINST1 contains debug instructions:
     - These SFRs can receive only three instructions namely, DMAGO, DMASEV, and DMAKILL.
     - DMAGO starts a channel. Refer to page number 3-38 to 3-40 and page number 4-6 to 4-8 in "PL330 TRM for more information.
   - DBGCMD executes the instruction stored in the DBGINST0 and SFRs.

Example 11-1 describes the USAGE model.

Example 11-1 USAGE Model

```
; Load channel control register
; Single transfer, 32-bit/non-secure
DMAMOV   CCR, SB1  SS32  SP2  DB1  DS32  DP2
   ; SB1, DB1      : Burst length: 1
   ; SS32, DS32    : 32-bit Data I/F
   ; SP0, DP0      : Secure access
   ; SP2, DP2      : Non-secure access
; in case of Peripheral transfer , should be Initialise peripheral
DMAFLUSHP 0

; Source: IntRAM0,   Destination: IntRAM1
DMAMOV   SAR, 0xd0020000
DMAMOV   DAR, 0xd0028000
DMALP  lc0, 32
DMALD
DMAST
DMALPEND lc0
DMASEV  E0
DMAEND
```

11.3.2.1 Security Scheme

DMA_mem runs in both secure and non-secure modes, while DMA_peri runs in non-secure mode only.

1. Channel thread:
   - DMA_mem: Runs in both secure (ns bit at DMAGO instruction is 0) and non-secure (ns bit at DMAGO instruction is 1) modes.
   - DMA_peri: Runs in non-secure (ns bit at DMAGO instruction is 1) mode only.

2. ASM code
   - For non-secure transaction:
     - Use SP2 and DP2 at DMAMOV instruction.
     - APROT[1] will be 1'b1.
   - For secure transaction:
     - Use SP0 and DP0 at DMAMOV instruction.
     - APROT[1] will be 1'b0.

11.3.2.2 Interrupts

DMAC provides IRQ signals for use as level sensitive interrupts the external CPUs. If you program the Interrupt Enable Register to generate an interrupt after DMAC executes DMASEV, DMASEV instruction sets the corresponding IRQ as HIGH.

Clear the interrupt by writing to the Interrupt Clear Register.

To control the interrupt:

1. Set up the Interrupt Enable Register to generate interrupts:
   - The interrupt enable register is a 32-bit register. Each bit of the INTEN register verifies whether the DMAC signals an interrupt by using the corresponding IRQ.
   - Program the appropriate bit to control the DMAC response on execution of DMASEV:
     - Bit[N] = 0: If you are executing DMASE for even N, then the DMAC signals event N to all the threads.
     - Bit[N] = 1: If you are executing DMASE for event N, then the DMAC sets irq[N] as HIGH.

2. Program an assembly code, to set the corresponding IRQ HIGH by executing DMASEV:
   - Use DMASEV instruction means an interrupt that uses one of the IRQ outputs.

3. Clear the interrupt by writing to the Interrupt Clear Register:
   - Each bit in the INTCLR register controls the clearing of an interrupt.
   - Program to control the clearing of the IRQ outputs:
     - Bit[N] = 0: The status of irq[N] does not change. Bit[N] = 1: The DMAC sets irq[N] as LOW.
     - If DMA is set to fault status, an interrupt occurs.

11.3.2.3 Summary

1. You can configure the DMAC with up to eight DMA channels. Each channel supports single concurrent thread of DMA operation. Additionally, there is a single DMA manager thread to initialize the DMA channel thread.

2. Channel thread
   - Each channel thread can operate the DMA. Accordingly, write an assembly code. If you require a number of independent DMA channels, write a number of assembly codes for each channel.
   - Assemble and link the codes into one file and load this file into the memory.

12 System Registers

12.1 Overview

The System Registers (SYSREG) generates various control signals for the ARM CPU, sub-blocks, IPs, and buses. SYSREG contains APB bus interface and registers. The register values come out of SYSREG module and are sent to appropriate destination.

12.2 Register Description

12.2.1 Register Map Summary

- Base Address: 0x1001_0000

Register	Offset	Description	Reset Value
GENERAL_CTRL_C2C	0x010C	General control register for C2C	0x832A_A803
GENERAL_CTRL_GPS	0x0110	General control register for GPS	0x0000_0005
ADC_CFG	0x0118	ADC control register	0x0000_0000
ISPBLK_CFG	0x020C	ISP control register	0x0000_3F80
LCDBLK_CFG	0x0210	Display control register	0x00F8_0000
LCDBLK_CFG2	0x0214	Display control register	0x0000_0001
CAMBLK_CFG	0x0218	Camera control register	0x01BF_FC00
USB_CFG	0x021C	USB control register	0x0000_0000
PPMU_CON	0x0320	PPMU control register	0x0000_0000
SMMU_CON	0x0330	SMMU control register	0x0000_0000

12.2.1.1 GENERAL_CTRL_C2C

- Base Address: 0x1001_0000
- Address = Base Address + 0x010C, Reset Value = 0x832A_A803

Name	Bit	Type	Description	Reset Value
CG	[31]	RW	Clock Gating 0 = Gating off 1 = Gating On	0x1
MO	[30]	RW	Master On	0x0
FCLK_FREQ	[29:20]	RW	Function Clock Frequency	0x32
TXW	[19:18]	RW	Default TX width 0x0 = 8-bit 0x1 = 10-bit 0x2 = 16-bit 0x3 = RSV	0x2
RXW	[17:16]	RW	Default RX width 0x0 = 8-bit 0x1 = 10-bit 0x2 = 16-bit 0x3 = RSV	0x2
RSTn	[15]	RW	Reset Signal of C2C IP.	0x1
MD	[14]	RW	dram_init_done signal for wake-up sequence.	0x0
RET_RSTn	[13]	RW	Reset Signal of Retention Register in C2C IP.	0x1
BA	[12:3]	RW	Dram Base Address	0x100
SIZE	[2:0]	RW	Accessible DRAM range 512 MB = 3'b111 256 MB = 3'b110 128 MB = 3'b101 64 MB = 3'b100 32 MB = 3'b011 16 MB = 3'b010 8 MB = 3'b001 4 MB = 3'b000	0x3

12.2.1.2 GENERAL_CTRL_GPS

- Base Address: 0x1001_0000
- Address = Base Address + 0x0110, Reset Value = 0x0000_0005

Name	Bit	Type	Description	Reset Value
RESERVED_CTRL	[31:16]	RW	Reserved for Control	0x0
RESERVED_STATUS	[15:3]	R	Reserved for Status	0x0
ALV_SRSTN	[2]	RW	Software reset of GPA_ALIVE	0x1
MUX_SEL	[1]	RW	Mux Selection Control	0x0
SRSTN	[0]	RW	Software Reset	0x1

12.2.1.3 ADC_CFG

- Base Address: 0x1001_0000
- Address = Base Address + 0x0118, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RESERVED	[31:17]	RW	Reserved	0x0
ADC_MUX_SEL	[16]	RW	Select ADC Mux 0 : General ADC 1 : MTCADC_ISP	0x0
RESERVED	[15:0]	R	Reserved	0x0

12.2.1.4 ISPBLK_CFG

- Base Address: 0x1001_0000
- Address = Base Address + 0x020C, Reset Value = 0x0000_3F80

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	RW	Reserved	0x0
FIFORST_ISPBLK	[13:7]	RW	FIFO Software Reset of ISP Block (active Low) [0]: CAM_BLK [1]: LITE_M [2]: LITE_S [3]: ITU_A [4]: ITU_B [5]: MIPI_A [6]: MIPI_B	0x7F
RST_MASK_ISPBLK	[6:0]	RW	Reset Mask of ISP Block [0]: CAM_BLK [1]: LITE_M [2]: LITE_S [3]: ITU_A [4]: ITU_B [5]: MIPI_A [6]: MIPI_B	0x0

12.2.1.5 LCDBLK_CFG

- Base Address: 0x1001_0000
- Address = Base Address + 0x0210, Reset Value = 0x00F8_0000

Name	Bit	Type	Description	Reset Value
SBZ	[31:29]	RW	It should be zero	0x0
RESERVE_LBLK1	[28:24]	RW	Reserved for Later Usage	0x0
FIFORST_LBLK	[23:19]	RW	LCD_BLK FIFO software reset (active Low) [0]: L0 [1]: L1 [2]: L2 [3]: L3 [4]: WB	0x1F
RST_MASK_LBLK	[18:14]	RW	LO_MASK of PIXELASYNC [0]: L0 [1]: L1 [2]: L2 [3]: L3 [4]: WB	0x0
SBZ	[13]	RW	It should be zero	0x0
SBZ	[12]	RW	It should be zero	0x0
VT_LBLK0	[11:10]	RW	Video Type Selection 00 = RGB Interface 01 = i80 Interface 10 = reserved 11 = reserved	0x0
RESERVE_LBLK0	[9:8]	RW	Reserved for Later Usage	0x0
I80MS_LBLK0	[7:5]	RW	i80 Master Slave Relation for LBLK0	0x0
RGB_LBLK0	[4:2]	RW	RGB Reordering for LBLK0	0x0
FIMDBYPASS_LBLK0	[1]	RW	FIMD of LBLK0 Bypass Selection 0 = MIE/MDNIE 1 = FIMD Bypass	0x0
MIE_LBLK0	[0]	RW	MIE of LBLK0 Selection 0 = MIE 1 = MDNIE	0x0

12.2.1.6 LCDBLK_CFG2

- Base Address: 0x1001_0000
- Address Base Address + 0x0214, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	RW	Reserved	0x0
MIE0_SLPIN	[1]	RW	MIE0_SLPIN: PWM output polarity control 0 = Disable 1 = PWM output control by BLMODEINSLP (MIE sfr)	0x0
MIE0_DISPON	[0]	RW	MIE0_DISPON: PWM output control 0 = Disable 1 = PWM outpupt enable	0x1

12.2.1.7 CAMBLK_CFG

- Base Address: 0x1001_0000
- Address = Base Address + 0x0218, Reset Value = 0x01BF_FC00

Name	Bit	Type	Description	Reset Value
RESERVE_CAMBLK	[31:25]	RW	Reserved for Later Usage	0x0
FIMD0WB_DEST	[24:23]	RW	FIMD0 WB Interface Destination 00 = FIMC0 01 = FIMC1 10 = FIMC2 11 = FIMC3 should be different value from FIMD1WB_DEST	0x3
ISPWB_FULL_EN	[22:20]	RW	ISP WB Pixel Asynchronous FIFO Full Signal Enable 3'b001 = Enables FIMC0 FIFO full signal only 3'b010 = Enables FIMC1 FIFO full signal only 3'b100 = Enables FIMC2 FIFO full signal only 3'b111 = Enables FIMC0 & FIMC1 & FIMC2 FIFO full signals	0x3
FIFORST_ISP_ CAMBLK_S7	[19]	RW	MIPI_B Local Interface FIFO Software Reset	0x1
FIFORST_ISP_ CAMBLK_S6	[18]	RW	MIPI_A Local Interface FIFO Software Reset	0x1
FIFORST_ISP_ CAMBLK_S5	[17]	RW	CAM_B Local Interface FIFO Software Reset	0x1
FIFORST_ISP_ CAMBLK_S4	[16]	RW	CAM_A Local Interface FIFO Software Reset	0x1
FIFORST_ISP_ CAMBLK_M1	[15]	RW	ISP Write Back Interface FIFO Software Reset	0x1
FIFORST_LCD0_ CAMBLK_S3	[14]	RW	FIMC3 Local Interface FIFO Software Reset	0x1
FIFORST_LCD0_ CAMBLK_S2	[13]	RW	FIMC2 Local Interface FIFO Software Reset	0x1
FIFORST_LCD0_ CAMBLK_S1	[12]	RW	FIMC1 Local Interface FIFO Software Reset	0x1
FIFORST_LCD0_ CAMBLK_S0	[11]	RW	FIMC0 Local Interface FIFO Software Reset	0x1
FIFORST_LCD0_ CAMBLK_M0	[10]	RW	LCD0 Write Back Interface FIFO Software Reset	0x1
RST_MASK _ISP_CAMBLK_S7	[9]	RW	MIPI_B Local Interface FIFO Local Reset Mask 0 = Enables local reset 1 = Masks local reset	0x0
RST_MASK _ISP_CAMBLK_S6	[8]	RW	MIPI_A Local Interface FIFO Local Reset Mask 0 = Enables local reset 1 = Masks local reset	0x0

Name	Bit	Type	Description	Reset Value
RST_MASK _ISP_CAMBLK_S5	[7]	RW	CAM_B Local Interface FIFO Local Reset Mask 0 = Enables Local Reset 1 = Masks local reset	0x0
RST_MASK _ISP_CAMBLK_S4	[6]	RW	CAM_A Local Interface FIFO Local Reset Mask 0 = Enables local reset 1 = Masks local reset	0x0
RST_MASK _ISP_CAMBLK_M1	[5]	RW	ISP Write Back Interface FIFO Local Reset Mask 0 = Enables local reset 1 = Masks local reset	0x0
RST_MASK _LCD0_CAMBLK_S3	[4]	RW	FIMC3 Local Interface FIFO Local Reset Mask 0 = Enables local reset 1 = Masks local reset	0x0
RST_MASK _LCD0_CAMBLK_S2	[3]	RW	FIMC2 Local Interface FIFO Local Reset Mask 0 = Enables local reset 1 = Masks local reset	0x0
RST_MASK _LCD0_CAMBLK_S1	[2]	RW	FIMC1 Local Interface FIFO Local Reset Mask 0 = Enables local reset 1 = Masks local reset	0x0
RST_MASK _LCD0_CAMBLK_S0	[1]	RW	FIMC0 Local Interface FIFO Local Reset Mask 0 = Enables local reset 1 = Masks local reset	0x0
RST_MASK _LCD0_CAMBLK_M0	[0]	RW	LCD0 Write Back Interface FIFO Local Reset Mask 0 = Enables local reset 1 = Masks local reset	0x0

<ISPWB_FULL_EN[2:0]>

This register can prevent transmission of unused FIFO_FULL signals to Pixel Async module.

Figure 12-1 illustrates the diagram of the ISPWB_FULL_EN usage.

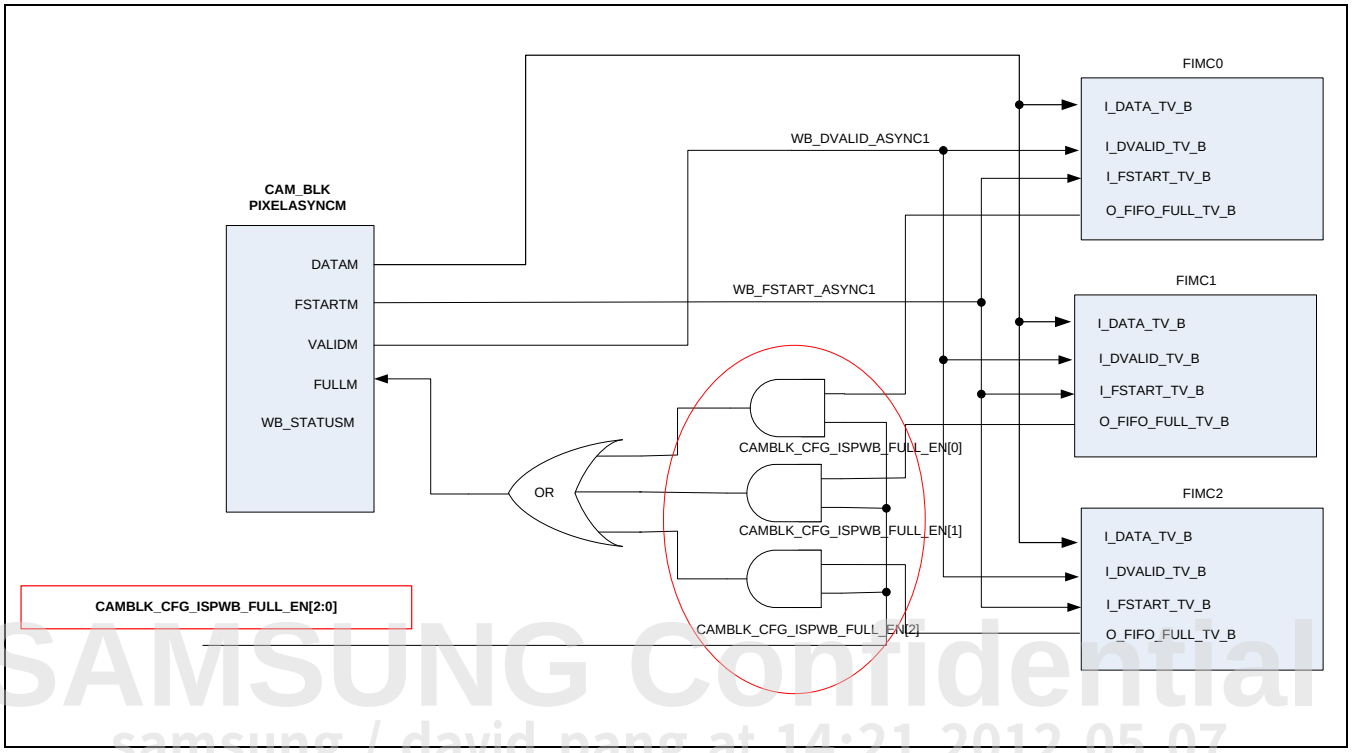


Figure 12-1 Diagram of the ISPWB_FULL_EN Usage

12.2.1.8 USB_CFG

- Base Address: 0x1001_0000
- Address = Base Address + 0x021C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	RW	Reserved	0x0
HOST_MODE	[0]	RW	USB PHY Mode Selection USB PHY operates as only one mode at a time by this selection. 0 = USB Device 1 = USB Host	0x0

12.2.1.9 PPMU_CON

- Base Address: 0x1001_0000
- Address = Base Address + 0x0320, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:21]	–	Reserved	0x0
ISPMX	[20]	RW	0 = Stop 1 = Start	0x0
ISPX	[19]	RW	0 = Stop 1 = Start	0x0
DMC1_M	[18]	RW	0 = Stop 1 = Start	0x0
DMC0_M	[17]	RW	0 = Stop 1 = Start	0x0
DMC1_A	[16]	RW	0 = Stop 1 = Start	0x0
DMC0_A	[15]	RW	0 = Stop 1 = Start	0x0
FSYS	[14]	RW	0 = Stop 1 = Start	0x0
RSVD	[13]	RW	Reserved	0x0
LCD0	[12]	RW	0 = Stop 1 = Start	0x0
IMAGE	[11]	RW	0 = Stop 1 = Start	0x0
G3D	[10]	RW	0 = Stop 1 = Start	0x0
MFC_L	[9]	RW	0 = Stop 1 = Start	0x0
MFC_R	[8]	RW	0 = Stop 1 = Start	0x0
TV	[7]	RW	0 = Stop 1 = Start	0x0
CAM	[6]	RW	0 = Stop 1 = Start	0x0
RIGHTBUS	[5]	RW	0 = Stop 1 = Start	0x0
LEFTBUS	[4]	RW	0 = Stop 1 = Start	0x0
DMC1	[3]	RW	0 = Stop 1 = Start	0x0
DMC0	[2]	RW	0 = Stop 1 = Start	0x0

Name	Bit	Type	Description	Reset Value
ACP	[1]	RW	0 = Stop 1 = Start	0x0
CPU	[0]	RW	0 = Stop 1 = Start	0x0

12.2.1.10 SMMU_CON

- Base Address: 0x1001_0000
- Address = Base Address + 0x0330, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	RW	Reserved	0x0
LITE1	[19]	RW	0 = Stop 1 = Start	0x0
LITE0	[18]	RW	0 = Stop 1 = Start	0x0
DRC	[17]	RW	0 = Stop 1 = Start	0x0
FD	[16]	RW	0 = Stop 1 = Start	0x0
ISP	[15]	RW	0 = Stop 1 = Start	0x0
ISPCX	[14]	RW	0 = Stop 1 = Start	0x0
FIMD0	[13]	RW	0 = Stop 1 = Start	0x0
MDMA	[12]	RW	0 = Stop 1 = Start	0x0
ROTATOR	[11]	RW	0 = Stop 1 = Start	0x0
G2D	[10]	RW	0 = Stop 1 = Start	0x0
MFC_L	[9]	RW	0 = Stop 1 = Start	0x0
MFC_R	[8]	RW	0 = Stop 1 = Start	0x0
TV	[7]	RW	0 = Stop 1 = Start	0x0
JPEG	[6]	RW	0 = Stop 1 = Start	0x0
FIMC3	[5]	RW	0 = Stop 1 = Start	0x0
FIMC2	[4]	RW	0 = Stop 1 = Start	0x0
FIMC1	[3]	RW	0 = Stop 1 = Start	0x0
FIMC0	[2]	RW	0 = Stop 1 = Start	0x0
SSS	[1]	RW	0 = Stop 1 = Start	0x0

Name	Bit	Type	Description	Reset Value
RSVD	[0]	RW	Reserved	0x0

13 CoreSight

13.1 Overview

This chapter describes the CoreSight of Exynos 4412 SCP. Coresight provides a system wide solution to real-time debug and trace in Exynos 4412 SCP.

Refer to CoreSight manual released by ARM for detailed description. Version of CoreSight Design Kit is r2p0.

Figure 13-1 illustrates the connection of the F4Q and CoreSight.

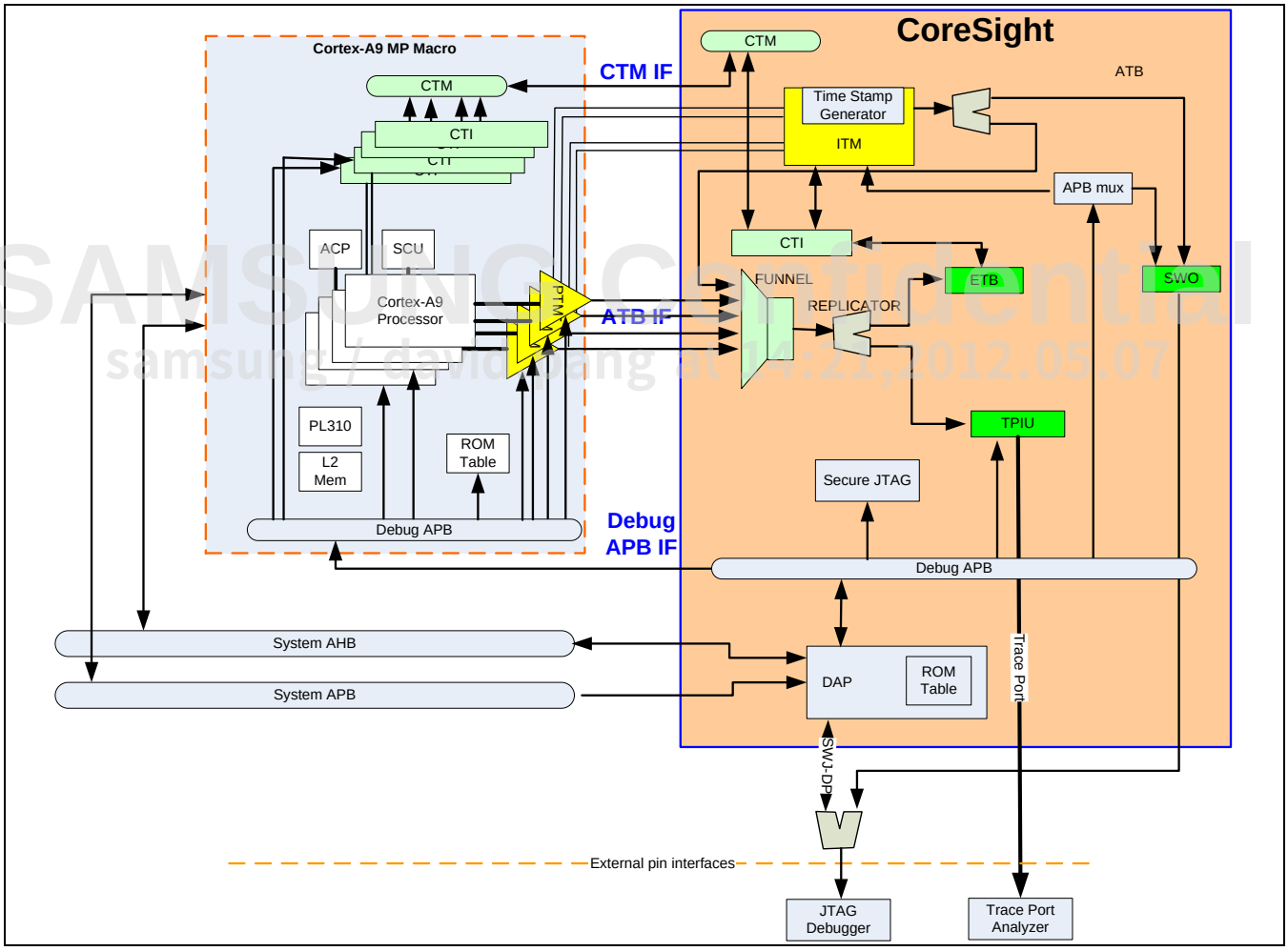


Figure 13-1 F4Q and CoreSight

When you see debug system, you can find three trace sources: PTM0 for CPU0, PTM1 for CPU1, ITM.
When trace sinks, only ETB and TPIU sources remain. You can operate PTM at up to 800 MHz. It can read ETB

through JTAG port or ARM can directly read AHB interface. For TPIU, it connects 16-bit data pin to external pads. It gathers all data stream from ITM through ETB or TPIU.

13.2 Components of CoreSight

13.2.1 This Section Includes:

- Debug Access Port (DAP)
- Single Wire and JTAG Debug Port (SWJ-DP)
- APB Multiplexer
- Trace Funnel
- CTI and CTM
- ROM Table and Multi-Drop ID

13.2.2 Debug Access Port (DAP)

Refer to CoreSight manual released by ARM for detailed description.

Figure 13-2 illustrates the structure of the CoreSight DAP components.

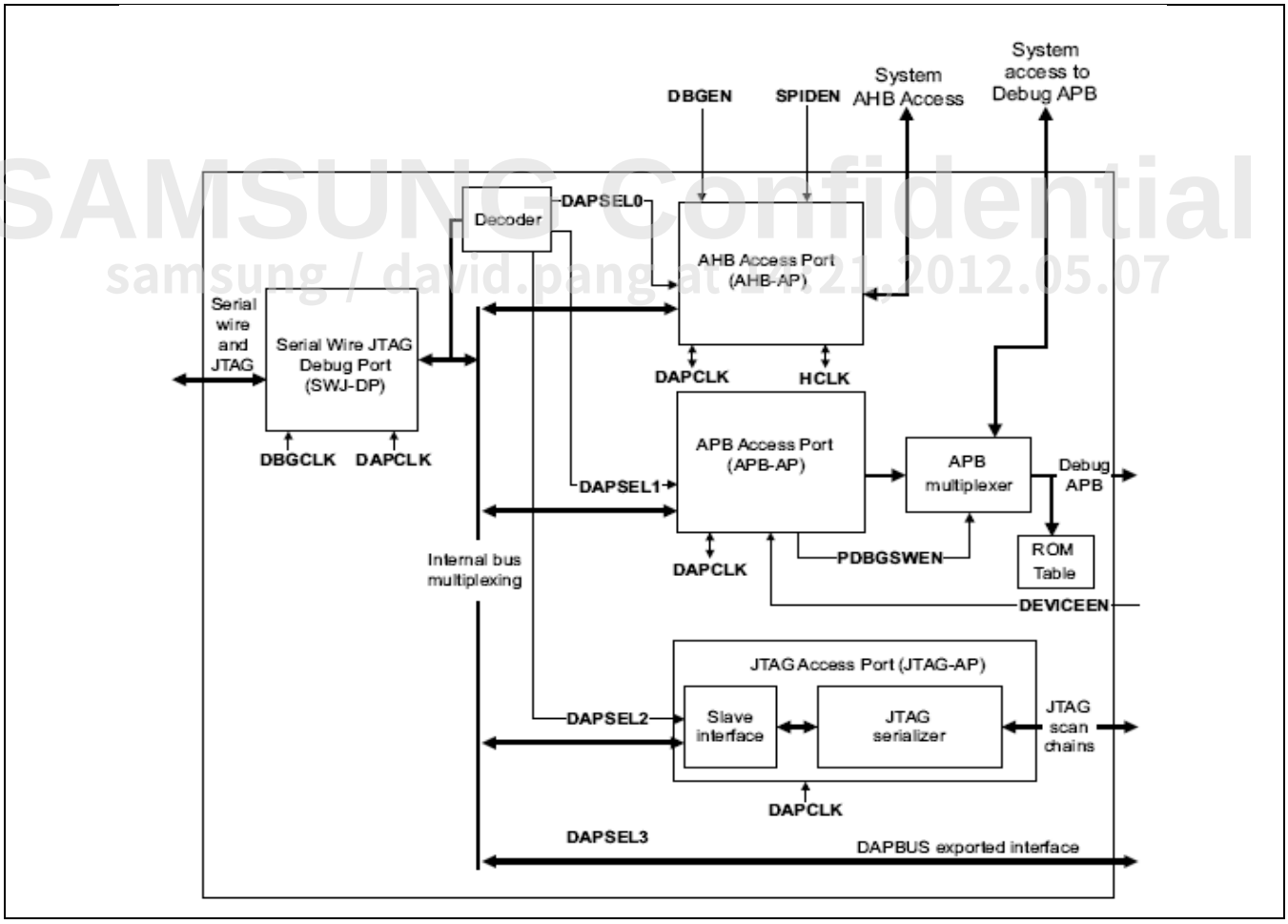


Figure 13-2 Structure of the CoreSight DAP Components

13.2.3 Single Wire and JTAG Debug Port (SWJ-DP)

Refer to CoreSight manual released by ARM for detailed description.

Figure 13-3 illustrates the SWJ-DP external connections.

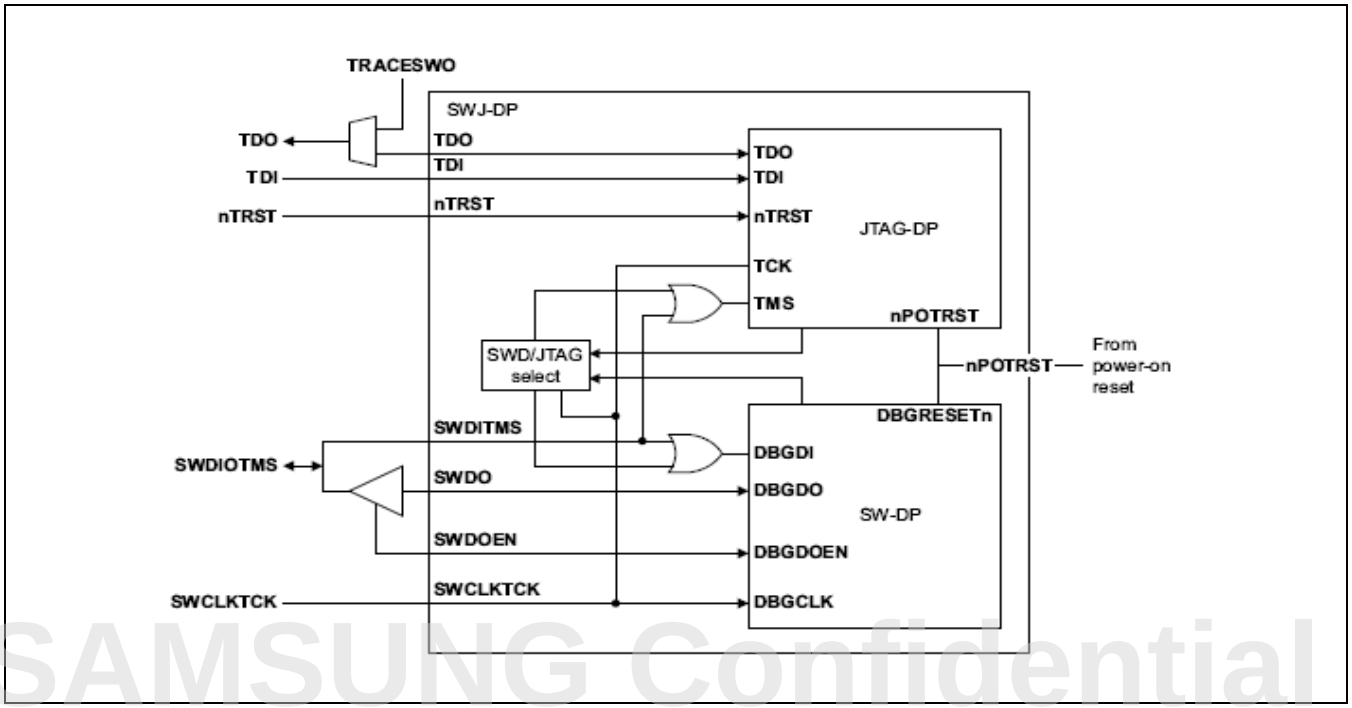


Figure 13-3 SWJ-DP External Connections

13.2.4 APB Multiplexer

Refer to CoreSight manual released by ARM for detailed description.

Figure 13-4 illustrates the APB multiplexer.

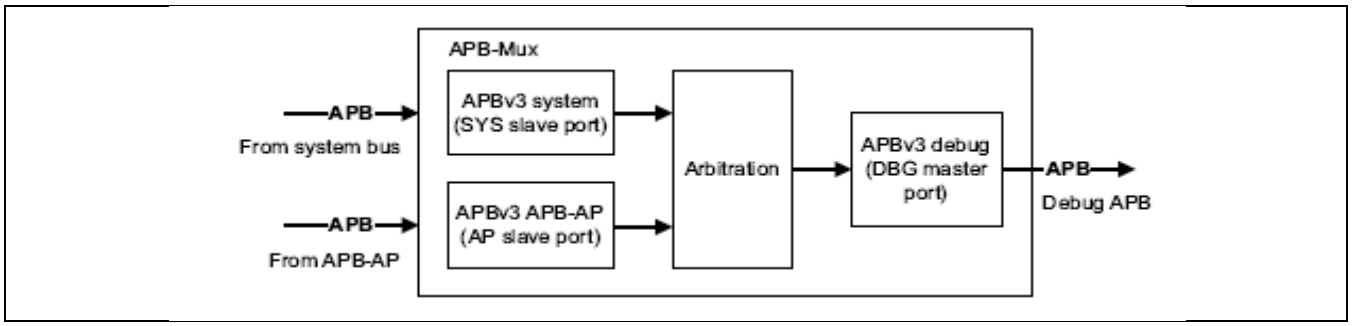


Figure 13-4 APB Multiplexer

13.2.5 Trace Funnel

Refer to CoreSight manual released by ARM for detailed description.

Figure 13-5 illustrates the trace funnel.

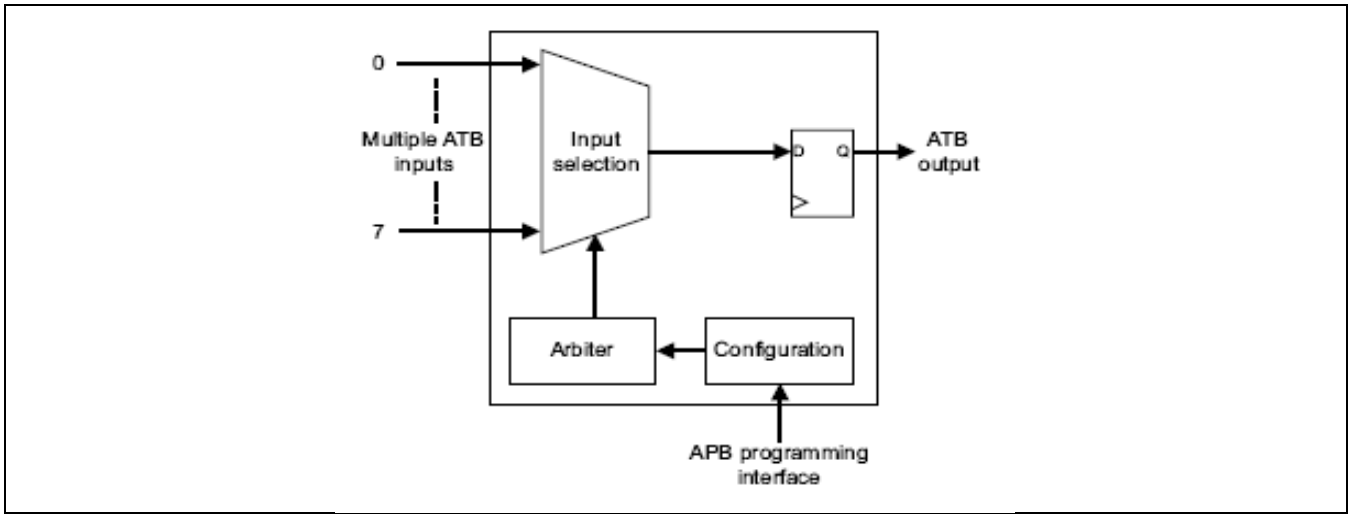


Figure 13-5 Trace Funnel

13.2.6 Cross Trigger Interface (CTI) and Cross Trigger Matrix (CTM)

Refer to CoreSight manual released by ARM for detailed description.

Figure 13-6 illustrates the CTI and CTM block diagram.

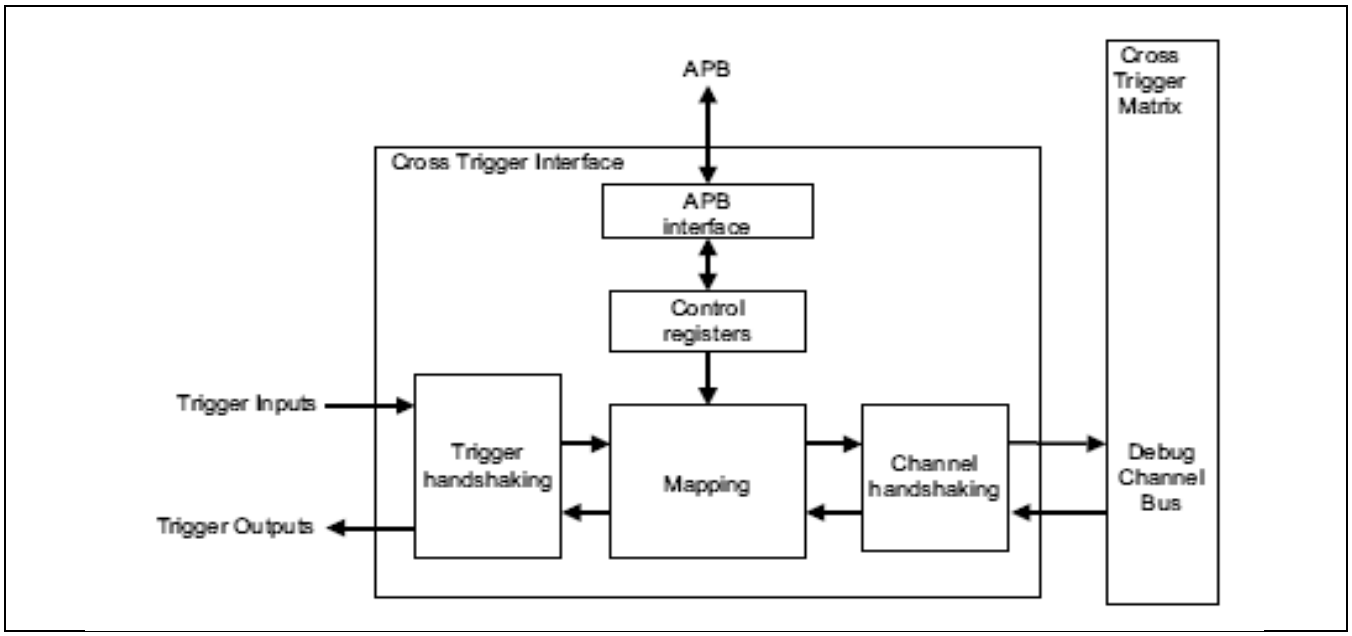


Figure 13-6 CTI and CTM Block Diagram

13.2.7 ROM Table and Multi-Drop ID

DAP ROM table register:

Samsung has the code 0xCE (8'b1100_1110) of JEDEC Standard Manufacturer. This code has 7-bit ID (7'b100_1110) and 1 MSB for parity bit.

- PeripheralID2[3] = 1'b1 (It uses JEDEC assigned value)
- PeripheralID2[2:0] = 3'b100
- PeripheralID1[7:4] = 4'b1110

Other register bits can be zeros.

Multi-Drop ID.

Multi-Drop feature is newly added in[13].

ROM Table can contain all debug components addresses or other ROM table base addresses.

Figure 13-7 illustrates the ROM table entries and address.

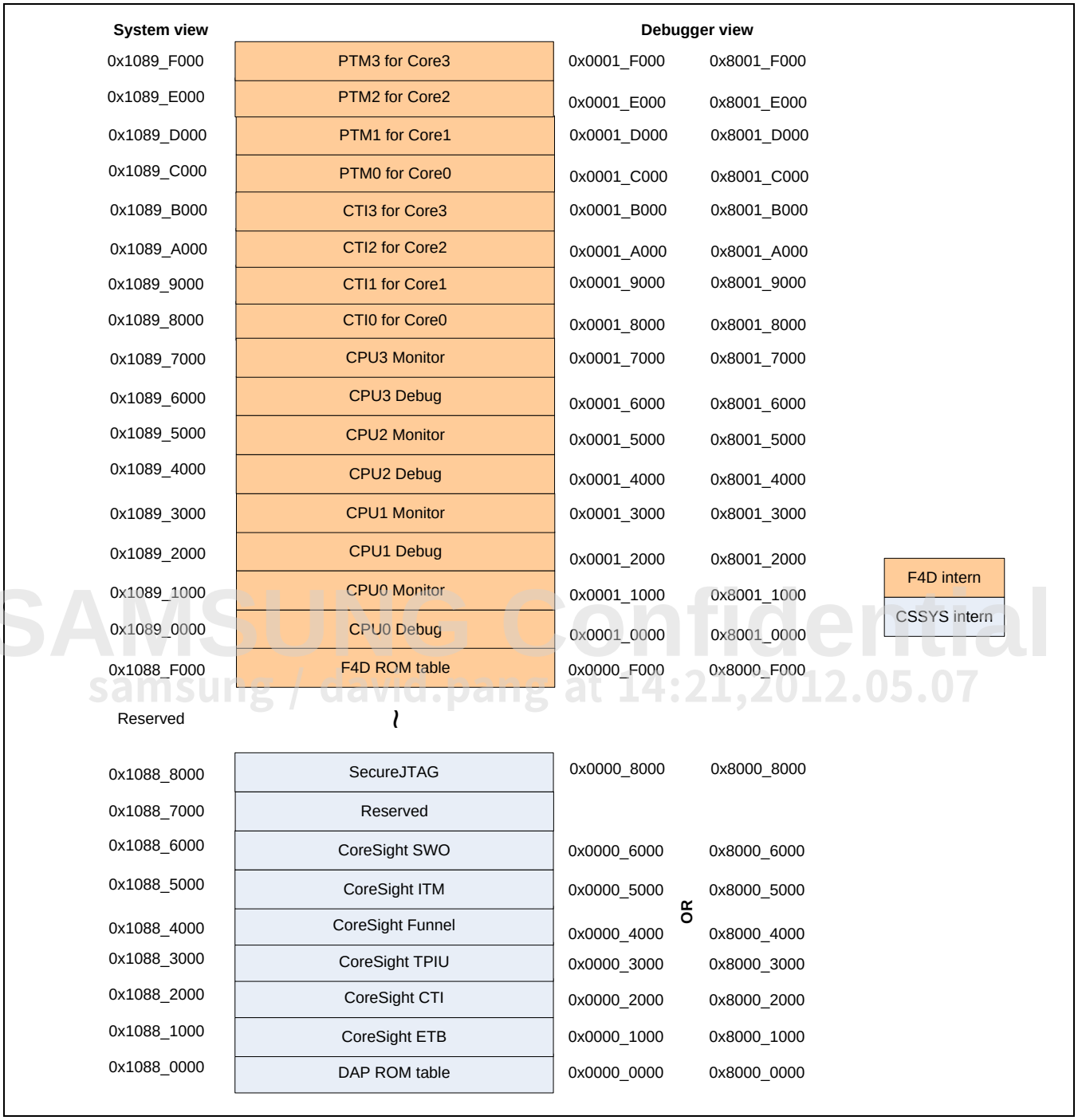


Figure 13-7 ROM Table Entries and Address

In the case of two Cortex-A9 processors, PADDRDBG[13] is used to select the processor that is accessible.

- 0x1089_0000: CPU0 access
- 0x1089_2000: CPU1 access

- 0x1089_4000: CPU2 access
- 0x1089_6000: CPU3 access

13.3 Clock Description

Table 13-1 describes the coreSight clock.

Table 13-1 CoreSight Clock

Clock	Description	Max. Clock
ATCLK	AMBA Trace BUS (ATB) clock	200 MHz
CSTCK	Generated clock by DAP (JTAG-AP) to drive JTAG interface other components	Does not use
CTICLK	Cross trigger interface clock. It can be synchronous or asynchronous to CTMCLK. CTICLK is connected to ATCLK	200 MHz (See F4Qi)
DAPCLK	Debug Access Port (DAP) internal clock. It should be equivalent to PCLKDBG. DAPCLK is connected to PCLK_DBG	200 MHz
CTMCLK	Cross trigger matrix clock. It can be synchronous or asynchronous to CTICLK CTMCLK is connected to ATCLK	200 MHz
HCLK	System facing AHB clock used by the DAP (AHB-AP). It is asynchronous to DAPCLK. HCLK is connected to ATCLK	200 MHz
PCLKDBG	Debug APB clock. It should be synchronous (equivalent or slower) to ATCLK. PCLK_DBG is connected to PCLK_DBG	200 MHz
PCLKSYS	System slave facing APB clock used by the DAP (APB-Mux). It can be asynchronous to DAPCLK. PCLKSYS is connected to PCLK_ACP	100 MHz
TCK	JTAG-DP TAP state machine clock. It is asynchronous to DAPCLK.	Not periodic
TRACECLKIN	ATCLK is divided by 2 and TRACECLKIN is connected to the divided clock. Nevertheless, TRACECLKIN has asynchronous relation with ATCLK.	100 MHz

ATCLK generates CTICLK, CTMCLK and HCLK clocks. For example CTICLK, CTMCLK, and HCLK have same frequency with ATCLK.

You can refer to CMU chapter to set the frequency of ATCLK, PCLK_DBG, and PCLK_ACP clocks.

PCLKDBG is equivalent to DAPCLK.
PCLKDBG, DAPCLK, and ATCLK should be synchronous.
PCLKDBG is less than or equal to ATCLK.
HCLK and PCLKSYS in a typical system are equivalent.

13.3.1 PCLKENDBG Generation

Connect ATCLK to ATCLK and PCLKDBG clock inputs on a component.
Generate a clock enable term that is derived from ATCLK and connect this to PCLKENDBG.

Figure 13-8 illustrates the CoreSight clock domain interactions.

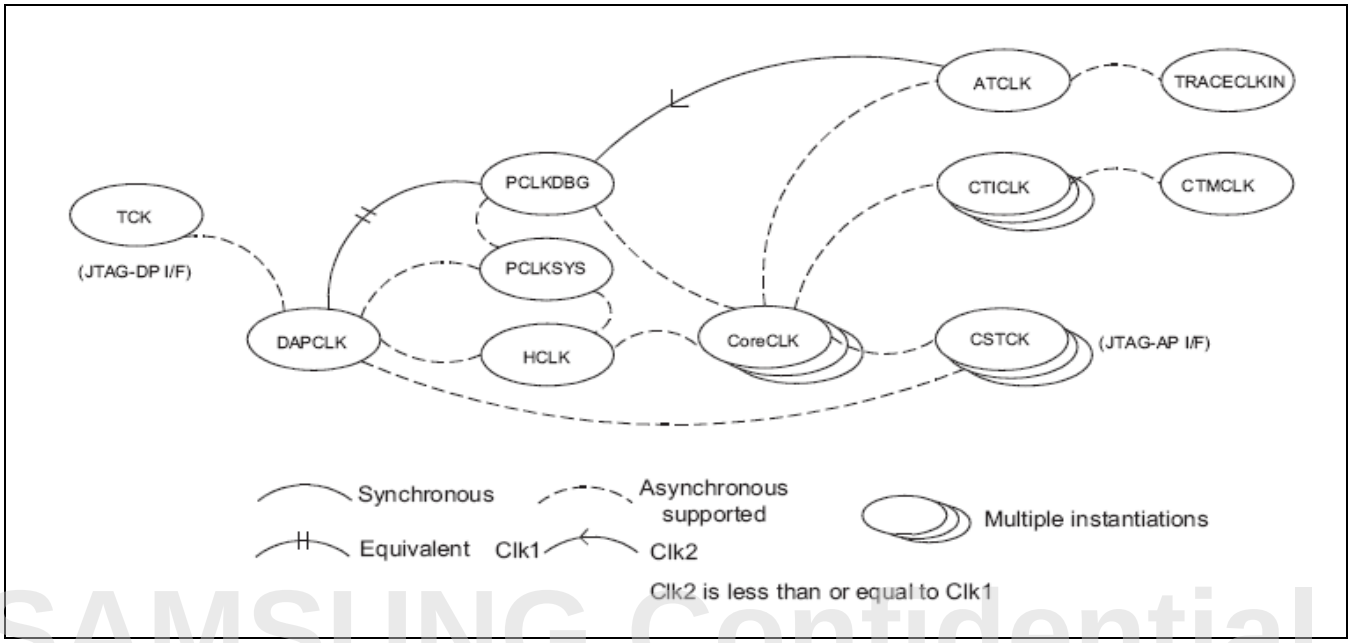


Figure 13-8 CoreSight Clock Domain Interactions

13.4 Reset

Table 13-2 describes the CoreSight reset.

Table 13-2 CoreSight Reset

Reset	Description	
ATRESETn	ATB reset. It resets all registers in the ATCLK domain. Active low.	–
nCTIRESET	CTI reset. It resets all registers clocked by CTICLK. Active low	See F4Q
DAPRESETn	DAP internal reset. Active low	HRESETn
HRESETn	SoC provided reset signal that reset all of the AMBA-on-chip interconnects.	–
nCSTRST	Internally generated reset signal that JTAG-AP controls and generates to reset TAP controller on connected components.	Does not use
nCTMRESET	CTM reset signal. It resets all registers that CTMCLK clocks. Active low.	–
nPORTRST	A true power on reset signal to the DAP JTAG-DP. It should only reset at power-on. Active low.	–
nTRST	JTAG-DP TAP state machine clock. Asynchronous to DAPCLK. Optional reset. It initializes Embedded-ICE logic. If it needs to keep the existing breakpoints and watchpoints, use reset using TCK and TMS. Refer to Multi-ICE and RealView ICE user guides for more information	–
PRESETDBGn	Debug APB reset. It resets all registers that PCLKDBG clocks. Active low.	–
PRESETSYSn	DAP APB-MUX reset signal that resets the APB slave input.	–
TRESETn	TPIU trace input reset signal. Active low.	–

13.5 Power

CoreSight system includes these power domains:

- **System Domain:**
  This is the domain in which most non-debug functionality resides. The clock frequencies in this domain can vary over time to respond to varying performance requirements. It stops the clocks altogether and it also removes the power that leads to the loss of all states.

- **Debug Domain:**
  This is the domain in which most debug functionality resides. The clock frequencies in this domain should not vary over time. When it does not require debug functionality, you can turn-off the power or stop the clock. This results in reduced power consumption.

- **Always on Domain:**
  In this domain power controller and debugger resides. Even when the power is dormant, the power is never removed. This allows the debugger to connect to the device even when it is powered down.

13.6 I/O Description

Signal	I/O	Description	Pad	Type
TRACECLK	O	TPIU out clock, is generated by TRACECLKIN (Same freq. with TRACECLKIN)	XispPCLK	Muxed
TRACECTL	O	TPIU control signal	XispRGB[8]	Muxed
TRACEDATA[0:15]	O	TPIU trace data[0:15]	XispRGB[0:7] XispRGB[9:13] XispVSYNC XispHSYNC XispMCLK	Muxed
nTRST	I	External JTAG and serial wire RESET	XjTRSTn	Dedicated
SWDITMS or SWDO	I/O	External JTAG TMS input or serial wire data output Debug mode selects function	XjTMS	Dedicated
SWCLKTCK	I	External JTAG data and serial wire clock	XjTCK	Dedicated
TDI	I	External JTAG data input	XjTDI	Dedicated
TDO or TRACESWO	O	External JTAG data output (TDO) or serial wire data output in JTAG mode (TRACESWO) Debug mode selects function	XjTDO	Dedicated

14 TrustZone Protection Controller (TZPC)

14.1 Overview

The TrustZone Protection Controller (TZPC) provides a software interface to the protection bits in a secure system in a TrustZone design. It configures each area of memory as secure or non-secure.

14.1.1 Features of TZPC

Features of TZPC are:

- Protection bits-Enables you to program a maximum of 32 areas of memory as secure or non-secure
- Secure region bits-Enables you to split an area of internal RAM into both secure and non-secure regions

TZPC includes AMBA APB system interface.

14.1.2 Block Diagram

Figure 14-1 illustrates the block diagram of TZPC.

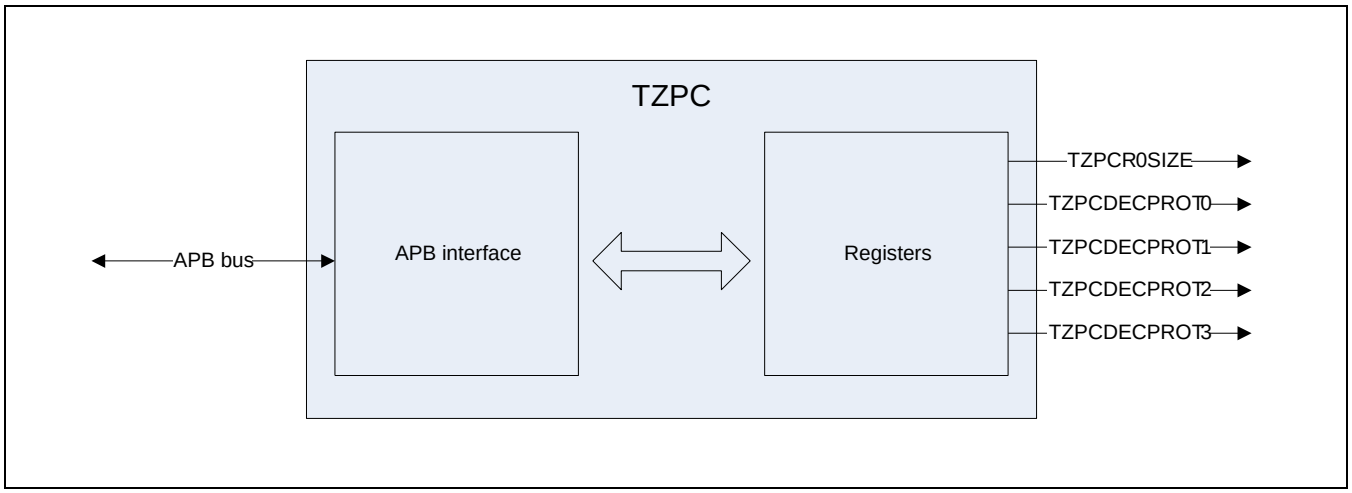


Figure 14-1 Block Diagram of Access Controller (TZPC)

14.2 Functional Description

TZPC provides a software interface to set up memory areas as secure or non-secure. It does this in two ways:

- Programmable protection bits that can be allocated to areas of memory as determined by an external decoder.
- Programmable region size value for use by an AXI TrustZone Memory Adapter (TZMA). You can use this to split the iRAM into two regions:
  - Secure
  - Non-secure

Figure 14-2 illustrates how to program the secure/non-secure region of internal SRAM. The internal ROM is always in secure area.

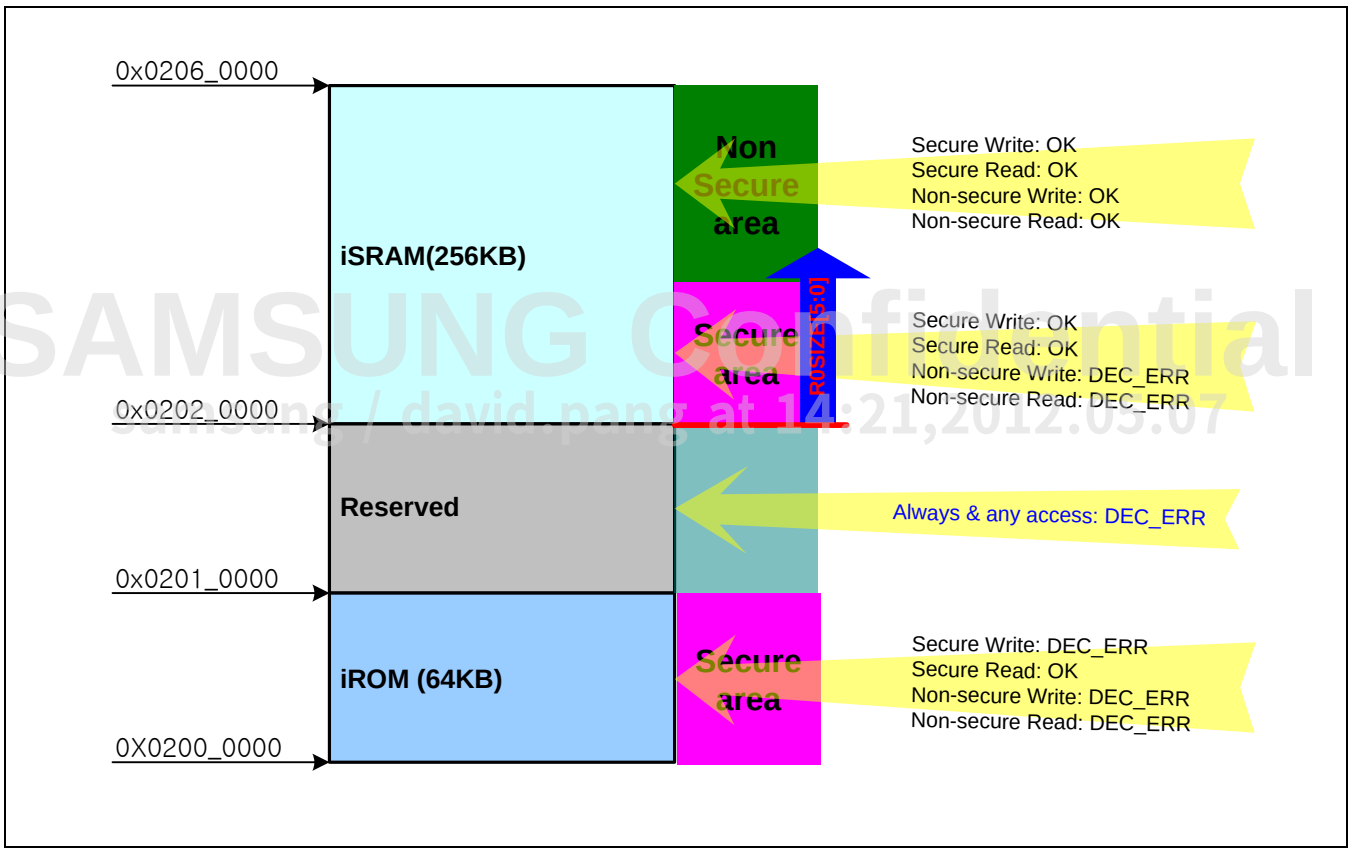


Figure 14-2 Secure Configuration of Internal Memory

The three types of masters in the secure function are:

- S/NS active master: CPU
- Non-secure master: PDMA
- S/NS inherited master: All other masters

The four types of masters in the slave function are:

- Non-secure slave: PDMA
- Secure Slave: SECKEY, TZPC*, iROM
- Address-configurable slave: iRAM.
- Port-configurable slave: All other slaves. (including AudioSS SRAM)

These masters enable the best use of memory and other system resources. You can determine that the specific and non-specific requirements of an application during:

- Boot-up
- OS or secure kernel port development work

This means that the secure and non-secure memory partitioning do not change dynamically during normal software operation as it is fixed at the time of compilation. You can configure the secure and non-secure memory partitioning only once during system boot-up. Ensure to boot up in secure-state to guarantee full security protection.

14.3 TZPC Configuration

Table 14-1 describes the TZPC configuration.

Table 14-1 TZPC Table

Register	Bit	TZPC0	TZPC1	TZPC2
		Module Name	Module Name	Module Name
DECPROT0	[7]	RTC	AXI_MAUDIOX	–
	[6]	WDT	–	–
	[5]	MCT	ASYNCAXI_MAUDIO	–
	[4]	HDMI_CEC	PPMU_ACP	SMMUJPEG
	[3]	CMU	–	SMMUFIMC3
	[2]	PMU	–	SMMUFIMC2
	[1]	SYSREG	SMMUSSS	SMMUFIMC1
	[0]	CHIPID	SMMUG2D_ACP	SMMUFIMC0
DECPROT1	[7]	GIC_distributor	ASYNCAXI_MFC_R	PPMU_CAMIF
	[6]	GIC_controller	ASYNCAXI_FSYSD	–
	[5]	IEM_APC	ASYNCAXI_ISP	–
	[4]	IEM_IEC	ASYNCAXI_LCD0	–
	[3]	KEYIF	ASYNCAXI_CAMIF	(SMMU, AS_A)_GPS
	[2]	Int_Combiner	AXI_GPR	–
	[1]	DMC1	AXI_GDR	PPMU_GPS
	[0]	DMC0	GPIO_right	GPS (When "0", All GPS bus related IP set to secure)
DECPROT2	[7]	TMU	–	PPMU_LCD0
	[6]	TZASC_EN_(LR, LW, RR, RW)	PPMU_left	SMMUFIMD0
	[5]	AS_A_(G_letft, G_right, C2C, F4D)	ASYNCAXI_MFC_L	AXI_LCD0X
	[4]	(AP, GPIO)_C2C	ASYNCAXI_TV	–
	[3]	–	ASYNCAXI_G3D	MIPI_DSI0
	[2]	AXI_Core_P	AXI_GPL	–
	[1]	TZASC_SEC_LOCK_(LR, LW, RR, RW)	AXI_GDL	MIE0
	[0]	–	GPIO_left	FIMD0
DECPROT3	[7]	G2D_ACP	AXI_CAMX	FIMD0_M0
	[6]	AXI_ACPX	MIPI_CSI1	FIMD0_M1
	[5]	Coresight	MIPI_CSI0	–
	[4]	SSS	JPEG	–

Register	Bit	TZPC0	TZPC1	TZPC2
		Module Name	Module Name	Module Name
	[3]	PPMU_CPU	FIMC3	ISP_TZinfo_3
	[2]	PPMU_DMC1	FIMC2	ISP_TZinfo_2
	[1]	PPMU_DMC0	FIMC1	ISP_TZinfo_1
	[0]	–	FIMC0	ISP_TZinfo_0

Register	Bit	TZPC3	TZPC4	TZPC5
		Module Name	Module Name	Module Name
DECPROT0	[7]	USBHOST0	MIXER_M	UART4
	[6]	MIPI HSI	PPMU_TV	UART3
	[5]	SDMMC4	SMMUTV	UART2
	[4]	SDMMC3	AXI_TVX	UART1
	[3]	SDMMC2	HDMI	UART0
	[2]	SDMMC1	VP_M	–
	[1]	SDMMC0	Mixer	–
	[0]	TSI	VP	I2CHDMI
DECPROT1	[7]	PPMU_FSYS	PPMU_MFC_R	I2C7
	[6]	AXI_FSYSS	PPMU_MFC_L	I2C6
	[5]	AXI_FSYSD	SMMUMFC_R	I2C5
	[4]	–	SMMUMFC_L	I2C4
	[3]	–	MFC	I2C3
	[2]	–	–	I2C2
	[1]	USBOTG	PPMU_3D	I2C1
	[0]	USBHOST1	G3D	I2C0
DECPROT2	[7]	SMMUMDMA	–	FIMC3_M
	[6]	–	PCM2	FIMC2_M
	[5]	–	PCM1	FIMC1_M
	[4]	ASYNC_GPS_FSYSD	PWMTimer	FIMC0_M
	[3]	–	–	–
	[2]	–	Slimbus	SROMC
	[1]	PDMA1	SPDIF	–
	[0]	PDMA0	AC97	AUDIOSS
DECPROT3	[7]	PPMU_IMAGE	I2S2	–
	[6]	–	I2S1	–
	[5]	–	–	–
	[4]	SMMURotator	SPI2	–

Register	Bit	TZPC3	TZPC4	TZPC5
		Module Name	Module Name	Module Name
	[3]	–	SPI1	SECJTAG_SPNIDEN
	[2]	AXI_IMGX	SPI0	SECJTAG_SPIDEN
	[1]	Rotator	MFC_R_M	SECJTAG_NIDEN
	[0]	–	MFC_L_M	SECJTAG_DBGEN

NOTE: If the non-secure master accesses to secure slave area, DECERR occurs.

Table 14-2 describes the TZPC transfer attribute.

Table 14-2 TZPC Transfer Attribute

Master Attribute	Transfer Attribute	Slave/Area Attribute	Response
Secure master	Secure transfer	Secure slave/area	OK
	Secure transfer	Non-secure slave/area	OK
	Non-secure transfer	Secure slave/area	DECERR
	Non-secure transfer	Non-secure slave/Area	OK

14.4 Register Description

14.4.1 Registers Map Summary

- Base Address: 0x1011_0000 (TZPC0)

Register	Offset	Description	Reset Value
R0SIZE	0x0000	Specifies the secure RAM region size register	0x0000_0040
DECPROT0Stat	0x0800	Specifies the decode protection 0 status register	0x0000_0000
DECPROT0Set	0x0804	Specifies the decode protection 0 set register	Undefined
DECPROT0Clr	0x0808	Specifies the decode protection 0 clear register	Undefined
DECPROT1Stat	0x080C	Specifies the decode protection 1 status register	0x0000_0000
DECPROT1Set	0x0810	Specifies the decode protection 1 set register	Undefined
DECPROT1Clr	0x0814	Specifies the decode protection 1 clear register	Undefined
DECPROT2Stat	0x0818	Specifies the decode protection 2 status register	0x0000_0000
DECPROT2Set	0x081C	Specifies the decode protection 2 set register	Undefined
DECPROT2Clr	0x0820	Specifies the decode protection 2 clear register	Undefined
DECPROT3Stat	0x0824	Specifies the decode protection 3 status register	0x0000_0000
DECPROT3Set	0x0828	Specifies the decode protection 3 set register	Undefined
DECPROT3Clr	0x082C	Specifies the decode protection 3 clear register	Undefined
PERIPHID0	0x0FE0	Specifies the TZPC peripheral identification register 0	0x0000_0070
PERIPHID1	0x0FE4	Specifies the TZPC peripheral identification register 1	0x0000_0018
PERIPHID2	0x0FE8	Specifies the TZPC peripheral identification register 2	0x0000_0004
PERIPHID3	0x0FEC	Specifies the TZPC peripheral identification register 3	0x0000_0000
PCELLID0	0x0FF0	Specifies the TZPC identification register 0	0x0000_000D
PCELLID1	0x0FF4	Specifies the TZPC identification register 1	0x0000_00F0
PCELLID2	0x0FF8	Specifies the TZPC identification register 2	0x0000_0005
PCELLID3	0x0FFC	Specifies the TZPC identification register 3	0x0000_00B1

- Base Address: 0x1012_0000 (TZPC1)

Register	Offset	Description	Reset Value
DECPROT0Stat	0x0800	Specifies the decode protection 0 status register	0x0000_0000
DECPROT0Set	0x0804	Specifies the decode protection 0 set register	Undefined
DECPROT0Clr	0x0808	Specifies the decode protection 0 clear register	Undefined
DECPROT1Stat	0x080C	Specifies the decode protection 1 status register	0x0000_0000
DECPROT1Set	0x0810	Specifies the decode protection 1 set register	Undefined
DECPROT1Clr	0x0814	Specifies the decode protection 1 clear register	Undefined
DECPROT2Stat	0x0818	Specifies the decode protection 2 status register	0x0000_0000
DECPROT2Set	0x081C	Specifies the decode protection 2 set register	Undefined
DECPROT2Clr	0x0820	Specifies the decode protection 2 clear register	Undefined
DECPROT3Stat	0x0824	Specifies the decode protection 3 status register	0x0000_0000
DECPROT3Set	0x0828	Specifies the decode protection 3 set register	Undefined
DECPROT3Clr	0x082C	Specifies the decode protection 3 clear register	Undefined
PERIPHID0	0x0FE0	Specifies the TZPC peripheral identification register 0	0x0000_0070
PERIPHID1	0x0FE4	Specifies the TZPC peripheral identification register 1	0x0000_0018
PERIPHID2	0x0FE8	Specifies the TZPC peripheral identification register 2	0x0000_0004
PERIPHID3	0x0FEC	Specifies the TZPC peripheral identification register 3	0x0000_0000
PCELLID0	0x0FF0	Specifies the TZPC identification register 0	0x0000_000D
PCELLID1	0x0FF4	Specifies the TZPC identification register 1	0x0000_00F0
PCELLID2	0x0FF8	Specifies the TZPC identification register 2	0x0000_0005
PCELLID3	0x0FFC	Specifies the TZPC identification register 3	0x0000_00B1

- Base Address: 0x1013_0000 (TZPC2)

Register	Offset	Description	Reset Value
DECPROT0Stat	0x0800	Specifies the decode protection 0 status register	0x0000_0000
DECPROT0Set	0x0804	Specifies the decode protection 0 set register	Undefined
DECPROT0Clr	0x0808	Specifies the decode protection 0 clear register	Undefined
DECPROT1Stat	0x080C	Specifies the decode protection 1 status register	0x0000_0000
DECPROT1Set	0x0810	Specifies the decode protection 1 set register	Undefined
DECPROT1Clr	0x0814	Specifies the decode protection 1 clear register	Undefined
DECPROT2Stat	0x0818	Specifies the decode protection 2 status register	0x0000_0000
DECPROT2Set	0x081C	Specifies the decode protection 2 set register	Undefined
DECPROT2Clr	0x0820	Specifies the decode protection 2 clear register	Undefined
DECPROT3Stat	0x0824	Specifies the decode protection 3 status register	0x0000_0000
DECPROT3Set	0x0828	Specifies the decode protection 3 set register	Undefined
DECPROT3Clr	0x082C	Specifies the decode protection 3 clear register	Undefined
PERIPHID0	0x0FE0	Specifies the TZPC peripheral identification register 0	0x0000_0070
PERIPHID1	0x0FE4	Specifies the TZPC peripheral identification register 1	0x0000_0018
PERIPHID2	0x0FE8	Specifies the TZPC peripheral identification register 2	0x0000_0000
PERIPHID3	0x0FEC	Specifies the TZPC peripheral identification register 3	0x0000_0004
PCELLID0	0x0FF0	Specifies the TZPC identification register 0	0x0000_000D
PCELLID1	0x0FF4	Specifies the TZPC identification register 1	0x0000_00F0
PCELLID2	0x0FF8	Specifies the TZPC identification register 2	0x0000_0005
PCELLID3	0x0FFC	Specifies the TZPC identification register 3	0x0000_00B1

- Base Address: 0x1014_0000 (TZPC3)

Register	Offset	Description	Reset Value
DECPROT0Stat	0x0800	Specifies the decode protection 0 status register	0x0000_0000
DECPROT0Set	0x0804	Specifies the decode protection 0 set register	Undefined
DECPROT0Clr	0x0808	Specifies the decode protection 0 clear register	Undefined
DECPROT1Stat	0x080C	Specifies the decode protection 1 status register	0x0000_0000
DECPROT1Set	0x0810	Specifies the decode protection 1 set register	Undefined
DECPROT1Clr	0x0814	Specifies the decode protection 1 clear register	Undefined
DECPROT2Stat	0x0818	Specifies the decode protection 2 status register	0x0000_0000
DECPROT2Set	0x081C	Specifies the decode protection 2 set register	Undefined
DECPROT2Clr	0x0820	Specifies the decode protection 2 clear register	Undefined
DECPROT3Stat	0x0824	Specifies the decode protection 3 status register	0x0000_0000
DECPROT3Set	0x0828	Specifies the decode protection 3 set register	Undefined
DECPROT3Clr	0x082C	Specifies the decode protection 3 clear register	Undefined
PERIPHID0	0x0FE0	Specifies the TZPC peripheral identification register 0	0x0000_0070
PERIPHID1	0x0FE4	Specifies the TZPC peripheral identification register 1	0x0000_0018
PERIPHID2	0x0FE8	Specifies the TZPC peripheral identification register 2	0x0000_0004
PERIPHID3	0x0FEC	Specifies the TZPC peripheral identification register 3	0x0000_0000
PCELLID0	0x0FF0	Specifies the TZPC identification register 0	0x0000_000D
PCELLID1	0x0FF4	Specifies the TZPC identification register 1	0x0000_00F0
PCELLID2	0x0FF8	Specifies the TZPC identification register 2	0x0000_0005
PCELLID3	0x0FFC	Specifies the TZPC identification register 3	0x0000_00B1

- Base Address: 0x1015_0000 (TZPC4)

Register	Offset	Description	Reset Value
DECPROT0Stat	0x0800	Specifies the decode protection 0 status register	0x0000_0000
DECPROT0Set	0x0804	Specifies the decode protection 0 set register	Undefined
DECPROT0Clr	0x0808	Specifies the decode protection 0 clear register	Undefined
DECPROT1Stat	0x080C	Specifies the decode protection 1 status register	0x0000_0000
DECPROT1Set	0x0810	Specifies the decode protection 1 set register	Undefined
DECPROT1Clr	0x0814	Specifies the decode protection 1 clear register	Undefined
DECPROT2Stat	0x0818	Specifies the decode protection 2 status register	0x0000_0000
DECPROT2Set	0x081C	Specifies the decode protection 2 set register	Undefined
DECPROT2Clr	0x0820	Specifies the decode protection 2 clear register	Undefined
DECPROT3Stat	0x0824	Specifies the decode protection 3 status register	0x0000_0000
DECPROT3Set	0x0828	Specifies the decode protection 3 set register	Undefined
DECPROT3Clr	0x082C	Specifies the decode protection 3 clear register	Undefined
PERIPHID0	0x0FE0	Specifies the TZPC peripheral identification register 0	0x0000_0070
PERIPHID1	0x0FE4	Specifies the TZPC peripheral identification register 1	0x0000_0018
PERIPHID2	0x0FE8	Specifies the TZPC peripheral identification register 2	0x0000_0004
PERIPHID3	0x0FEC	Specifies the TZPC peripheral identification register 3	0x0000_0000
PCELLID0	0x0FF0	Specifies the TZPC identification register 0	0x0000_000D
PCELLID1	0x0FF4	Specifies the TZPC identification register 1	0x0000_00F0
PCELLID2	0x0FF8	Specifies the TZPC identification register 2	0x0000_0005
PCELLID3	0x0FFC	Specifies the TZPC identification register 3	0x0000_00B1

- Base Address: 0x1016_0000 (TZPC5)

Register	Offset	Description	Reset Value
DECPROT0Stat	0x0800	Specifies the decode protection 0 status register	0x0000_0000
DECPROT0Set	0x0804	Specifies the decode protection 0 set register	Undefined
DECPROT0Clr	0x0808	Specifies the decode protection 0 clear register	Undefined
DECPROT1Stat	0x080C	Specifies the decode protection 1 status register	0x0000_0000
DECPROT1Set	0x0810	Specifies the decode protection 1 set register	Undefined
DECPROT1Clr	0x0814	Specifies the decode protection 1 clear register	Undefined
DECPROT2Stat	0x0818	Specifies the decode protection 2 status register	0x0000_0000
DECPROT2Set	0x081C	Specifies the decode protection 2 set register	Undefined
DECPROT2Clr	0x0820	Specifies the decode protection 2 clear register	Undefined
DECPROT3Stat	0x0824	Specifies the decode protection 3 status register	0x0000_0000
DECPROT3Set	0x0828	Specifies the decode protection 3 set register	Undefined
DECPROT3Clr	0x082C	Specifies the decode protection 3 clear register	Undefined
PERIPHID0	0x0FE0	Specifies the TZPC peripheral identification register 0	0x0000_0070
PERIPHID1	0x0FE4	Specifies the TZPC peripheral identification register 1	0x0000_0018
PERIPHID2	0x0FE8	Specifies the TZPC peripheral identification register 2	0x0000_0004
PERIPHID3	0x0FEC	Specifies the TZPC peripheral identification register 3	0x0000_0000
PCELLID0	0x0FF0	Specifies the TZPC identification register 0	0x0000_000D
PCELLID1	0x0FF4	Specifies the TZPC identification register 1	0x0000_00F0
PCELLID2	0x0FF8	Specifies the TZPC identification register 2	0x0000_0005
PCELLID3	0x0FFC	Specifies the TZPC identification register 3	0x0000_00B1

14.4.2 TZPC0 Registers

14.4.2.1 R0SIZE

- Base Address: 0x1011_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	0
R0Size	[5:0]	RW	Secure RAM region size in 4 KB steps: 0x00000000 = Non-secure region 0x00000001 = 4 KB secure region 0x00000002 = 8 KB secure region … 0x00000020 = 128 KB secure region … 0x00000040 = 256 KB secure region	0x40

14.4.2.2 DECPROTnStat

- Base Address: 0x1011_0000
- DECPROT0Stat Address = Base Address + 0x0800, Reset Value = 0x0000_0000
- DECPROT1Stat Address = Base Address + 0x080C, Reset Value = 0x0000_0000
- DECPROT2Stat Address = Base Address + 0x0818, Reset Value = 0x0000_0000
- DECPROT3Stat Address = Base Address + 0x0824, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
DECPROTnStat	[7:0]	R	Shows the status of the decode protection output: 0 = Decode region corresponding to the bit is secure 1 = Decode region corresponding to the bit is non-secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	0x000

14.4.2.3 DECPROTnSet

- Base Address: 0x1011_0000
- DECPROT0Set Address = Base Address + 0x0804, Reset Value = Undefined
- DECPROT1Set Address = Base Address + 0x0810, Reset Value = Undefined
- DECPROT2Set Address = Base Address + 0x081C, Reset Value = Undefined
- DECPROT3Set Address = Base Address + 0x0828, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
DECPROTnSet	[7:0]	W	Sets the corresponding decode protection output: 0 = No effect 1 = Sets decode region to non-secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	–

14.4.2.4 DECPROTnClr

- Base Address: 0x1011_0000
- DECPROT0Clr Address = Base Address + 0x0808, Reset Value = Undefined
- DECPROT1Clr Address = Base Address + 0x0814, Reset Value = Undefined
- DECPROT2Clr Address = Base Address + 0x0820, Reset Value = Undefined
- DECPROT3Clr Address = Base Address + 0x082C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
DECPROTnClr	[7:0]	W	Clears the corresponding decode protection output: 0 = No effect 1 = Sets decode region to secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	–

14.4.2.5 PERIPHID0

- Base Address: 0x1011_0000
- Address = Base Address + 0x0FE0, Reset Value = 0x0000_0070

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Partnumber0	[7:0]	R	These bits read back as 0x70	0x70

14.4.2.6 PERIPHID1

- Base Address: 0x1011_0000
- Address = Base Address + 0x0FE4, Reset Value = 0x0000_0018

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Designer0	[7:4]	R	These bits read back as 0x1	0x1
Partnumber1	[3:0]	R	These bits read back as 0x8	0x8

14.4.2.7 PERIPHID2

- Base Address: 0x1011_0000
- Address = Base Address + 0x0FE8, Reset Value = 0x0000_0004

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Revision	[7:4]	R	These bits read back as the revision number. The range of revision numbers can be 0-15	0x0
Designer1	[3:0]	R	These bits read back as 0x4	0x4

14.4.2.8 PERIPHID3

- Base Address: 0x1011_0000
- Address = Base Address + 0x0FEC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Configuration	[7:0]	R	These bits read back as 0x00	0x0

14.4.2.9 PCELLID0

- Base Address: 0x1011_0000
- Address = Base Address + 0x0FF0, Reset Value = 0x0000_000D

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID0	[7:0]	R	These bits read back as 0x0D	0x0D

14.4.2.10 PCELLID1

- Base Address: 0x1011_0000
- Address = Base Address + 0x0FF4, Reset Value = 0x0000_00F0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID1	[7:0]	R	These bits read back as 0xF0	0xF0

14.4.2.11 PCELLID2

- Base Address: 0x1011_0000
- Address = Base Address + 0x0FF8, Reset Value = 0x0000_0005

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID2	[7:0]	R	These bits read back as 0x05	0x05

14.4.2.12 PCELLID3

- Base Address: 0x1011_0000
- Address = Base Address + 0x0FFC, Reset Value = 0x0000_00B1

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID3	[7:0]	R	These bits read back as 0x00	0x00

14.4.3 TZPC1 Registers

14.4.3.1 DECPROTnStat

- Base Address: 0x1012_0000
- DECPROT0Stat Address = Base Address + 0x0800, Reset Value = 0x0000_0000
- DECPROT1Stat Address = Base Address + 0x080C, Reset Value = 0x0000_0000
- DECPROT2Stat Address = Base Address + 0x0818, Reset Value = 0x0000_0000
- DECPROT3Stat Address = Base Address + 0x0824, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
DECPROTnStat	[7:0]	R	Shows the status of the decode protection output: 0 = Decode region corresponding to the bit is secure 1 = Decode region corresponding to the bit is non-secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	0x000

14.4.3.2 DECPROTnSet

- Base Address: 0x1012_0000
- DECPROT0Set Address = Base Address + 0x0804, Reset Value = Undefined
- DECPROT1Set Address = Base Address + 0x0810, Reset Value = Undefined
- DECPROT2Set Address = Base Address + 0x081C, Reset Value = Undefined
- DECPROT3Set Address = Base Address + 0x0828, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
DECPROTnSet	[7:0]	W	Sets the corresponding decode protection output: 0 = No effect 1 = Sets decode region to non-secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	–

14.4.3.3 DECPROTnClr

- Base Address: 0x1012_0000
- DECPROT0Clr Address = Base Address + 0x0808, Reset Value = Undefined
- DECPROT1Clr Address = Base Address + 0x0814, Reset Value = Undefined
- DECPROT2Clr Address = Base Address + 0x0820, Reset Value = Undefined
- DECPROT3Clr Address = Base Address + 0x082C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
DECPROTnClr	[7:0]	W	Clears the corresponding decode protection output: 0 = No effect 1 = Sets decode region to secure There is one bit of the register for each protection output; eight outputs are implemented as standard.	–

14.4.3.4 PERIPHID0

- Base Address: 0x1012_0000
- Address = Base Address + 0x0FE0, Reset Value = 0x0000_0070

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Partnumber0	[7:0]	R	These bits read back as 0x70	0x70

14.4.3.5 PERIPHID1

- Base Address: 0x1012_0000
- Address = Base Address + 0x0FE4, Reset Value = 0x0000_0018

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Designer0	[7:4]	R	These bits read back as 0x1	0x1
Partnumber1	[3:0]	R	These bits read back as 0x8	0x8

14.4.3.6 PERIPHID2

- Base Address: 0x1012_0000
- Address = Base Address + 0x0FE8, Reset Value = 0x0000_0004

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Revision	[7:4]	R	These bits read back as the revision number. The range of revision numbers can be 0-15	0x0
Designer1	[3:0]	R	These bits read back as 0x4	0x4

14.4.3.7 PERIPHID3

- Base Address: 0x1012_0000
- Address = Base Address + 0x0FEC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Configuration	[7:0]	R	These bits read back as 0x00	0x0

14.4.3.8 PCELLID0

- Base Address: 0x1012_0000
- Address = Base Address + 0x0FF0, Reset Value = 0x0000_000D

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID0	[7:0]	R	These bits read back as 0x0D	0x0D

14.4.3.9 PCELLID1

- Base Address: 0x1012_0000
- Address = Base Address + 0x0FF4, Reset Value = 0x0000_00F0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID1	[7:0]	R	These bits read back as 0xF0	0xF0

14.4.3.10 PCELLID2

- Base Address: 0x1012_0000
- Address = Base Address + 0x0FF8, Reset Value = 0x0000_0005

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID2	[7:0]	R	These bits read back as 0x05	0x05

14.4.3.11 PCELLID3

- Base Address: 0x1012_0000
- Address = Base Address + 0x0FFC, Reset Value = 0x0000_00B1

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID3	[7:0]	R	These bits read back as 0x00	0x00

14.4.4 TZPC2 Registers

14.4.4.1 DECPROTnStat

- Base Address: 0x1013_0000
- DECPROT0Stat Address = Base Address + 0x0800, Reset Value = 0x0000_0000
- DECPROT1Stat Address = Base Address + 0x080C, Reset Value = 0x0000_0000
- DECPROT2Stat Address = Base Address + 0x0818, Reset Value = 0x0000_0000
- DECPROT3Stat Address = Base Address + 0x0824, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
DECPROTnStat	[7:0]	R	Shows the status of the decode protection output: 0 = Decode region corresponding to the bit is secure 1 = Decode region corresponding to the bit is non-secure There is one bit of the register for each protection output; eight outputs are implemented as standard.	0x000

14.4.4.2 DECPROTnSet

- Base Address: 0x1013_0000
- DECPROT0Set Address = Base Address + 0x0804, Reset Value = Undefined
- DECPROT1Set Address = Base Address + 0x0810, Reset Value = Undefined
- DECPROT2Set Address = Base Address + 0x081C, Reset Value = Undefined
- DECPROT3Set Address = Base Address + 0x0828, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
DECPROTnSet	[7:0]	W	Sets the corresponding decode protection output: 0 = No effect 1 = Sets decode region to non-secure There is one bit of the register for each protection output; eight outputs are implemented as standard.	–

14.4.4.3 DECPROTnClr

- Base Address: 0x1013_0000
- DECPROT0Clr Address = Base Address + 0x0808, Reset Value = Undefined
- DECPROT1Clr Address = Base Address + 0x0814, Reset Value = Undefined
- DECPROT2Clr Address = Base Address + 0x0820, Reset Value = Undefined
- DECPROT3Clr Address = Base Address + 0x082C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
DECPROTnClr	[7:0]	W	Clears the corresponding decode protection output: 0 = No effect 1 = Sets decode region to secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	–

14.4.4.4 PERIPHID0

- Base Address: 0x1013_0000
- Address = Base Address + 0x0FE0, Reset Value = 0x0000_0070

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Partnumber0	[7:0]	R	These bits read back as 0x70	0x70

14.4.4.5 PERIPHID1

- Base Address: 0x1013_0000
- Address = Base Address + 0x0FE4, Reset Value = 0x0000_0018

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Designer0	[7:4]	R	These bits read back as 0x1	0x1
Partnumber1	[3:0]	R	These bits read back as 0x8	0x8

14.4.4.6 PERIPHID2

- Base Address: 0x1013_0000
- Address = Base Address + 0x0FE8, Reset Value = 0x0000_0004

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Revision	[7:4]	R	These bits read back as the revision number. The range of revision numbers can be 0-15	0x0
Designer1	[3:0]	R	These bits read back as 0x4	0x4

14.4.4.7 PERIPHID3

- Base Address: 0x1013_0000
- Address = Base Address + 0x0FEC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Configuration	[7:0]	R	These bits read back as 0x00	0x0

14.4.4.8 PCELLID0

- Base Address: 0x1013_0000
- Address = Base Address + 0x0FF0, Reset Value = 0x0000_000D

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID0	[7:0]	R	These bits read back as 0x0D	0x0D

14.4.4.9 PCELLID1

- Base Address: 0x1013_0000
- Address = Base Address + 0x0FF4, Reset Value = 0x0000_00F0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID1	[7:0]	R	These bits read back as 0xF0	0xF0

14.4.4.10 PCELLID2

- Base Address: 0x1013_0000
- Address = Base Address + 0x0FF8, Reset Value = 0x0000_0005

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID2	[7:0]	R	These bits read back as 0x05	0x05

14.4.4.11 PCELLID3

- Base Address: 0x1013_0000
- Address = Base Address + 0x0FFC, Reset Value = 0x0000_00B1

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID3	[7:0]	R	These bits read back as 0x00	0x00

14.4.5 TZPC3 Registers

14.4.5.1 DECPROTnStat

- Base Address: 0x1014_0000
- DECPROT0Stat Address = Base Address + 0x0800, Reset Value = 0x0000_0000
- DECPROT1Stat Address = Base Address + 0x080C, Reset Value = 0x0000_0000
- DECPROT2Stat Address = Base Address + 0x0818, Reset Value = 0x0000_0000
- DECPROT3Stat Address = Base Address + 0x0824, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
DECPROTnStat	[7:0]	R	Shows the status of the decode protection output: 0 = Decode region corresponding to the bit is secure 1 = Decode region corresponding to the bit is non-secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	0x000

14.4.5.2 DECPROTnSet

- Base Address: 0x1014_0000
- DECPROT0Set Address = Base Address + 0x0804, Reset Value = Undefined
- DECPROT1Set Address = Base Address + 0x0810, Reset Value = Undefined
- DECPROT2Set Address = Base Address + 0x081C, Reset Value = Undefined
- DECPROT3Set Address = Base Address + 0x0828, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
DECPROTnSet	[7:0]	W	Sets the corresponding decode protection output: 0 = No effect 1 = Sets decode region to non-secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	–

14.4.5.3 DECPROTnClr

- Base Address: 0x1014_0000
- DECPROT0Clr Address = Base Address + 0x0808, Reset Value = Undefined
- DECPROT1Clr Address = Base Address + 0x0814, Reset Value = Undefined
- DECPROT2Clr Address = Base Address + 0x0820, Reset Value = Undefined
- DECPROT3Clr Address = Base Address + 0x082C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
DECPROTnClr	[7:0]	W	Clears the corresponding decode protection output: 0 = No effect 1 = Sets decode region to secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	–

14.4.5.4 PERIPHID0

- Base Address: 0x1014_0000
- Address = Base Address + 0x0FE0, Reset Value = 0x0000_0070

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Partnumber0	[7:0]	R	These bits read back as 0x70	0x70

14.4.5.5 PERIPHID1

- Base Address: 0x1014_0000
- Address = Base Address + 0x0FE4, Reset Value = 0x0000_0018

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Designer0	[7:4]	R	These bits read back as 0x1	0x1
Partnumber1	[3:0]	R	These bits read back as 0x8	0x8

14.4.5.6 PERIPHID2

- Base Address: 0x1014_0000
- Address = Base Address + 0x0FE8, Reset Value = 0x0000_0004

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Revision	[7:4]	R	These bits read back as the revision number. The range of revision numbers can be 0-15	0x0
Designer1	[3:0]	R	These bits read back as 0x4	0x4

14.4.5.7 PERIPHID3

- Base Address: 0x1014_0000
- Address = Base Address + 0x0FEC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Configuration	[7:0]	R	These bits read back as 0x00	0x0

14.4.5.8 PCELLID0

- Base Address: 0x1014_0000
- Address = Base Address + 0x0FF0, Reset Value = 0x0000_000D

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID0	[7:0]	R	These bits read back as 0x0D	0x0D

14.4.5.9 PCELLID1

- Base Address: 0x1014_0000
- Address = Base Address + 0x0FF4, Reset Value = 0x0000_00F0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID1	[7:0]	R	These bits read back as 0xF0	0xF0

14.4.5.10 PCELLID2

- Base Address: 0x1014_0000
- Address = Base Address + 0x0FF8, Reset Value = 0x0000_0005

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID2	[7:0]	R	These bits read back as 0x05	0x05

14.4.5.11 PCELLID3

- Base Address: 0x1014_0000
- Address = Base Address + 0x0FFC, Reset Value = 0x0000_00B1

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID3	[7:0]	R	These bits read back as 0x00	0x00

14.4.6 TZPC4 Registers

14.4.6.1 DECPROTnStat

- Base Address: 0x1015_0000
- DECPROT0Stat Address = Base Address + 0x0800, Reset Value = 0x0000_0000
- DECPROT1Stat Address = Base Address + 0x080C, Reset Value = 0x0000_0000
- DECPROT2Stat Address = Base Address + 0x0818, Reset Value = 0x0000_0000
- DECPROT3Stat Address = Base Address + 0x0824, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
DECPROTnStat	[7:0]	R	Shows the status of the decode protection output: 0 = Decode region corresponding to the bit is secure 1 = Decode region corresponding to the bit is non-secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	0x000

14.4.6.2 DECPROTnSet

- Base Address: 0x1015_0000
- DECPROT0Set Address = Base Address + 0x0804, Reset Value = Undefined
- DECPROT1Set Address = Base Address + 0x0810, Reset Value = Undefined
- DECPROT2Set Address = Base Address + 0x081C, Reset Value = Undefined
- DECPROT3Set Address = Base Address + 0x0828, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
DECPROTnSet	[7:0]	W	Sets the corresponding decode protection output: 0 = No effect 1 = Sets decode region to non-secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	–

14.4.6.3 DECPROTnClr

- Base Address: 0x1015_0000
- DECPROT0Clr Address = Base Address + 0x0808, Reset Value = Undefined
- DECPROT1Clr Address = Base Address + 0x0814, Reset Value = Undefined
- DECPROT2Clr Address = Base Address + 0x0820, Reset Value = Undefined
- DECPROT3Clr Address = Base Address + 0x082C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
DECPROTnClr	[7:0]	W	Clears the corresponding decode protection output: 0 = No effect 1 = Sets decode region to secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	–

14.4.6.4 PERIPHID0

- Base Address: 0x1015_0000
- Address = Base Address + 0x0FE0, Reset Value = 0x0000_0070

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Partnumber0	[7:0]	R	These bits read back as 0x70	0x70

14.4.6.5 PERIPHID1

- Base Address: 0x1015_0000
- Address = Base Address + 0x0FE4, Reset Value = 0x0000_0018

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Designer0	[7:4]	R	These bits read back as 0x1	0x1
Partnumber1	[3:0]	R	These bits read back as 0x8	0x8

14.4.6.6 PERIPHID2

- Base Address: 0x1015_0000
- Address = Base Address + 0x0FE8, Reset Value = 0x0000_0004

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Revision	[7:4]	R	These bits read back as the revision number. The range of revision numbers can be 0-15	0x0
Designer1	[3:0]	R	These bits read back as 0x4	0x4

14.4.6.7 PERIPHID3

- Base Address: 0x1015_0000
- Address = Base Address + 0x0FEC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Configuration	[7:0]	R	These bits read back as 0x00	0x0

14.4.6.8 PCELLID0

- Base Address: 0x1015_0000
- Address = Base Address + 0x0FF0, Reset Value = 0x0000_000D

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID0	[7:0]	R	These bits read back as 0x0D	0x0D

14.4.6.9 PCELLID1

- Base Address: 0x1015_0000
- Address = Base Address + 0x0FF4, Reset Value = 0x0000_00F0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID1	[7:0]	R	These bits read back as 0xF0	0xF0

14.4.6.10 PCELLID2

- Base Address: 0x1015_0000
- Address = Base Address + 0x0FF8, Reset Value = 0x0000_0005

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID2	[7:0]	R	These bits read back as 0x05	0x05

14.4.6.11 PCELLID3

- Base Address: 0x1015_0000
- Address = Base Address + 0x0FFC, Reset Value = 0x0000_00B1

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID3	[7:0]	R	These bits read back as 0x00	0x00

14.4.7 TZPC5 Registers

14.4.7.1 DECPROTnStat

- Base Address: 0x1016_0000
- DECPROT0Stat Address = Base Address + 0x0800, Reset Value = 0x0000_0000
- DECPROT1Stat Address = Base Address + 0x080C, Reset Value = 0x0000_0000
- DECPROT2Stat Address = Base Address + 0x0818, Reset Value = 0x0000_0000
- DECPROT3Stat Address = Base Address + 0x0824, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
DECPROTnStat	[7:0]	R	Shows the status of the decode protection output: 0 = Decode region corresponding to the bit is secure 1 = Decode region corresponding to the bit is non-secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	0x000

14.4.7.2 DECPROTnSet

- Base Address: 0x1016_0000
- DECPROT0Set Address = Base Address + 0x0804, Reset Value = Undefined
- DECPROT1Set Address = Base Address + 0x0810, Reset Value = Undefined
- DECPROT2Set Address = Base Address + 0x081C, Reset Value = Undefined
- DECPROT3Set Address = Base Address + 0x0828, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
DECPROTnSet	[7:0]	W	Sets the corresponding decode protection output: 0 = No effect 1 = Sets decode region to non-secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	–

14.4.7.3 DECPROTnClr

- Base Address: 0x1016_0000
- DECPROT0Clr Address = Base Address + 0x0808, Reset Value = Undefined
- DECPROT1Clr Address = Base Address + 0x0814, Reset Value = Undefined
- DECPROT2Clr Address = Base Address + 0x0820, Reset Value = Undefined
- DECPROT3Clr Address = Base Address + 0x082C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
DECPROTnClr	[7:0]	W	Clears the corresponding decode protection output: 0 = No effect 1 = Sets decode region to secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	–

14.4.7.4 PERIPHID0

- Base Address: 0x1016_0000
- Address = Base Address + 0x0FE0, Reset Value = 0x0000_0070

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Partnumber0	[7:0]	R	These bits read back as 0x70	0x70

14.4.7.5 PERIPHID1

- Base Address: 0x1016_0000
- Address = Base Address + 0x0FE4, Reset Value = 0x0000_0018

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Designer0	[7:4]	R	These bits read back as 0x1	0x1
Partnumber1	[3:0]	R	These bits read back as 0x8	0x8

14.4.7.6 PERIPHID2

- Base Address: 0x1016_0000
- Address = Base Address + 0x0FE8, Reset Value = 0x0000_0004

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Revision	[7:4]	R	These bits read back as the revision number. The range of revision numbers can be 0-15	0x0
Designer1	[3:0]	R	These bits read back as 0x4	0x4

14.4.7.7 PERIPHID3

- Base Address: 0x1016_0000
- Address = Base Address + 0x0FEC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Configuration	[7:0]	R	These bits read back as 0x00	0x0

14.4.7.8 PCELLID0

- Base Address: 0x1016_0000
- Address = Base Address + 0x0FF0, Reset Value = 0x0000_000D

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID0	[7:0]	R	These bits read back as 0x0D	0x0D

14.4.7.9 PCELLID1

- Base Address: 0x1016_0000
- Address = Base Address + 0x0FF4, Reset Value = 0x0000_00F0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID1	[7:0]	R	These bits read back as 0xF0	0xF0

14.4.7.10 PCELLID2

- Base Address: 0x1016_0000
- Address = Base Address + 0x0FF8, Reset Value = 0x0000_0005

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID2	[7:0]	R	These bits read back as 0x05	0x05

14.4.7.11 PCELLID3

- Base Address: 0x1016_0000
- Address = Base Address + 0x0FFC, Reset Value = 0x0000_00B1

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PCELLID3	[7:0]	R	These bits read back as 0x00	0x00

15 TrustZone Address Access Controller (TZASC)

15.1 Overview

The TrustZone Address Space Controller (TZASC) is an Advanced Microcontroller Bus Architecture (AMBA) compliant System-on-Chip (SoC) peripheral. It is a high-performance, area-optimized address space controller with on-chip AMBA bus interfaces that conform to the AMBA Advanced eXtensible Interface (AXI) protocol and the AMBA Advanced Peripheral Bus (APB) protocol.

You can configure the TZASC to provide the optimum security address region control functions required for your intended application. See Features of the TZASC for a summary of the configurable features supported.

15.1.1 Features of the TZASC

The TZASC provides the following features:

- Enables you to program security access permissions for each address region
- Permits the transfer of data between master and slave only if the security status of the AXI transaction matches the security settings of the memory region it addresses
- Prevents write access to various registers after assertion of secure_boot_lock.
- Supports 4 address regions

15.1.2 Block Diagram

Figure 15-1 illustrates the block diagram of access controller.

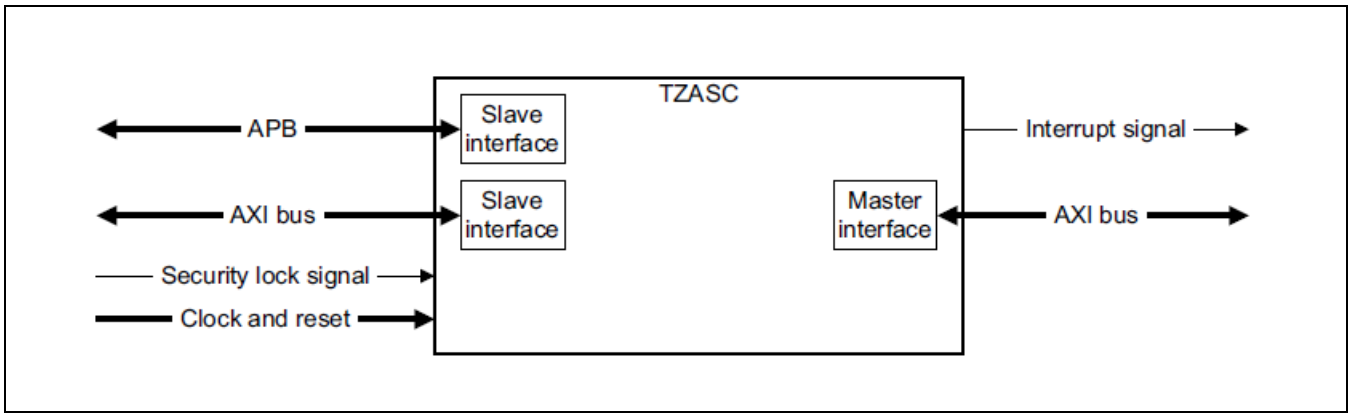


Figure 15-1 Block Diagram of Access Controller (TZASC)

15.2 Functional Description

The functional description section includes:

- Functional interfaces
- Functional operation
- Constraints of use

15.2.1 Functional Interfaces

The main interfaces of the TZASC are:

- AXI bus interfaces
- APB slave interface
- Miscellaneous signals
- Clock and reset

15.2.1.1 AXI Bus Interfaces

The TZASC provides the following AXI bus interfaces:

- AXI slave interface
- AXI master interface

Each AXI bus interface consists of the following AXI channels:

- Write Address (AW)
- Write Data (W)
- Write Response (B)
- Read Address (AR)
- Read Data (R).

15.2.1.1.1 AXI Slave Interface

Table 15-1 describes the AXI slave interface attributes and their values.

Table 15-1 AXI Slave Interface Attributes

Attributes	Value
Read acceptance capability	Equals the configurable transaction tracker queue depth. If register slices are enabled on AR channel, the read acceptance capability is configurable transaction tracker queue depth plus one.
Write acceptance capability	Equals the configurable transaction tracker queue depth. If register slices are enabled on AW channel, the write acceptance capability is configurable transaction tracker queue depth plus one.
Combined acceptance capability	Equals the configurable transaction tracker queue depth. If register slices are enabled on AR and AW channels, the combined acceptance capability is configurable transaction tracker queue depth plus two.
Write interleave depth	1
Read data reordering depth	Equal to zero. The TZASC does not re-order read data. However, the TZASC supports the re-ordering depth of the downstream slave.

15.2.1.1.2 AXI Master Interface

Table 15-2 describes the AXI master interface attributes and their values.

Table 15-2 AXI Master Interface Attributes

Attributes	Value
Combined issuing capability	Equals the transaction tracking queue depth that is configurable
Write issuing capability	Equals the transaction tracking queue depth that is configurable
Read issuing capability	Equals the transaction tracking queue depth that is configurable

15.2.1.2 APB Slave Interface

The APB slave interfaces provide access to the TZASC registers that enables you to program the system configuration parameters and obtain status information. See *15.3 Register Description*.

NOTE: There are two ways to set up memory areas as secure or non-secure. The APB slave interface should only be accessible to processors in secure state. Otherwise it can compromise the security of the system.

15.2.1.3 Miscellaneous Signals

There are two miscellaneous signals:

- secure_boot_lock
- tzasc_int

Figure 15-2 illustrates the miscellaneous signals that the TZASC provides.

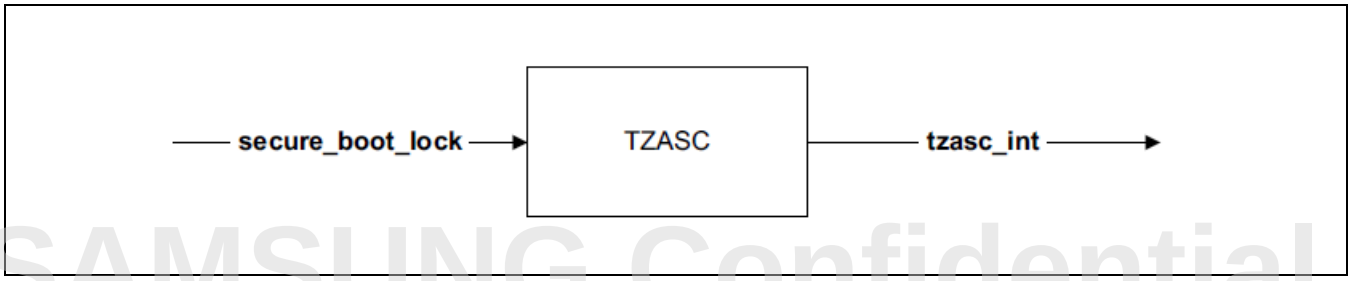


Figure 15-2 Miscellaneous Signals

Asserting secure_boot_lock enhances the security of the TZASC. See *15.2.2.8 Preventing Writes to Registers and using secure_boot_lock*.

You can program the TZASC to assert tzasc_int when it denies an AXI master access to a region. See *15.2.2.6 Denied AXI Transactions.*

15.2.1.4 Clock and Reset

This section describes:

- Clock
- Reset
- Pclken

15.2.1.4.1 Clock

All configurations of the TZASC use a single clock input, aclk.

15.2.1.4.2 Reset

The TZASC provides a single reset input, aresetn.

15.2.1.4.3 pclken

Clock enable signal that enables the APB slave interface to operate at either:

- The aclk frequency, or
- A divided integer multiple of aclk that is synchronous to aclk.

NOTE: If you do not use pclken, you must tie it HIGH. This results in the APB slave interface being clocked directly by aclk.

15.2.2 Functional Operation

TZASC is a systems IP that performs security checks on AXI accesses to memory or off-chip peripheral. This supports configurable number of regions. Each region is programmable for size, base address, enable, and security parameters. Using the secure_boot_lock, the programmers view can be locked to prevent erroneous writes. See *15.2.2.8 Preventing Writes to Registers and using secure_boot_lock*. The IP provides programmability in reporting faults using AXI response channel and interrupt.

Figure 15-3 illustrates the functional operation of TZASC.

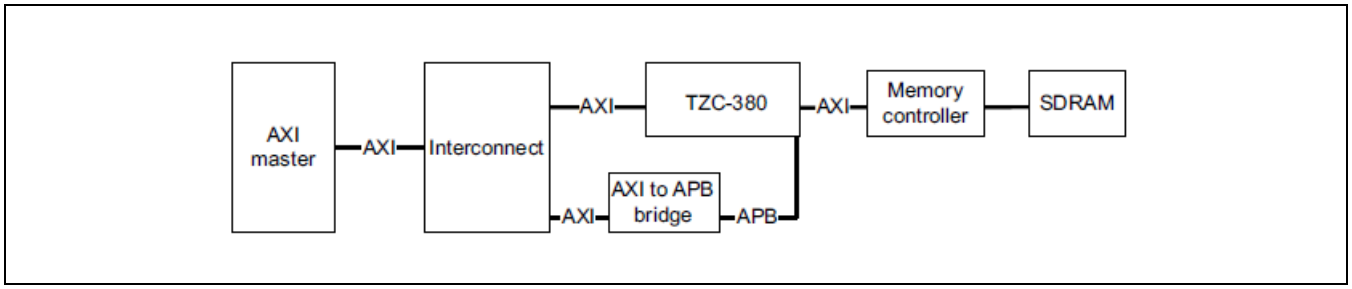


Figure 15-3 Functional Operation of TZASC

This section describes:

- Regions
- Priority
- Subregions
- Subregion disable
- Region security permissions
- Denied AXI transactions
- Speculative accesses
- Preventing writes to registers and using secure_boot_lock
- Using locked transaction sequences
- Using exclusive accesses

15.2.2.1 Regions

A region is a contiguous area of address space. The TZASC provides each region with a programmable security permissions field. The security permissions value is used to enable the TZASC to either accept or deny a transaction access to that region. The transactions arprots[2:0] or awprots[2:0] signals are used to determine the security settings of that transaction.

The TZASC always provides two regions- region 0 and region 1, and you can configure it to provide additional regions. With the exception of region 0, the TZASC enables you to program the following operating parameters for each region.

The programmable operating parameters are:

- Region enable
- Security permissions
- Base address
- Size. The minimum address size of a region is 32 KB.
- Subregion disable. See *15.2.2.3 Subregions*.

NOTE: Region 0 is known as the background region because it occupies the total memory space. You can program the security permissions of region 0 but the following parameters are fixed:

Base address :0x0

Size :The AXI_ADDRESS_MSB configuration parameter controls the address range of the TZASC, and therefore the region size.

Subregion disable :This feature is not available for region 0.

15.2.2.2 Priority

The priority of a region is fixed and is determined by the region number.

Figure 15-4 illustrates how the priority of a region increases with the region number.

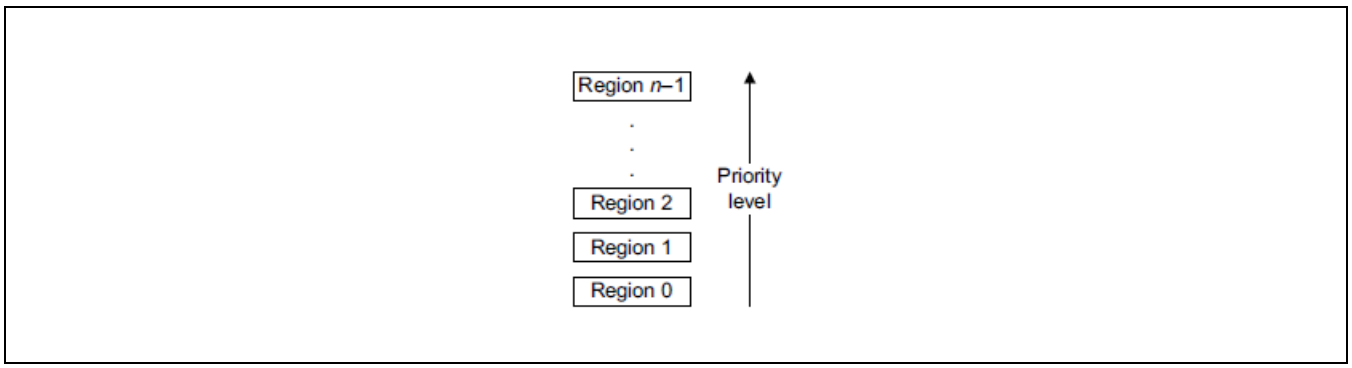


Figure 15-4 Region Priority

When a transaction is received, its address is checked for a match with all the configured regions in turn. The order in which the regions are checked is determined by the priority level, the highest priority level is first. The first region that matches the transaction address match is used as the matching region. The matching regions security permission determines whether the transaction is permitted.

15.2.2.3 Subregions

The TZASC divides each region into eight equal-sized, non-overlapping subregions.

Figure 15-5 illustrates the subregions for an example region that is programmed to occupy an address span of 32 KB.

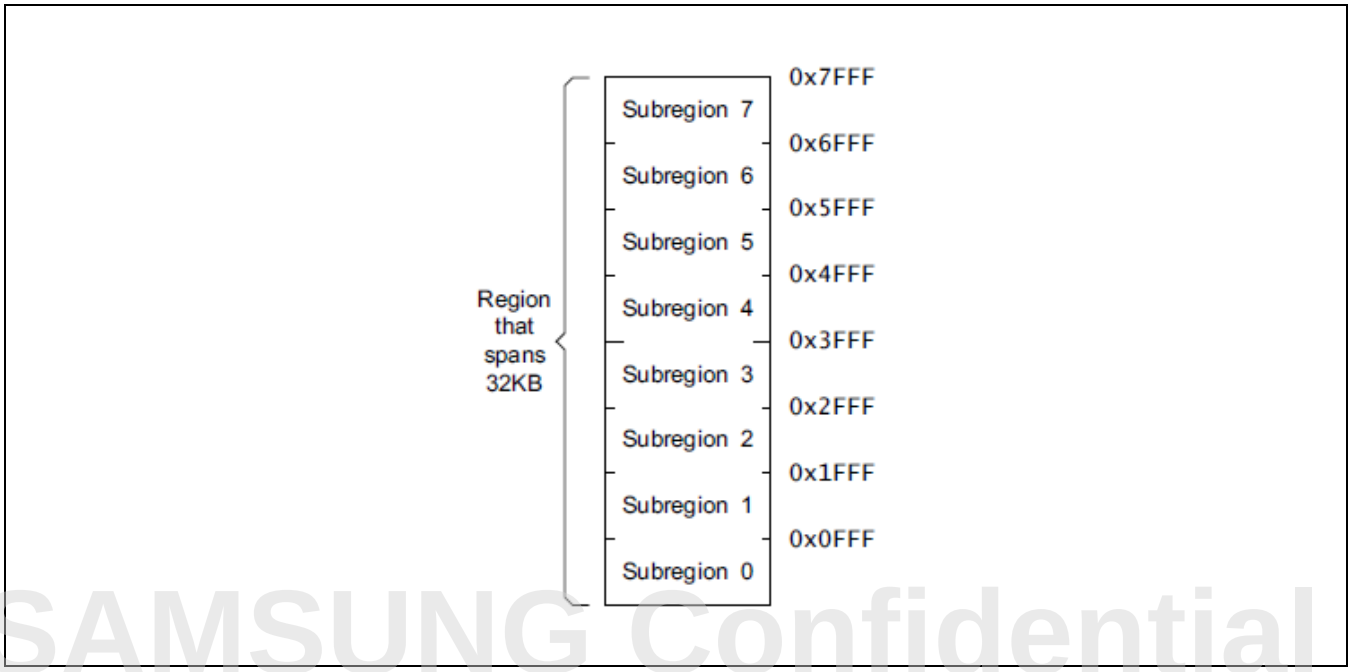


Figure 15-5 Subregion Example

15.2.2.4 Subregion Disable

With the exception of region 0, you can program the TZASC to disable any or all of the eight subregions that comprise a region. When a subregion is disabled, the security permissions for its address range are provided by the next highest priority region that overlaps the address range.

Example 15-1 Example Configuration for Subregion Disable

Figure 15-6 illustrates an example configuration that supports four regions. ,where:
• Region 2 and region 3 are partially overlapped.
• Region 1 and region 3 are partially overlapped.
• Region 0 is overlapped with all regions.
With some subregions of region 1, region 2, and region 3 are disabled, and the resulting region permissions of the entire address space is shown in the *Figure 15-6*.

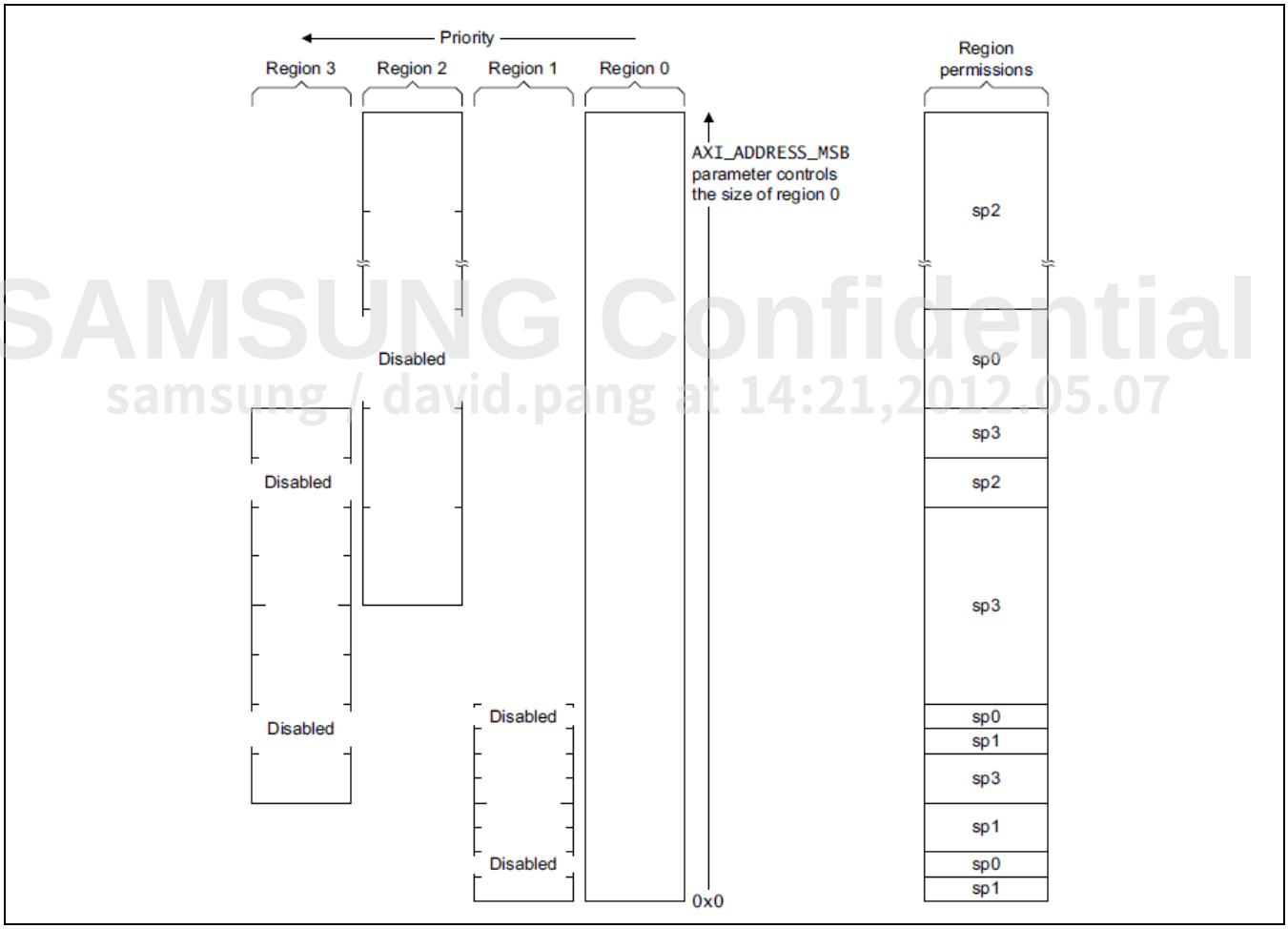


Figure 15-6 Subregion Disable Example

NOTE: In *Figure 15-6:*

1. All subregions are enabled unless otherwise stated
2. spn represents the region permissions of region n.

15.2.2.5 Region Security Permissions

The TZASC enables you to program the security access permissions for any region that it is configured. A region is assigned a security permissions field, sp <n>, in its region_attributes_<n> Register that enables you to have complete control of the permissions for that region.

15.2.2.5.1 Security Inversion

There are two modes of operation for the region security permissions:

- With security inversion
- Without security inversion.

By default, if you program a region to support non-secure accesses, the TZASC ensures that region must also support secure accesses. For example, if you program the region permissions for region 3 to be non-secure read only, the TZASC permits access to region 3 for secure reads and non-secure reads.

If you require that some regions are not accessible to masters in secure state, but are accessible in Non-secure state, then you must enable security inversion. See *15.2.2.5 Region Security Permissions* and *15.2.2.5.1 Security Inversion*

15.2.2.5.2 Programming Security Permissions when Security Inversion is Disabled

By default, security inversion is disabled. And therefore the TZASC only permits you to program certain combinations of security permissions. These combinations ensure that a master in secure state is not denied access to a region that is programmed to only accept non-secure accesses.

Table 15-3 lists the possible security permissions when security inversion is disabled.

Table 15-3 Region Security Permissions when Security Inversion is Disabled

–	SP <n> Field Controls If the TZASC Permits Access for the Following AXI Transactions			
SP <n> Field	**Secure Read**	**Secure Write**	**Non-secure Read**	**Non-secure Write**
b0000	No	No	No	No
b0100	No	Yes	No	No
b0001, b0101	No	Yes	No	Yes
b1000	Yes	No	No	No
b0010, b1010	Yes	No	Yes	No
b1100	Yes	Yes	No	Yes
b1001, b1101	Yes	Yes	No	Yes
b0110, b1110	Yes	Yes	Yes	No
b0011, b0111, b1011, b1111	Yes	Yes	Yes	Yes

15.2.2.5.3 Programming Security Permissions when Security Inversion is Enabled

If you enable security inversion, the TZASC permits you to program any combination of security permissions.

Table 15-4 lists the region security permissions when security inversion is enabled.

Table 15-4 Region Security Permissions when Security Inversion is Enabled

–	SP <n> Field Controls If the TZASC Permits Access for the Following AXI Transactions			
SP <n> field	**Secure Read**	**Secure Write**	**Non-secure Read**	**Non-secure Write**
b0000	No	No	No	No
b0001	No	No	No	Yes
b0010	No	No	Yes	No
b0011	No	No	Yes	Yes
b0100	No	Yes	No	No
b0101	No	Yes	No	Yes
b0110	No	Yes	Yes	No
b0111	No	Yes	Yes	Yes
b1000	Yes	No	No	No
b1001	Yes	No	No	Yes
b1010	Yes	No	Yes	No
b1011	Yes	No	Yes	Yes
b1100	Yes	Yes	No	No
b1101	Yes	Yes	No	Yes
b1110	Yes	Yes	Yes	No
b1111	Yes	Yes	Yes	Yes

Table 15-5 describes the typical example of memory map along with the register programming. The TZASC is configured to have 4 regions.

Table 15-5 Example OF Memory Map Along with the Register Programming

Region	Region	Lock	Starting Address	Region Size	Size Field	SP	Description
Region_0 (Default)	Enable	No	0x0	max	–	1100	Secure Read Write access (RW).
Region_1	Enable	No	0x0	64 MB	b011001	1111	Non-secure Read or Write access (RW) and Secure RW.
Region_2	Enable	No	0x0	16 MB	b010111	1110	Non-secure Read Only access (RO) and Secure RW for the normal world OS kernel.
Region_3	Enable	No	0x3D00000	512 KB	b010010	1111	Regularly switched Non-secure or Secure RW for a more complex shared memory buffers.

15.2.2.6 Denied AXI Transactions

If an AXI transaction has insufficient security privileges then for:

15.2.2.6.1 Reads

The TZASC responds to the master by setting all bits of the read data bus, rdatas[AXI_DATA_MSB:0], to zero.

NOTE: If the TZASC is programmed to perform speculative accesses, it discards the data that it receives on rdatam[AXI_DATA_MSB:0].

15.2.2.6.2 Writes

The TZASC prevents the transfer of data from the master to the slave by discarding the data that wdatas[AXI_DATA_MSB:0] contains. If you program the TZASC to perform speculative accesses, it modifies the transfer to the slave by setting all bits of the:

- Write data bus, wdatam[AXI_DATA_MSB:0], to zero.
- Write data strobe, wstrbm[AXI_STRB_MSB:0], to zero.

NOTE: The action Register controls whether the TZASC signals to the master when a region permission failure occurs, and if so, the type of response it provides. See *15.3.1.2 Action*.

15.2.2.7 Speculative Accesses

By default, the TZASC performs read or write speculative accesses. that means it forwards an AXI transaction address to a slave before it verifies that the AXI transaction is permitted to read address or write address respectively.

The TZASC only permits the transfer of data between its AXI bus interfaces, after verifying the access that the read or write access is permitted respectively. If the verification fails, then it prevents the transfer of data between the master and slave as *15.2.2.6 Denied AXI Transactions* describes.

You can disable speculative accesses by programming the speculation_control Register.

See *15.3.1.11 speculation_control*. When speculative accesses are disabled, the TZASC verifies the permissions of the access before it forwards the access to the slave. If the TZASC:

- Permits the access, it commences an AXI transaction to the slave and it adds one clock latency.
- Denies the access, it prevents the transfer of data between the master and slave as *15.2.2.6 Denied AXI Transactions* describes. In this situation, the slave is unaware when the TZASC prevents the master from accessing the slave.

NOTE: Enabling speculative access is a potential security risk, if the device that is being protected reacts to this transaction. Most devices do not have to react to this level of access, and speculative access is much faster than validating the address before issuing the transaction.

15.2.2.8 Preventing Writes to Registers and using secure_boot_lock

By suitably programming lockdown Register, see *15.3.1.4 lockdown_select*, and asserting secure_boot_lock signal makes the following registers read only:

- speculation_control Register. See *15.3.1.11 speculation_control*.
- security_inversion_en Register. See *15.3.1.12 security_inversion_en*.
- lockdown_range Register. See *15.3.1.3 lockdown_range*.

15.2.2.8.1 Locking Down the Region Using Lockdown_range and Lockdown_select Registers

By programming the lockdown_select, and lockdown_range registers, and asserting the secure_boot_lock signal, you can lockdown the behavior of the TZASC so that it prevents unintentional or erroneous write to the regions specified in the lockdown_range Register.

However, read access to those regions is permitted:

- region_setup_low_<n> Register. See *15.3.1.13 region_setup_low_n*.
- region_setup_high_<n> Register. See *15.3.1.14 region_setup_high_n*.
- region_attributes_<n> Register. See *15.3.1.3 lockdown_range*.

The TZASC expects the secure_boot_lock signal to be asserted for at least one clock cycle. One clock after the secure_boot_lock is sampled HIGH by TZASC, then the registers mentioned in Locking down the region using lockdown_range and *15.3.1.4 lockdown_select* cannot be written, unless the TZASC is reset by asserting aresetn.

15.2.2.9 Using Locked Transaction Sequences

If a master performs locked transaction sequences, a transaction might stall, or an AXI protocol violation might occur when:

- Transaction Sequence Crosses a 4 KB Boundary
- Secure State Change
- Reads and Writes

15.2.2.9.1 Transaction Sequence Crosses a 4 KB Boundary

If a locked transaction sequence crosses a 4 KB boundary and the regions have different region permissions, the TZASC might prevent access to the second region and therefore the slave would not receive the latter part of the locked transaction sequence.

NOTE: The AXI protocol recommends that locked transaction sequences do not cross a 4 KB address boundary.

15.2.2.9.2 Secure State Change

During a locked transaction sequence, if a master changes the state of arprots[1] or awprots[1] and the region has different region permissions for Secure state and Non-secure state, the TZASC might deny a transaction and therefore the slave would not receive the latter part of the locked transaction sequence.

15.2.2.9.3 Reads and Writes

During a locked transaction sequence, if a master performs reads and writes to a region, depending on the region permissions, the TZASC might deny a transaction and therefore the slave would not receive the latter part of the locked transaction sequence.

15.2.2.10 Using Exclusive Accesses

If a master performs exclusive accesses to an address region, you must program the TZASC to permit read and write accesses to that address region, for the expected settings of arprots[1] and awprots[1], otherwise the read or write transaction might fail.

15.2.3 Constraints of Use

The TZASC has the following considerations relating to change in programmers view on an active system:

- When changing the setting of a TZASC region,
  - The current accepted AXI transaction, if it falls into that region, would act according to the previous settings for that region.
  - Any other outstanding AXI transactions that falls into that region, would effect by the new settings for that region.
- Given little ability to predict that the mentioned AXI transactions would effect, it is obviously desirable that there are no outstanding AXI transactions when a regions setting are changed.
  - In simple systems this can potentially be achieved by the core not accessing the given region during the period of the cores transition between security states. Even in these cases, the status of cached data and instructions needs to be considered.
  - In more complicated systems the code that changes the TZASC region settings must have to inform other AXI bus masters to desist or complete acting on that region before performing the region setting changes. After having such an action acknowledged the code must also have to instigate a suitable delay before then acting.

An example of this can be an LCD controller dealing with a frame buffer that is switching between a Normal world and secure world use.

NOTE: There is no direct mechanism to ascertain if there are any outstanding AXI transactions, and so the designer must use their system knowledge to apply reasonable mechanisms.

It is recommended that any DECERR, or TZASC interrupt handler is designed to expect, and potentially ignore events generated under these circumstances.

15.3 Register Description

15.3.1 Registers Map Summary

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)

Register	Offset	Description	Reset Value
configuration	0x0000	Configuration register	0x0000_1F03
action	0x0004	Action register	0x0000_0001
lockdown_range	0x0008	Lockdown range register	0x0000_0000
lockdown_select	0x00C	Lockdown select register	0x0000_0000
int_status	0x0010	Interrupt status register	0x0000_0000
int_clear	0x0014	Interrupt clear register	0x0000_0000
fail_address_low	0x0020	Fail address low register	0x0000_0000
fail_address_high	0x0024	Fail address high register	0x0000_0000
fail_control	0x0028	Fail control register	0x0000_0000
fail_id	0x002C	Fail Id register	0x0000_0000
speculation_control	0x0030	Speculation control register	0x0000_0000
security_inversion_en	0x0034	Security inversion enable register	0x0000_0000
region_setup_low_0	0x0100	Region setup low <0> register	0x0000_0000
region_setup_low_1	0x0110	Region setup low <1> register	0x0000_0000
region_setup_low_2	0x0120	Region setup low <2> register	0x0000_0000
region_setup_low_3	0x0130	Region setup low <3> register	0x0000_0000
region_setup_high_0	0x0104	Region setup high <0> register	0x0000_0000
region_setup_high_1	0x0114	Region setup high <1> register	0x0000_0000
region_setup_high_2	0x0124	Region setup high <2> register	0x0000_0000
region_setup_high_3	0x0134	Region setup high <3> register	0x0000_0000
region_attributes_0	0x0108	Region attributes <0> register	0x0000_0000
region_attributes_1	0x0118	Region attributes <1> register	0x0000_001C
region_attributes_2	0x0128	Region attributes <2> register	0x0000_001C
region_attributes_3	0x0138	Region attributes <3> register	0x0000_001C
itcrg	0x0E00	Integration test control register	0x0000_0000
itip	0x0E04	Integration test input register	0x0000_0000
itop	0x0E08	Integration test output register	0x0000_0000
periph_id_4	0x0FD0	Peripheral identification 4 registers	0x0000_0004
periph_id_0	0x0FE0	Peripheral identification 0 registers	0x0000_0080
periph_id_1	0x0FE4	Peripheral identification 1 registers	0x0000_00B3

Register	Offset	Description	Reset Value
periph_id_2	0x0FE8	Peripheral identification 2 registers	0x0000_001B
periph_id_3	0x0FEC	Peripheral identification 3 registers	0x0000_0000
component_id_0	0x0FF0	Component identification 0 registers	0x0000_000D
component_id_1	0x0FF4	Component identification 1 registers	0x0000_00F0
component_id_2	0x0FF8	Component identification 2 registers	0x0000_0005
component_id_3	0x0FFC	Component identification 3 registers	0x0000_00B1

15.3.1.1 Configuration

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0000, Reset Value = 0x0000_1F03

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	–	Reserved	–
address_width	[13:8]	RO	Returns the width of the AXI address bus. Read as: b000000 to b011110 = Reserved b011111 = 32-bit b100000 = 33-bit b100001 = 34-bit … b111110 = 63-bit b111111 = 64-bit	0x1F
RSVD	[7:4]	–	Reserved	–
no_of_regions	[3:0]	RO	Returns the number of regions the TZASC provides: b0000 = Reserved b0001 = 2 regions b0010 = 3 regions b0011 = 4 regions	0x3

15.3.1.2 Action

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0004, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	–
reaction_value	[1:0]	RW	Controls how the TZASC uses the bresps[1:0], rresps[1:0], and tzasc_int signals when a region permission failure occurs: b00 = Sets tzasc_int LOW and issues an OKAY response. b01 = Sets tzasc_int LOW and issues a DECERR response. b10 = Sets tzasc_int HIGH and issues an OKAY response. b11 = Sets tzasc_int HIGH and issues a DECERR response.	0x1

15.3.1.3 lockdown_range

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
enable	[31]	RW	When set to 1, it enables the lockdown_regions field to control the regions that are to be locked.	0x0
RSVD	[30:4]	–	Reserved	–
lockdown_ regions	[3:0]	RW	Controls the number of regions to lockdown when the enable bit is set to 1: b0000 = Region no_of_regions–1 is locked. b0001 = Region no_of_regions–1 to region no_of_regions–2 are locked. b0010 = Region no_of_regions–1 to region no_of_regions–3 are locked. b0011 = Region no_of_regions–1 to region no_of_regions–4 are locked.	0x0

NOTE: The value programmed in lockdown_range Register must not be greater than no_of_regions–1 else all regions are locked.

15.3.1.4 lockdown_select

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	–
acc_speculation_cntl	[2]	RW	Modifies the access type of the speculation_control Register: 0 = No effect. speculation_control Register remains RW. 1 = Speculation_control Register is RO. See *15.3.1.11 speculation_control*.	0x0
security_inv	[1]	RW	Modifies the access type of the security_inversion_en Register: 0 = No effect. security_inversion_en Register remains RW. 1 = Security_inversion_en Register is RO. See *15.3.1.12 security_inversion_en*.	0x0
region_register	[0]	RW	Modifies the access type of the lockdown_range Register: 0 = No effect. lockdown_range Register remains RW. 1 = Lockdown_range Register is RO. See *15.3.1.3 lockdown_range*.	0x0

15.3.1.5 int_status

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	–
overrun	[1]	RO	When set to 1, it indicates the occurrence of two or more region permission failures, since the interrupt was last cleared.	0x0
status	[0]	RO	Returns the status of the interrupt: 0 = Interrupt is inactive. 1 = Interrupt is active.	0x0

15.3.1.6 int_clear

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Writing any value to the int_clear Register sets the:

- Status bit to 0 in the int_status Register
- Overrun bit to 0 in the int_status Register.

15.3.1.7 fail_address_low

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
add_status_low	[31:0]	RO	Returns the AXI address bits[31:0] of the first access to fail a region permission check after the interrupt was cleared.	0x0

15.3.1.8 fail_address_high

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31: n + 1] (NOTE)	–	Reserved	–
add_status_high	[n: 0]	RO	Returns the address bits[AXI_ADDRESS_MSB: 32] of the first access to fail a region permission check, after the interrupt was cleared	0x0

NOTE: The value of n = AXI_ADDRESS_MSB – 32.

15.3.1.9 fail_control

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	–
write	[24]	RO	This bit indicates whether the first access to fail a region permission check was a write or read as: 0 = Read access 1 = Write access	0x0
RSVD	[23:22]	–	Reserved	–
nonsecure	[21]	RO	After clearing the interrupt status, this bit indicates whether the first access to fail a region permission check was non-secure. Read as: 0 = Secure access 1 = Non-secure access	0x0
privileged	[20]	RW	After clearing the interrupt status, this bit indicates whether the first access to fail a region permission check was privileged. Read as: 0 = Unprivileged access 1 = Privileged access.	0x0
RSVD	[19:0]	–	Reserved	–

15.3.1.10 fail_id

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31: n + 1] (NOTE)	–	Reserved	–
Id	[n:0]	RO	Returns the master AXI ID of the first access to fail a region permission check, after the interrupt was cleared.	0x0

NOTE: The value of n = AID_WIDTH – 1.

15.3.1.11 speculation_control

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	–
write_speculation	[1]	RW	Controls the write access speculation: 0 = Write access speculation is enabled. This is the default. 1 = Write access speculation is disabled.	0x0
read_speculation	[0]	RW	Controls the read access speculation: 0 = Read access speculation is enabled. This is the default. 1 = Read access speculation is disabled.	0x0

15.3.1.12 security_inversion_en

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	–
Security _inversion_en	[0]	RW	Controls whether the TZASC permits security inversion to occur: 0 = Security inversion is not permitted. This is the default. 1 = Security inversion is permitted. This enables a region to be accessible to masters in Non-secure State but not accessible to masters in Secure state.	0x0

15.3.1.13 region_setup_low_n

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0100, 0x0110, 0x0120, 0x0130, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
base_address_low <n>	[31:15]	RW	Controls the base address[31:15] of region <n> (NOTE). The TZASC only permits a region to start at address 0x0, or at a multiple of its region size. For example, if the size of a region is 512 MB, and it is not at address 0x0, the only valid settings for this field are: • b00100000000000000 • b01000000000000000 • b01100000000000000 • b10000000000000000 • b10100000000000000 • b11000000000000000 • b11100000000000000 If you attempt to set an inappropriate base address for the size of the region, the TZASC ignores certain bits depending on the region size. See Table 15-6.	0x0
RSVD	[14:0]	–	Reserved	–

NOTE: For region 0, this field is Read Only (RO). The TZASC sets the base address of region 0 to 0x0.

15.3.1.14 region_setup_high_n

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0104, 0x0114, 0x0124, 0x0134, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
base_address_high <n>	[31:0]	RW	Controls the base address[63:32] of region <n> (NOTE). The TZASC only permits a region to start at address 0x0, or at a multiple of its region size. If you program a region size to be 8GB or more, then the TZASC might ignore certain bits depending on the region size. See Table 15-6.	0x0
RSVD	[14:0]	–	Reserved	–

NOTE: For region 0, this field is RO. The TZASC sets the base address of region 0 to 0x0.

15.3.1.15 region_attributes_n

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0108, 0x0118, 0x0128, 0x0138 Reset Value = 0xC000_0000, 0x0000_001C

Name	Bit	Type	Description	Reset Value
sp <n>	[31:28]	RW	Permissions setting for region <n>. If an AXI transaction occurs to region n, the value in the sp <n> field controls whether the TZASC permits the transaction to proceed. For more information, see: • *15.2.2.5.2 Programming Security Permissions when Security Inversion is Disabled* • *15.2.2.5.3 Programming Security Permissions when Security Inversion is Enabled*	0x0
RSVD	[27:16]	–	Reserved	–
subregion_ disable <n> (NOTE)	[15:8]	RW	Regions are split into eight equal-sized sub-regions, and each bit enables the corresponding subregion to be disabled: Bit[15] = 1 Subregion 7 is disabled. Bit[14] = 1 Subregion 6 is disabled. Bit[13] = 1 Subregion 5 is disabled. Bit[12] = 1 Subregion 4 is disabled. Bit[11] = 1 Subregion 3 is disabled. Bit[10] = 1 Subregion 2 is disabled. Bit[9] = 1 Subregion 1 is disabled. Bit[8] = 1 Subregion 0 is disabled. For more information, see *15.2.2.3 Subregions* and *15.2.2.4 Subregion Disable*	0x0
RSVD	[7]	–	Reserved	–
size <n> (NOTE)	[6:1]	RW	Size of region <n>. See *Table 15-6.* The AXI address width, that is AXI_ADDRESS_MSB + 1, controls the upper limit value of this field.	0x0
en <n> (NOTE)	[0]	RW	Enable for region <n>: 0 = Region <n> is disabled. 1 = Region <n> is enabled.	0x0

NOTE: For region 0, this field is reserved.

Table 15-6 describes the region size.

Table 15-6 Region Size

Size <n> Field	Size of Region <n>	Base Address (NOTE) Constraints
b000000 to b001101	Reserved	–
b001110	32 KB	–
b001111	64 KB	Bit[15] must be zero
b010000	128 KB	Bits[16:15] must be zero
b010001	256 KB	Bits[17:15] must be zero
b010010	512 KB	Bits[18:15] must be zero
b010011	1 MB	Bits[19:15] must be zero
b010100	2 MB	Bits[20:15] must be zero
b010101	4 MB	Bits[21:15] must be zero
b010110	8 MB	Bits[22:15] must be zero
b010111	16 MB	Bits[23:15] must be zero
b011000	32 MB	Bits[24:15] must be zero
b011001	64 MB	Bits[25:15] must be zero
b011010	128 MB	Bits[26:15] must be zero
b011011	256 MB	Bits[27:15] must be zero
b011100	512 MB	Bits[28:15] must be zero
b011101	1 GB	Bits[29:15] must be zero
b011110	2 GB	Bits[30:15] must be zero
b011111	4 GB	Bits[31:15] must be zero
b100000	8 GB	Bits[32:15] must be zero
b100001	16 GB	Bits[33:15] must be zero
b100010	32 GB	Bits[34:15] must be zero
b100011	64 GB	Bits[35:15] must be zero
b100100	128 GB	Bits[36:15] must be zero
b100101	256 GB	Bits[37:15] must be zero
b100110	512 GB	Bits[38:15] must be zero
b100111	1 TB	Bits[39:15] must be zero
b101000	2 TB	Bits[40:15] must be zero
b101001	4 TB	Bits[41:15] must be zero
b101010	8 TB	Bits[42:15] must be zero
b101011	16 TB	Bits[43:15] must be zero
b101100	32 TB	Bits[44:15] must be zero
b101101	64 TB	Bits[45:15] must be zero
b101110	128 TB	Bits[46:15] must be zero

Size <n> Field	Size of Region <n>	Base Address (NOTE) Constraints
b101111	256 TB	Bits[47:15] must be zero
b110000	512 TB	Bits[48:15] must be zero
b110001	1 PB	Bits[49:15] must be zero
b110010	2 PB	Bits[50:15] must be zero
b110011	4 PB	Bits[51:15] must be zero
b110100	8 PB	Bits[52:15] must be zero
b110101	16 PB	Bits[53:15] must be zero
b110110	32 PB	Bits[54:15] must be zero
b110111	64 PB	Bits[55:15] must be zero
b111000	128 PB	Bits[56:15] must be zero
b111001	256 PB	Bits[57:15] must be zero
b111010	512 PB	Bits[58:15] must be zero
b111011	1 EB	Bits[59:15] must be zero
b111100	2 EB	Bits[60:15] must be zero
b111101	4 EB	Bits[61:15] must be zero
b111110	8 EB	Bits[62:15] must be zero
b111111	16 EB	Bits[63:15] must be zero

NOTE: The region_setup_low_<n> Register and region_setup_high_<n> Register contain the base address.

15.3.1.16 itcrg

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0E00, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	–
int_test_en	[0]	RW	Controls the enabling of, or provides the status of, the integration test logic: 0 = Integration test logic is disabled 1 = Integration test logic is enabled	0x0

15.3.1.17 itip

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0E04, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	–
itip_secure_boot_lock	[0]	RO	Returns the status of secure_boot_lock: 0 = secure_boot_lock is LOW 1 = secure_boot_lock is HIGH	0x0

15.3.1.18 itop

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0E08, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	–
itip_secure_boot_lock	[0]	RW	Returns the status of secure_boot_lock: 0 = secure_boot_lock is LOW 1 = secure_boot_lock is HIGH	0x0

15.3.1.19 periph_id_4

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0FD0, Reset Value = 0x0000_0004

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
4KB_count	[7:4]	R	The number of 4 KB address blocks you require to access the registers is expressed in powers of 2. These bits read back as 0x0.	0x0
jep106_c_code	[3:0]	R	The JEP106 continuation code value represents how many 0x7F continuation characters occur in the manufacturer's identity code. See JEP106, Standard Manufacturer's Identification Code. These bits read back as 0x4.	0x4

The periph_id_[4:0] Registers provide information about the configuration of the peripheral.

15.3.1.20 periph_id_0

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0FE0, Reset Value = 0x0000_0080

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
part_number_0	[7:0]	R	These bits read back as 0x80.	0x80

The periph_id_[4:0] Registers provide information about the configuration of the peripheral.

15.3.1.21 periph_id_1

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0FE4, Reset Value = 0x0000_00B3

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
jep106_id_3_0	[7:4]	R	JEP106 identity code[3:0]. See the JEP106, Standard Manufacturer's Identification Code. These bits read back as 0xB because ARM is the designer of the peripheral.	0xB
part_number_1	[3:0]	R	These bits read back as 0x3.	0x3

The periph_id_[4:0] Registers provide information about the configuration of the peripheral.

15.3.1.22 periph_id_2

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0FE8, Reset Value = 0x0000_001B

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
revision	[7:4]	R	Identifies the revision of the TZASC. For revision r0p1, this field is set to 0x1.	0x1
jedec_used	[3]	R	This indicates that the TZASC uses a manufacturer's identity code that was allocated by JEDEC according to JEP106. This bit always reads back as 0x1.	0x1
jep106_id_6_4	[2:0]	R	JEP106 identity code[6:4]. See the JEP106, Standard Manufacturer's Identification Code. These bits read back as b011 because ARM is the designer of the peripheral.	0x3

The periph_id_[4:0] Registers provide information about the configuration of the peripheral.

15.3.1.23 periph_id_3

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0FEC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
RevAnd	[7:4]	R	The top-level RTL provides four AND gates that are tied-off to provide an output value of 0x0. When silicon is available, if metal fixes are necessary, the manufacturer can modify the tie-offs to indicate that a revision of the silicon has occurred.	0x0
mod_number	[3:0]	R	You can update this field by modifying the RTL of the TZASC. ARM sets this to 0x0.	0x0

The periph_id_[4:0] Registers provide information about the configuration of the peripheral.

15.3.1.24 component_id_0

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0FF0, Reset Value = 0x0000_000D

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
component_id_0	[7:0]	R	These bits read back as 0x0D	0x0D

The component_id_0 Register is hard-coded and the fields in the register control the reset value.

The component_id_[3:0] Registers are four, 8-bit wide registers, that can conceptually be treated as a single register that holds a 32-bit Component ID value. You can use the register for automatic BIOS configuration. The component_id Register is set to 0xB105F00D.

15.3.1.25 component_id_1

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0FF4, Reset Value = 0x0000_00F0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
component_id_1	[7:0]	R	These bits read back as 0xF0	0xF0

The component_id_1 Register is hard-coded and the fields in the register control the reset value.

The component_id_[3:0] Registers are four, 8-bit wide registers, and that can conceptually be treated as a single register that holds a 32-bit Component ID value. You can use the register for automatic BIOS configuration. The component_id Register is set to 0xB105F00D.

15.3.1.26 component_id_2

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0FF8, Reset Value = 0x0000_0005

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
component_id_2	[7:0]	R	These bits read back as 0x5	0x5

The component_id_2 Register is hard-coded and the fields in the register control the reset value.

The component_id_[3:0] Registers are four, 8-bit wide registers, and that can conceptually be treated as a single register that holds a 32-bit Component ID value. You can use the register for automatic BIOS configuration. The component_id Register is set to 0xB105F00D.

15.3.1.27 component_id_3

- Base Address: 0x1070_0000 (TZASC_LR)
- Base Address: 0x1071_0000 (TZASC_LW)
- Base Address: 0x1072_0000 (TZASC_RR)
- Base Address: 0x1073_0000 (TZASC_RW)
- Address = Base Address + 0x0FFC, Reset Value = 0x0000_00B1

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
component_id_3	[7:0]	R	These bits read back as 0xB1	0xB1

The component_id_3 Register is hard-coded and the fields in the register control the reset value.

The component_id_[3:0] Registers are four, 8-bit wide registers, and that can conceptually be treated as a single register that holds a 32-bit Component ID value. You can use the register for automatic BIOS configuration. The component_id Register is set to 0xB105F00D.

16 System Memory Management Unit

16.1 Introduction

The SysMMU is an Advanced Microcontroller Bus Architecture (AMBA) Advanced eXtensible Interface (AXI) compliant address translation unit. It is responsible for handling accesses to memory subsystem that master IPs request in a SoC. It normally translates virtual page numbers to physical page numbers through an associative cache, called a Translation Lookaside Buffer (TLB). It also concatenates physical page number into the page offset to obtain a complete physical address.

16.1.1 Key Features

The features of SysMMU version 1.1 are:

- A slave port compatible with AMBA3 AXI specification for address translation
- A master port compatible with AMBA3 AXI specification
- A slave port compatible with AMBA3 Advanced Peripheral Bus (APB) specification for programming registers
- Translates 32-bit virtual addresses to 32-bit physical addresses
- Page size: 4 KB, 64 KB, 1 MB and 16 MB
- Supports identical page table entry format with ARMv7 architecture
- Supports page protection: read-only/read-write, security
- Provides Round-Robin and LRU policies for TLB entry replacement
- Interrupt: page fault, R/W protection fault, TLB multi-hit fault, Bus error
- Supports DISABLE mode to bypass AXI channels
- Zero-cycle latency on TLB hit
- The master port provides AxUSER to assign shareable bit of page table entry

16.1.2 Functional Description

This chapter provides functional overview of SysMMU and a detailed description of key functions of SysMMU: The address translation and the page table walk.

16.1.3 Functional Overview

AXI Slave Interface block accepts AXI requests from master IPs. TLB lookup operations are performed in Address Translation block on Write address channel (AW)/Read address channel (AR) addresses passing through the AXI Slave Interface. In case of a TLB hit, the Address Translation block translates virtual AW/AR address to physical AW/AR address. The AXI Master Interface block issues AXI requests equipped with the translated AW/AR physical address to memory subsystem.

In case of a TLB miss, the Page Table Walk block performs page table walking to update TLB entries. Page Table Walk block reads page tables from system working memory through the AXI Master Interface block and finally refills TLB. The operating system (OS) or middleware manages page tables that you should write and make available in working memory along with initialization of the SysMMU.

Figure 16-1 illustrates the conceptual block diagram of SysMMU.

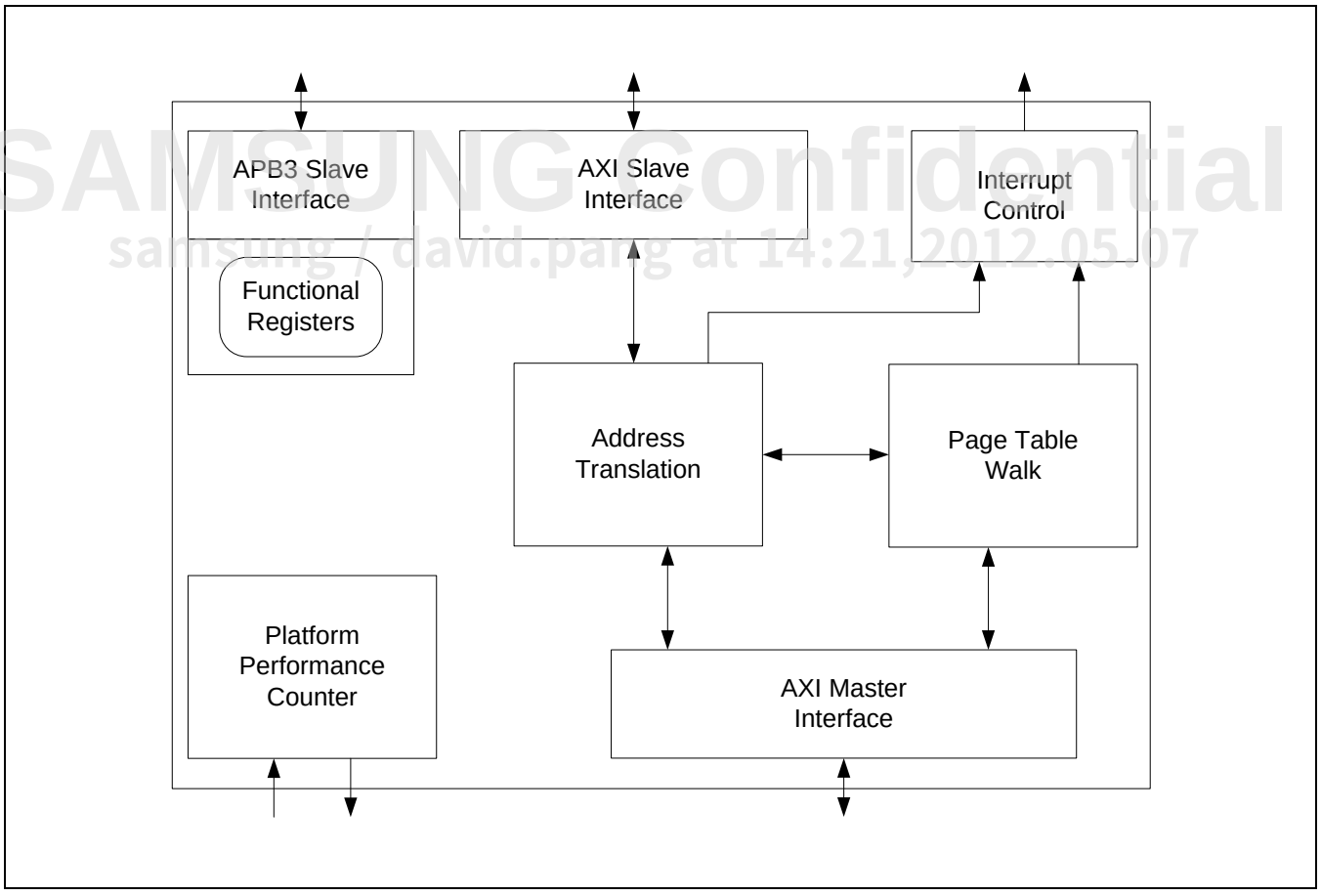


Figure 16-1 SysMMU High-level Block Diagram

The SysMMU supports four page sizes:

- 4 KB
- 64 KB
- 1 MB
- 16 MB

Defined by ARMv7 architecture.

Figure 16-2 illustrates the example of system integration with SysMMU.

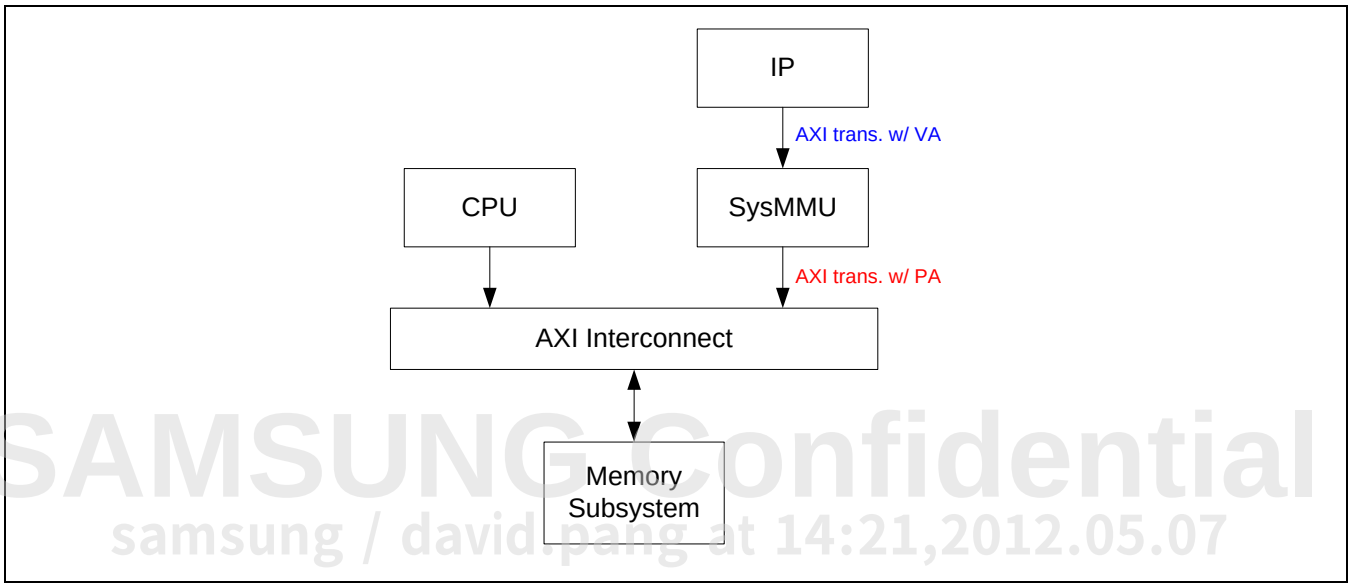


Figure 16-2 An Example of System Integration

16.1.4 Address Translation

Figure 16-3 illustrates the conceptual block diagram of Address Translation Unit. It has a unified TLB. Simultaneously TLB access is also possible for address translation of AW and AR transactions.

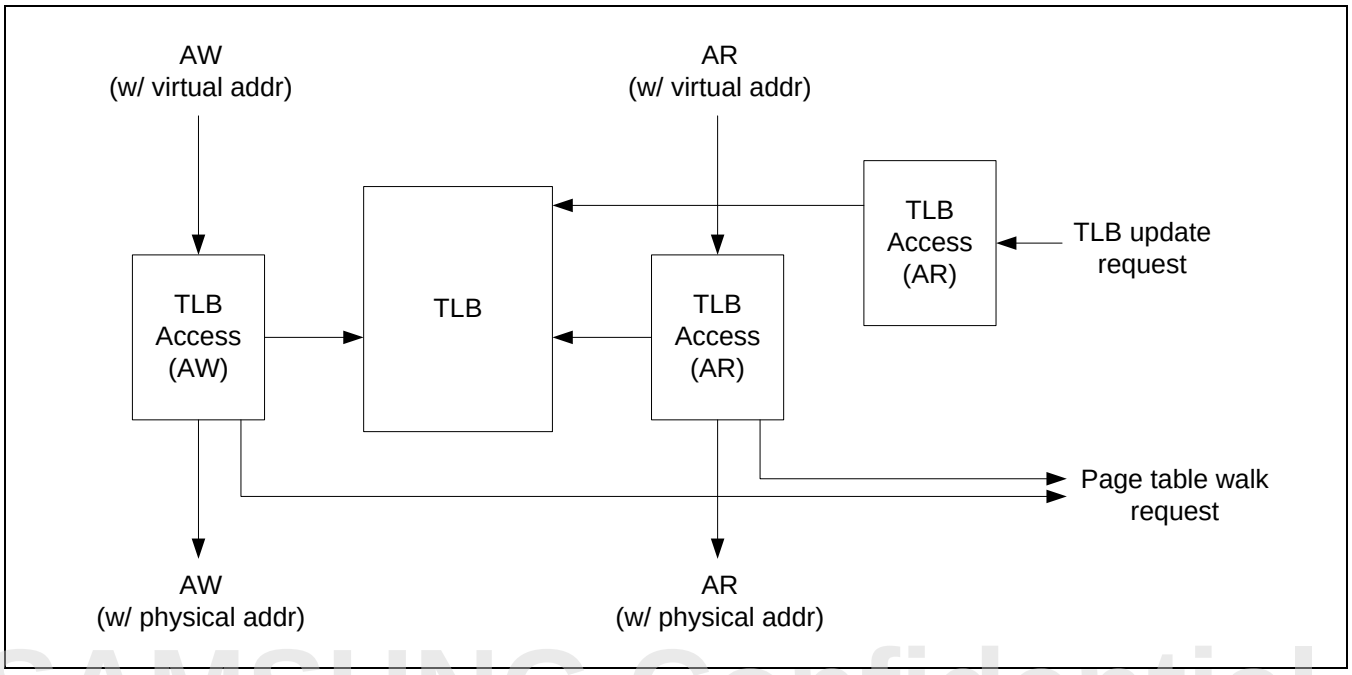


Figure 16-3 Conceptual Block Diagram of Address Translation Unit

Table 16-1 describes the structure of TLB entry.

Table 16-1 The Structure of a TLB Entry

Field	#Bits	Description
VPN	20	Virtual page number
V	1	Valid b0 = Not valid b1 = Valid
PS	2	Page size b01 = 4 KB b00 = 64 KB b10 = 1 MB b11 = 16 MB
PPBA	20	Physical page base address
S	1	Shareable b0 = Non-shareable b1 = Shareable
NS	1	Non-secure b0 = Secure (security fault for non-secure transaction) b1 = Non-secure
AP	3	Access permission When MMU_CFG.AFE is 0, b000 = No-access (access fault) b001, b010, b011 = Read-write b100 = Reserved (access fault) b101, b110, b111 = Read-only When MMU_CFG.AFE is 1 b0xx = Read-write b1xx = Read-only

An entry of TLB contains an address translation information. TLB lookup translates virtual address (VA) to physical address (PA). To look up for address translation information for VA, VA is compared with virtual page number (VPN) of all valid TLB entries according to page size of entries. Comparison bits for each page size are:

- For 4 KB page size, VA[31:12] is compared with VPN[19:0]
- For 64 KB page size, VA[31:16] is compared with VPN[19:4]
- For 1 MB page size, VA[31:20] is compared with VPN[19:8]
- For 16 MB page size, VA[31:24] is compared with VPN[19:12]

And the shareable bit of the TLB entry is assigned to AxUSER_M0 master port.

When a TLB entry matches with VA (i.e. TLB hit), the protection attributes should be checked for page according to SysMMU configuration. It supports two kinds of protections: security protection and access protection. When MMU_CFG.Enable_security_prot is 1, then Address Translation Unit checks non-secure (NS) field of the TLB entry. If security permission is violated, it issues security fault.

When MMU_CFG.Enable_access_prot is 1, then Address Translation Unit checks access permission (AP) field of the TLB entry. If access permission is violated, it issues access fault. On protection fault, it translates VA into DEFAULT_SLAVE_ADDR to complete faulty transaction. When more than one TLB entries match with VA, TLB multi-hit fault is issued.

On TLB hit, the Address Translation Unit translates VA into PA with the physical page base address (PPBA) of TLB entry.

- For 4 KB page, PA[31:0] = {PPBA[19:0], VA[11:0]}
- For 64 KB page, PA[31:0] = {PPBA[19:4], VA[15:0]}
- For 1 MB page, PA[31:0] = {PPBA[19:8], VA[19:0]}
- For 16 MB page, PA[31:0] = {PPBA[19:12], VA[23:0]}

And the shareable bit of the TLB entry is assigned to AxUSER_M0 master port.

When Address Translation Unit fails to find corresponding address translation information in TLB, it requests Page Table Walk Unit to perform page table walking. The Page Table Walk Unit looks up for address translation information in the page table allocated in external memory. It sends TLB update request to the Address Translation Unit after finishing page table walk. Address Translation Unit updates (replaces) a TLB entry with requested address translation information as per the TLB replacement policy.

TLB replacement policy can be either round-robin replacement or LRU replacement. The TLB replacement policy is configured by MMU_CFG register. It should be configured before enabling the SysMMU.

16.1.5 Page Table Walk

In response to a page table walking request from the Address Translation Unit, Page Table Walk Unit reads page table from external memory. It loads missing address translation information into TLB. It requests the AXI master interface to access external memory and read a page table entry. Page table structure has two levels according to page size. There is one first-level page table. Each entry in first-level page table contains first-level page descriptor (FLPD). It describes a second-level page table or 1 MB/16 MB page. Each entry in second-level page table contains second-level page descriptor (SLPD) which describes 4 KB/64 KB page.

Therefore, to read descriptor for 1 MB/16 MB page, the Page Table Walk Unit should access external memory once. For 4 KB/64 KB page, it should access external memory twice. To find the location of FLPD in external memory, the Page Table Walk Unit refers to PT_BASE_ADDR register. The physical address (PA) of FLPD for a virtual address (VA) is:

PA_of_FLPD[31:0] = {PT_BASE_ADDR[31:14-MMU_CFG.PTN], VA[31-MMU_CFG.PTN:20], 2'b0}

When FLPD indicates page fault then Page Table Walk Unit should issue page fault exception. When FLPD describes 1 MB or 16 MB page then Page Table Walk Unit requests Address Translation Unit to update TLB with the base physical address, the page size, and the protection information of FLPD. When FLPD describes second-level page table then PA of SLPD is:

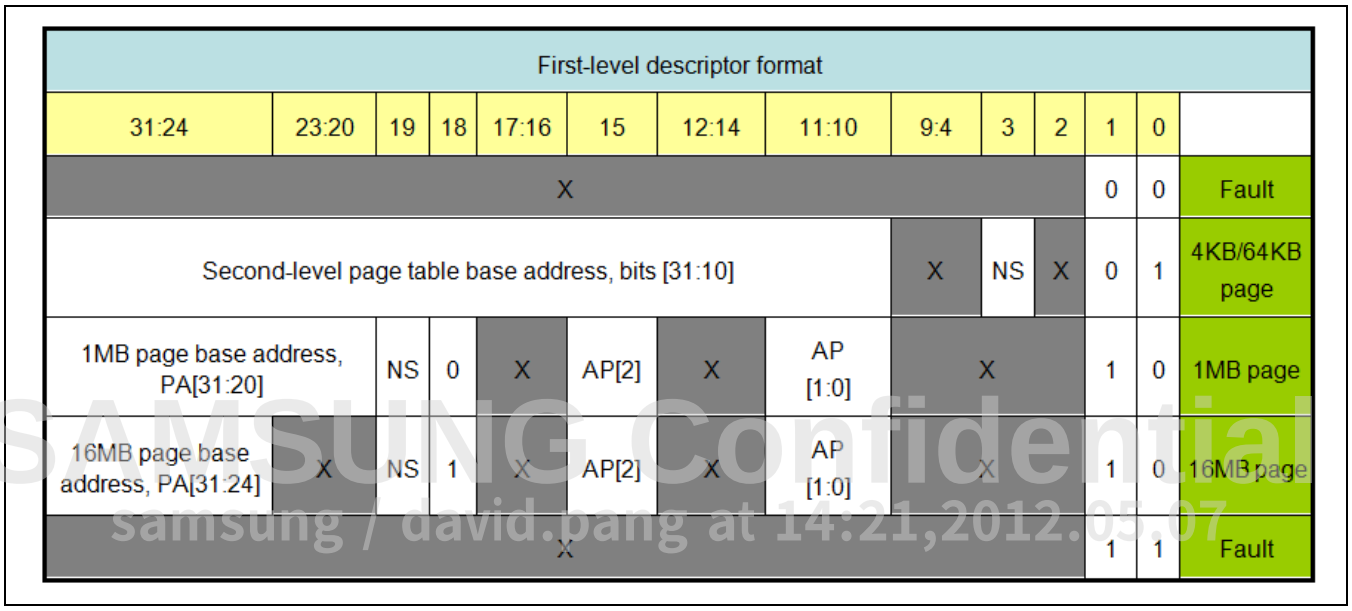
PA_of_SLPD[31:0] = {FLPD[31:10], VA[19:12], 2'b0}

When SLPD indicates page fault then Page Table Walk Unit should issue page fault exception. Otherwise, it requests the Address Translation Unit to update TLB with base physical address, page size, and protection information of the SLPD.

SW should write page table in external memory and set PT_BASE_ADDR register before enabling SysMMU. Since the top four bits of 64 KB page index region of the VA overlap with bottom four bits of second-level table index, the descriptor of 64 KB page should occur first on a sixteen-word boundary and be repeated in 16 consecutive memory locations in a second-level page table. With the same reason, 16 MB page should occur first on a sixteen-word boundary and be repeated in 16 consecutive memory locations in a first-level page table. Refer to ARMv7 architecture reference manual for more detailed explanation.

When SysMMU performs page table walking, the value of ARQOS_M0 master port signal is set with the value of MMU_CFG.PTW_QOS, and value of ARUSER_M0 master port signal is set with the value of MMU_CFG.PT_sharable.

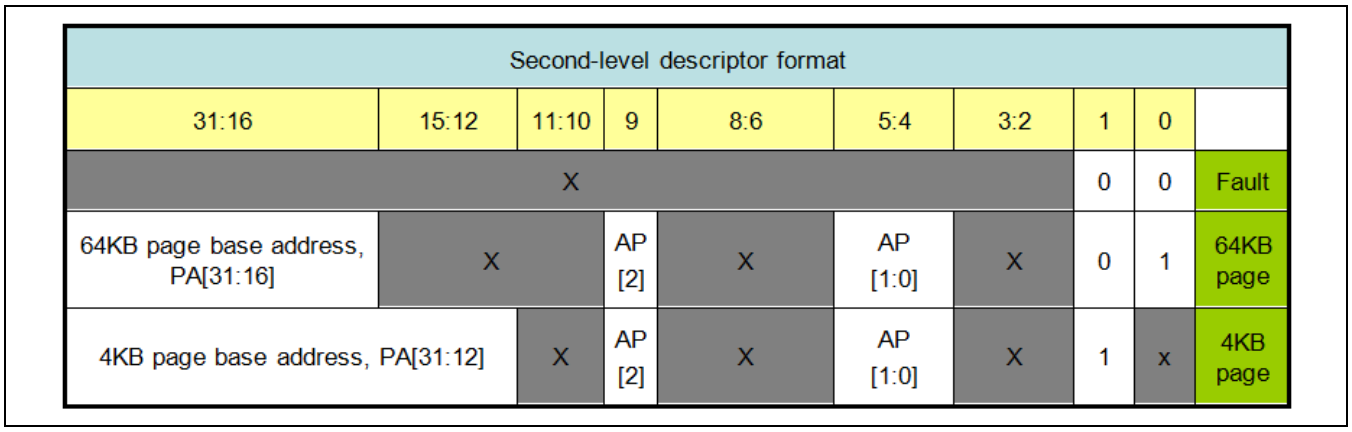
Figure 16-4 illustrates the first-level page descriptor format.



First-level descriptor format													
31:24	23:20	19	18	17:16	15	12:14	11:10	9:4	3	2	1	0	
X											0	0	Fault
Second-level page table base address, bits [31:10]								X	NS	X	0	1	4KB/64KB page
1MB page base address, PA[31:20]		NS	0	X	AP[2]	X	AP [1:0]	X			1	0	1MB page
16MB page base address, PA[31:24]	X	NS	1	X	AP[2]	X	AP [1:0]	X			1	0	16MB page
X											1	1	Fault

Figure 16-4 First-level Page Descriptor Format

Figure 16-5 illustrates the second-level page descriptor format.



Second-level descriptor format								
31:16	15:12	11:10	9	8:6	5:4	3:2	1	0	
X							0	0	Fault
64KB page base address, PA[31:16]	X		AP [2]	X	AP [1:0]	X	0	1	64KB page
4KB page base address, PA[31:12]		X	AP [2]	X	AP [1:0]	X	1	x	4KB page

Figure 16-5 Second-level Page Descriptor Format

16.2 Configuration and Programming View

This section provides information about programming SysMMU and describes its registers used for control and configuration.

16.2.1 Disable Mode

On reset, SysMMU is in disable mode. , in which no address translation is performed and no interrupt is occurred. To enter disable mode from enable mode, set MMU_CTRL.Enable_MMU = 1'b0.

NOTE: It should be confirmed that there is no AXI transaction issued from master IP before SysMMU enters disable mode. It is recommended that the SysMMU mode change between disable and enable mode be performed only during system booting time before the master IP activates.

16.2.2 Enable Mode

To enable SysMMU, set MMU_CTRL.Enable_MMU = 1'b1.

NOTE: It should be confirmed that there is no AXI transaction issued from master IP before SysMMU enters enable mode. It is recommended that the SysMMU mode change between disable and enable mode be performed only when system booting time before the master IP activates.

16.2.3 Block Mode

Block mode is sub mode of enable mode. In block mode, SysMMU blocks AW/AR request. To enter block mode, set MMU_CTRL.Block_MMU = 1'b1. To come back from block mode, set MMU_CTRL.Block_MMU = 1'b0.

Since the SysMMU does not go to be blocked immediately when there are pending AW/AR transactions. MMU_STATUS.Blocked bit should be checked to see if the SysMMU actually goes to be blocked. In some cases, such as page table walking, the SysMMU goes to be blocked (MMU_STATUS.Blocked = 1'b1) automatically.

16.2.4 TLB Replacement Policy

TLB replacement policy can be either round-robin replacement or LRU replacement. The TLB replacement policy is configured by TLB_replace bit of MMU_CFG register. It should be configured before enabling the SysMMU.

16.2.5 TLB Flush

To flush all TLB entries, set MMU_FLUSH.Flush_all_entry = 1'b1. To flush only TLB entries that match virtual address, use MMU_FLUSH_ENTRY register. TLB flush should be performed in block mode.

16.2.6 Fault Handling

The SysMMU checks access protection fault, security protection fault, TLB multi-hit fault, page fault and bus error. If the interrupt is enabled (MMU_CTRL.Enable_INT = 1'b1), it sends interrupt on faults that is not masked by INT_MASK register. INT_STATUS register shows current fault status. To clear interrupt, use INT_CLEAR register. Clearing interrupt command should be issued after SysMMU goes to be blocked (MMU_STATUS.Blocked = 1'b1). Clearing interrupt command does not work until SysMMU is in blocked state.

AW_FAULT_ADDR (AR_FAULT_ADDR) register holds AW (AR) virtual address causing last access protection fault, security protection fault, or TLB multi-hit fault. PAGE_FAULT_ADDR register holds virtual address causing last page fault or bus error.

16.2.6.1 Access Protection Fault

If MMU_CFG.Enable_access_prot bit is configured to be 1'b1, then SysMMU checks access protection fault according to MMU_CFG.AFE bit as described in *Table 16-1*. Access protection fault is issued by the Address Translation Unit when the access permission for the corresponding page is violated.

On access protection fault, SysMMU forwards faulty AXI AW/AR transaction to the address defined in DEFAUT_SLAVE_ADDR register and goes to block state.

16.2.6.2 Security Protection Fault

If MMU_CFG.Enable_security_prot is configured to be 1'b1, then SysMMU checks security protection fault. Security protection fault is issued by the Address Translation Unit when a non-secure AXI transaction accesses a secure page.

On security protection fault, SysMMU forwards faulty AXI AW/AR transaction to address defined in DEFAUT_SLAVE_ADDR register and goes to block state.

16.2.6.3 TLB Multi-hit Fault

TLB multi-hit fault is issued when more than one TLB entries are hit for an AXI AW/AR transaction.

On TLB multi-hit fault, SysMMU forwards the faulty AXI AW/AR transaction to address defined in DEFAUT_SLAVE_ADDR register and goes to block state.

16.2.6.4 Page Fault

Page fault is issued when the SysMMU reads invalid FLPD or SLPD during page table walk. On page fault, SysMMU goes to blocked state. SysMMU retries the page table walk after the page fault is cleared.

16.2.6.5 Bus Error

Bus error is issued when the SysMMU receives bus error response during page table walk. On bus error, SysMMU goes to blocked state. SysMMU retries page table walk after the bus error is cleared.

16.2.7 Miss Rate Measurement

The SysMMU provides event count values through APB access to measure AW/AR miss rate respectively.
To measure AW (AR) miss rate, count total number of AW (AR) transaction and number of AW (AR) misses.

Table 16-2 describes the overview of PPMU_PPC registers.

Table 16-2 Overview of PPMU_PPC Registers

Control Function		Description
Enable/Disable[counter/adder]		PPC_CNTENS/PPC_CNTENC
Enable/Disable[counter/adder]'s interrupt		PPC_INTENS/PPC_INTENC
Start/Stop PPC	Start_mode == 0	Starts and Stops by Register : PPC_PMNC[0] == 1 → Starst, PPC_PMNC[0] == 0 → Stops
	Start_mode == 1	Starts and Stops by External Trigger Trigger == 1 → Starst (In this case, PPC_PMNC[0] read value becomes 1) Trigger == 0 → Stops (In this case, PPC_PMNC[0] read value becomes 0)

PPC_CNTENC and PPC_CNTENS are a pair of related registers. For both, a write of "1" results in accept and a write of "0" will result in ignore. When you want to enable one counter, you should write 1 to its corresponding bit of PPC_CNTENS. After this writing, values you read from PPC_CNTENC and PPC_CNTENS are both changed, which shows corresponding bit of enabled counter as 1. When you want to disable one counter, you should write 1 to its corresponding bit of PPC_CNTENC. Even after this writing, the values you read from PPC_CNTENC and PPC_CNTENS are both changed, which shows corresponding bit of disabled counter as 0. Their initial values are 0x0000_0000.

PPC_INTENS and PPC_INTENC are used for enable and disable interrupt generation of counters. Their setting rules are same as above two registers.

PPC_CCNT is a R/W register. Before you start counting, set the initial value of PPC_CCNT by writing some value to it. When you read after this write, the read value should be same as your written value. After you start counting (suppose that PPC_CCNT counter is enabled), PPC_CCNT increases by one with every cycle from its initial value until you stop counting. Any value you read or write from/to counter registers during counting is ignored.

For an example, the steps to measure miss rate are:

- Enable events: set MMU_CTRL.Enable_PPC_event = 0x1
- Enable PPC interrupt: set PPC_INTENS = 0x8000_000F
- Enable PPC counters: set PPC_CNTENS = 0x8000_000F
- Clear PPC overflow flags: set PPC_FLAG = 0x8000_000F
- Reset all PPC counters: set PPC_PMNC = 0x0000_0006
- Setup sampling duration cycle time: for example, measure miss rate during 10,000 (0x2710) cycles, set PPC_CCNT = 0xFFFF_D8EF (0xFFFF_FFFF – 0x2710)
- Start all PPC counters: set PPC_PMNC = 0x0000_0001
- PPC generates interrupt due to overflow on PPC_CCNT
- Stop all PPC counters: set PPC_PMNC = 0x0000_0000
- Read PPC_FLAG to check whether PPC cycle counter overflow generates interrupt.
- Read PPC counter values:
  - PPC_PMCNT0 = total number of AW transactions
  - PPC_PMCNT1 = total number of AR transactions
  - PPC_PMCNT2 = number of AW transactions causing TLB miss
  - PPC_PMCNT3 = number of AR transactions causing TLB miss
- AW/AR miss rates are calculated by
  - AW miss rate = (value of PPC_PMCNT2)/(value of PPC_PMCNT0)
  - AR miss rate = (value of PPC_PMCNT3)/(value of PPC_PMCNT1)

16.2.8 Register Descriptions

The SysMMU has registers that are used to configure, control or provide its operations and status of certain features.

16.2.9 Register Map Summary

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)

Register	Offset	Description	Reset Value
MMU_CTRL	0x0000	SysMMU control register	0x0000_0000
MMU_CFG	0x0004	SysMMU configuration register	0x0000_0000
MMU_STATUS	0x0008	SysMMU status register	0x0000_0000
MMU_FLUSH	0x000C	Flush all TLB entry	0x0000_0000
MMU_FLUSH_ENTRY	0x0010	Flush an entry by virtual address	0x0000_0000
PT_BASE_ADDR	0x0014	Page table base address	0x0000_0000
INT_STATUS	0x0018	Interrupt status	0x0000_0000
INT_CLEAR	0x001C	Clears interrupt	0x0000_0000
INT_MASK	0x0020	Masks interrupt	0x0000_0000
PAGE_FAULT_ADDR	0x0024	Virtual address that causes last page fault or bus error	0x0000_0000
AW_FAULT_ADDR	0x0028	Virtual address that causes last fault during AW address translation	0x0000_0000
AR_FAULT_ADDR	0x002C	Virtual address that causes last fault during AR	0x0000_0000

Register	Offset	Description	Reset Value
		address translation	
DEFAUT_SLAVE_ADDR	0x0030	Physical address of default slave	0x0000_0000
MMU_VERSION	0x0034	SysMMU version information	0x1020_0020 (NOTE)
PPC_PMNC	0x0800	Performance Monitor Control Register	0x0000_0000
PPC_CNTENS	0x0810	Count Enable Set Register	0x0000_0000
PPC_CNTENC	0x0820	Count Enable Clear Register	0x0000_0000
PPC_INTENS	0x0830	Interrupt Enable Set Register	0x0000_0000
PPC_INTENC	0x0840	Interrupt Enable Clear Register	0x0000_0000
PPC_FLAG	0x0850	Overflow Flag Status Register	0x0000_0000
PPC_CCNT	0x0900	Cycle Count Register	0x0000_0000
PPC_PMCNT0	0x0910	Performance Monitor Count0 Registers (total number of AW transactions)	0x0000_0000
PPC_PMCNT1	0x0920	Performance Monitor Count1 Registers (total number of AR transactions)	0x0000_0000
PPC_PMCNT2	0x0930	Performance Monitor Count2 Registers (number of AW transactions causing TLB miss)	0x0000_0000
PPC_PMCNT3	0x0940	Performance Monitor Count3 Registers (number of AR transactions causing TLB miss)	0x0000_0000

NOTE: The value depends on implementation.

16.2.9.1 MMU_CTRL

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	RW	Reserved (Should be zero, read as zero)	28'b0
Enable_PPMU_EVENT	[3]	RW	0 = Disable PPMU event 1 = Enables PPMU event	1'b0
Enable_INT	[2]	RW	0 = Disables interrupt 1 = Enables interrupt	1'b0
Block_MMU	[1]	RW	Effective only when MMU is enabled 0 = Un-blocks MMU 1 = Blocks MMU	1'b0
Enable_MMU	[0]	RW	0 = Disables MMU (bypass mode) 1 = Enables MMU	1'b0

16.2.9.2 MMU_CFG

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	RW	Reserved. Should be zero, read as zero	25'b0
PT_sharable	[12]	RW	0 = Page table is in non-sharable memory region 1 = Page table is in sharable memory region	1'b0
RSVD	[11]	RW	Reserved (Should be zero, read as zero)	1'b0
PTW_QOS	[10:7]	RW	PL301 r2p0 ARQOS value for PTW	4'b0
PTN	[6:4]	RW	Controls size of First Level Page Table. The virtual address range also changes according to size. b000 = 16 KB (0x0000_0000 to 0xffff_ffff) b001 = 8 KB (0x0000_0000 to 0x7fff_ffff) b010 = 4 KB (0x0000_0000 to 0x3fff_ffff) b011 = 2 KB (0x0000_0000 to 0x1fff_ffff) b100 = 1 KB (0x0000_0000 to 0x0fff_ffff) b101 = 512 B (0x0000_0000 to 0x07ff_ffff) b110 = 256 B (0x0000_0000 to 0x03ff_ffff) b111 = 128 B (0x0000_0000 to 0x01ff_ffff)	3'b0
Enable_security_prot	[3]	RW	0 = Disables security protection check 1 = Enables security protection check	1'b0
Enable_access_prot	[2]	RW	0 = Disables access protection check 1 = Enables access protection check	1'b0

Name	Bit	Type	Description	Reset Value
AFE	[1]	RW	0 = Checks access protection with AP[2:1] of page table entry 1 = Checks access protection with AP[2:0] of page table entry	1'b0
TLB_replace	[0]	RW	0 = Round-robin replacement 1 = LRU replacement	1'b0

16.2.9.3 MMU_STATUS

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	R	Reserved (Read as zero)	28’b0
AR_pend	[3]	R	1 = AR pending state	1'b0
AW_pend	[2]	R	1 = AW pending state	1'b0
PTW	[1]	R	1 = PTW state	1'b0
Blocked	[0]	R	1 = Blocks MMU	1'b0

16.2.9.4 MMU_FLUSH

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	W	Reserved (Should be zero)	30'b0
Flush_all_entry	[0]	W	1 = Flush all TLB entry	1'b0

16.2.9.5 MMU_FLUSH_ENTRY

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
VPN	[31:12]	W	Virtual page number	20'b0
RSVD	[11:1]	W	Reserved (Should be zero)	11'b0
Flush_entry	[0]	W	1 = Flush a entry by VPN	1'b0

16.2.9.6 PT_BASE_ADDR

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PT_base_addr	[31:7]	RW	Page table base address, bit[31:7]	25'b0
RSVD	[6:0]	RW	Reserved (Should be zero, read as zero)	7'b0

16.2.9.7 INT_STATUS

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	R	Reserved (Read as zero)	24'b0
AW_Access_fault	[7]	R	AW Access protection fault	1'b0
AW_Security_fault	[6]	R	AW Security protection fault	1'b0
AR_Access_fault	[5]	R	AR Access protection fault	1'b0
AR_Security_fault	[4]	R	AR Security protection fault	1'b0
Bus_error	[3]	R	Bus error	1'b0
AW_MH_fault	[2]	R	AW multi-hit fault	1'b0
AR_MH_fault	[1]	R	AR multi-hit fault	1'b0
Page_fault	[0]	R	Page fault	1'b0

16.2.9.8 INT_CLEAR

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	W	Reserved	24'b0
AW_access_fault	[7]	W	1 = Clears AW access protection fault	1'b0
AW_security_fault	[6]	W	1 = Clears AW security protection fault	1'b0
AR_access_fault	[5]	W	1 = Clears AR access protection fault	1'b0
AR_security_fault	[4]	W	1 = Clears AR security protection fault	1'b0
Bus_error	[3]	W	1 = Clears bus error	1'b0
AW_MH_fault	[2]	W	1 = Clears AW multi-hit fault	1'b0
AR_MH_fault	[1]	W	1 = Clears AR multi-hit fault	1'b0
Page_fault	[0]	W	1 = Clears page fault	1'b0

16.2.9.9 INT_MASK

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	RW	Reserved (Should be zero, read as zero)	24'b0
AW_Access_fault	[7]	RW	1 = Masks AW access protection fault	1'b0
AW_Security_fault	[6]	RW	1 = Masks AW security protection fault	1'b0
AR_Access_fault	[5]	RW	1 = Masks AR access protection fault	1'b0
AR_Security_fault	[4]	RW	1 = Masks AR security protection fault	1'b0
Bus_error	[3]	RW	1 = Masks bus error	1'b0
AW_MH_fault	[2]	RW	1 = Masks AW multi-hit fault	1'b0
AR_MH_fault	[1]	RW	1 = Masks AR multi-hit fault	1'b0
Page_fault	[0]	RW	1 = Masks page fault	1'b0

16.2.9.10 PAGE_FAULT_ADDR

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Page_Fault_addr	[31:0]	R	Virtual address that causes last page fault or bus error	32'b0

16.2.9.11 AW_FAULT_ADDR

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
AW_Fault_addr	[31:0]	R	Virtual address that causes last AW multi-hit/access/security fault	32'b0

16.2.9.12 AR_FAULT_ADDR

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
AR_Fault_addr	[31:0]	RO	Virtual address that causes last AR multi-hit/access/security fault	32'b0

16.2.9.13 DEFAUT_SLAVE_ADDR

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Default_slave_addr	[31:4]	RW	Physical address of default slave[31:4]	28'b0
RSVD	[3:0]	RW	Reserved (Should be zero, read as zero)	4'b0

16.2.9.14 MMU_VERSION

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0034, Reset Value = 0x1020_0020 (NOTE)

Name	Bit	Type	Description	Reset Value
Major_arch_version	[31:28]	R	Major architecture version	4'b1
Minor_arch_version	[27:21]	R	Minor architecture version	7'b1
RTL_version	[20:7]	R	RTL revision version	14'b0 (NOTE)
TLB_size	[6:0]	R	Number_of_TLB_entry	7'd32 (NOTE)

NOTE: Value is implementation dependant.

16.2.9.15 PPC_PMNC

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0800, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	RW	Reserved	12'b0
RSVD	[19:17]	RW	Reserved	3'b0
START MODE	[16]	RW	PPMU Start Mode 0 = S/W (by CPU) 1 = H/W (by SYSCON)	1'b0
RSVD	[15:5]	RW	Reserved	11'b0
RSVD	[4]	RW	Reserved	1'b0
CC DIVIDER	[3]	RW	Cycle count divider 0 = Counts every processor clock cycle, resets value 1 = Counts every 64th processor clock cycle	1'b0
CC RESET	[2]	W	Cycle counter reset 0 = No action 1 = Resets cycle counter (CCNT) to zero	1'b0
PPMU COUNTER RESET	[1]	W	Performance counter reset 0 = No action 1 = Resets all performance counters to zero	1'b0
PPMU ENABLE	[0]	RW	Enables bit 0 = Disables all counters, including CCNT	1'b0

Name	Bit	Type	Description	Reset Value
			1 = Enables all counters including CCNT When you read it, "1" means it is counting and "0" means it is idle (stop counting). You can write it only when start mode is set to be "0" (PPMU is started by CPU). At this time, you can write this bit by "1" to start counting and write it by "0" to stop counting. When the start mode is set to "1" (PPMU is started by SYSCON), you only can read it and get the status of the PPMU. At this time, PPMU is controlled by external trigger. When external trigger is "1", counting starts, and when external trigger is "0", counting stops.	

16.2.9.16 PPC_CNTENS

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0810, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	Enables Cycle Counter	1'b0
RSVD	[30:4]	RW	Reserved	27'b0
PMCNT3	[3]	RW	Enables Counter 3	1'b0
PMCNT2	[2]	RW	Enables Counter 2	1'b0
PMCNT1	[1]	RW	Enables Counter 1	1'b0
PMCNT0	[0]	RW	Enables Counter 0	1'b0

16.2.9.17 PPC_CNTENC

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0820, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	Disables cycle counter	1'b0
RSVD	[30:4]	RW	Reserved	27'b0
PMCNT3	[3]	RW	Disables Counter 3	1'b0
PMCNT2	[2]	RW	Disables Counter 2	1'b0
PMCNT1	[1]	RW	Disables Counter 1	1'b0
PMCNT0	[0]	RW	Disables Counter 0	1'b0

16.2.9.18 PPC_INTENS

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0830, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	Enables CCNT overflow interrupt	1'b0
RSVD	[30:4]	RW	Reserved	27'b0
PMCNT3	[3]	RW	Enables Counter 3 overflow interrupt	1'b0
PMCNT2	[2]	RW	Enables Counter 2 overflow interrupt	1'b0
PMCNT1	[1]	RW	Enables Counter 1 overflow interrupt	1'b0
PMCNT0	[0]	RW	Enables Counter 0 overflow interrupt	1'b0

16.2.9.19 PPC_INTENC

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0840, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	Disables CCNT overflow interrupt	1'b0
RSVD	[30:4]	RW	Reserved	27'b0
PMCNT3	[3]	RW	Disables Counter 3 overflow interrupt	1'b0
PMCNT2	[2]	RW	Disables Counter 2 overflow interrupt	1'b0
PMCNT1	[1]	RW	Disables Counter 1 overflow interrupt	1'b0
PMCNT0	[0]	RW	Disables Counter 0 overflow interrupt	1'b0

16.2.9.20 PPC_FLAG

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0850, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	Cycle counter overflow flag	1'b0
RSVD	[30:4]	RW	Reserved	27'b0
PMCNT3	[3]	RW	Counter 3 overflow flag	1'b0
PMCNT2	[2]	RW	Counter 2 overflow flag	1'b0
PMCNT1	[1]	RW	Counter 1 overflow flag	1'b0
PMCNT0	[0]	RW	Counter 0 overflow flag	1'b0

16.2.9.21 PPC_CCNT

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0900, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31:0]	RW	CCNT Register contains an event count	32'b0

16.2.9.22 PPC_PMCNT0

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0910, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PMCNT	[31:0]	RW	PMCNT Register contains an event count (total number of AW transactions)	32'b0

16.2.9.23 PPC_PMCNT1

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0920, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PMCNT	[31:0]	RW	PMCNT Register contains an event count (total number of AR transactions)	32'b0

16.2.9.24 PPC_PMCNT2

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0930, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PMCNT	[31:0]	RW	PMCNT Register contains an event count (number of AW transactions that causes TLB miss)	32'b0

16.2.9.25 PPC_PMCNT3

- Base Address: 0x10A4_0000 (SMMUG2D_ACP)
- Base Address: 0x10A5_0000 (SMMUSSS)
- Base Address: 0x11A2_0000 (SMMUFIMC0)
- Base Address: 0x11A3_0000 (SMMUFIMC1)
- Base Address: 0x11A4_0000 (SMMUFIMC2)
- Base Address: 0x11A5_0000 (SMMUFIMC3)
- Base Address: 0x11A6_0000 (SMMUJPEG)
- Base Address: 0x11E2_0000 (SMMUFIMD0)
- Base Address: 0x1226_0000 (sysMMU_FIMC-ISP)
- Base Address: 0x1227_0000 (sysMMU_FIMC-DRC)
- Base Address: 0x122A_0000 (sysMMU_FIMC-FD)
- Base Address: 0x122B_0000 (sysMMU_ISPCPU)
- Base Address: 0x123B_0000 (sysMMU_FIMC-LITE0)
- Base Address: 0x123C_0000 (sysMMU_FIMC-LITE1)
- Base Address: 0x1273_0000 (SMMUGPS)
- Base Address: 0x12A3_0000 (SMMURotator)
- Base Address: 0x12A4_0000 (SMMUMDMA)
- Base Address: 0x12E2_0000 (SMMUTV)
- Address = Base Address + 0x0940, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PMCNT	[31:0]	RW	PMCNT Register contains an event count (number of AR transactions that causes TLB miss)	32'b0

17 System Memory Management Unit

17.1 Overview

System Memory Management Unit (SysMMU) version 2.0 is an Advanced Microcontroller Bus Architecture (AMBATM) AXI compliant address translation unit. It is responsible for handling accesses to memory subsystem requested by master IPs in a SoC. It normally translates virtual page numbers to physical page numbers through an associative cache, called a Translation Lookaside Buffer (L1-TLB). It also concatenates the physical page number into the page offset to obtain a complete physical address. In comparison to the SysMMU version 1.0, the SysMMU version 2.0 has additional secondary TLB (L2-TLB) to reduce the L1-TLB miss penalty. The SysMMU indicates the SysMMU version 2.0 in this document.

17.1.1 Features

The features of SysMMU are:

- A slave port compatible with AMBA3 AXI specification for address translation
- A master port compatible with AMBA3 AXI specification
- A slave port compatible with AMBA3 APB specification for programming registers
- Translates 32-bit virtual addresses to 32-bit physical addresses
- Page size: 4 KB, 64 KB, 1 MB, and 16 MB
- Supports identical page table entry format with ARMv7
- Supports page protection: Read-only/Read-Write, and security
- Provides Round-Robin and Least Recently Used (LRU) policies for L1-TLB entry replacement
- Provides 512-entry, 8-way set associative secondary TLB (L2-TLB) to reduce L1-TLB miss penalty
- Interrupt: page fault, Read/Write protection fault, L1-TLB multi-hit fault, and Bus error
- Supports DISABLE mode to bypass AXI channels
- Zero-cycle latency on L1-TLB hit

17.2 Functional Description

This section describes the functional overview and key functions of the SysMMU.

The key functions of SysMMU are:

- Address translation
- Page table walk

17.2.1 Functional Overview

The AXI Slave Interface block accepts AXI requests from master IPs. It performs L1-TLB lookup operations in the address translation block on AW/AR addresses passing through the AXI Slave Interface. In case of L1-TLB hit, the address translation block translates virtual AW/AR address to physical AW/AR l address.

The AXI Master Interface block issues AXI requests equipped with the translated AW/AR physical address to memory subsystem. In case of L1-TLB miss, the L2-TLB access block searches for the corresponding address translation information in the L2-TLB. When the L2-TLB access block finds the information, it updates the L1-TLB with the address translation information. Otherwise, the Page Table Walk block performs page table walking to find the information from the page table.

The Page Table Walk block reads page tables from system working memory through the AXI Master Interface block and finally refills both L1-TLB and L2-TLB. The Operating System (OS) or middleware manages the page tables that should be written to and available in working memory along with the initialization of the SysMMU.

Figure 17-1 illustrates the conceptual block diagram of the SysMMU.

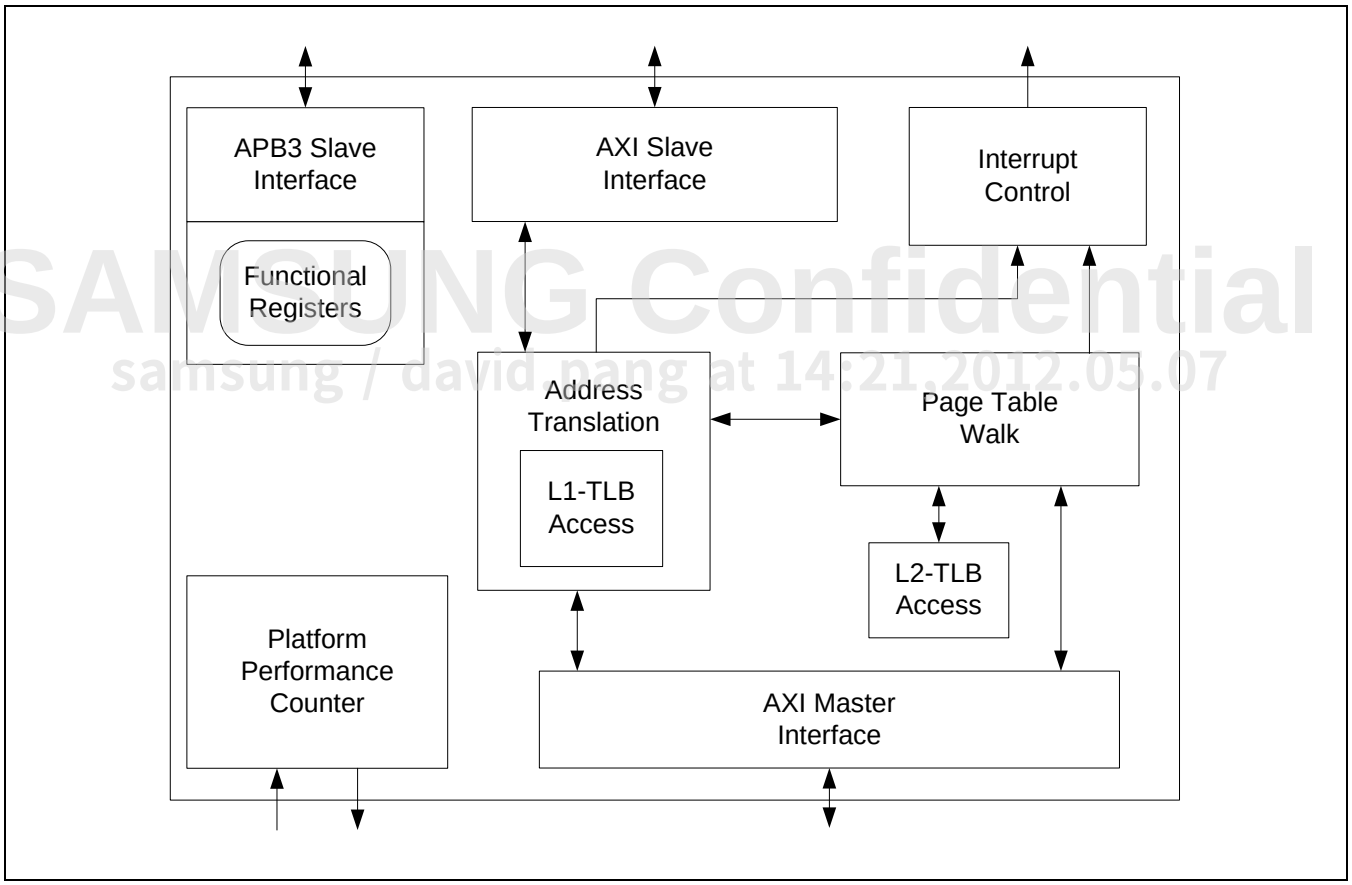


Figure 17-1 SysMMU High-level Block Diagram

The SysMMU supports four page sizes. They are, 4 KB, 64 KB, 1 MB, and 16 MB. ARMv7 architecture defines the page sizes.

Figure 16-2 illustrates an example of system integration with the SysMMU.

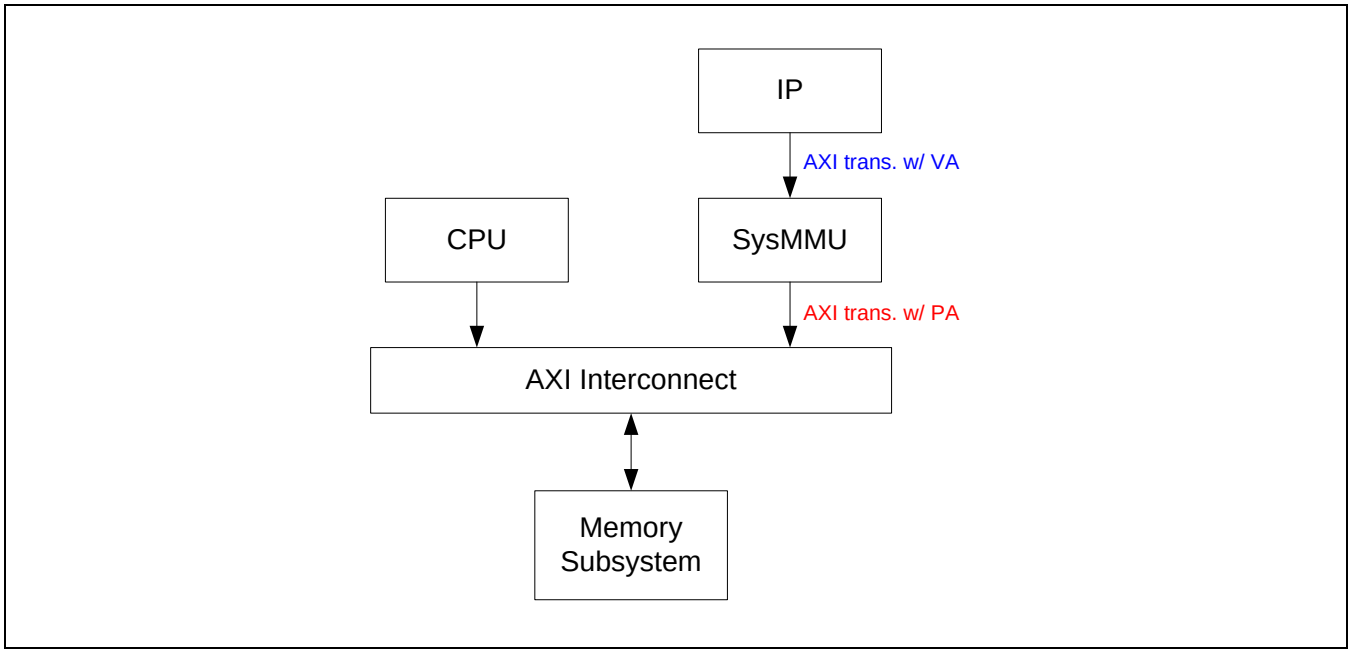


Figure 17-2 An Example of System Integration

17.2.2 Address Translation

Address Translation Unit has a unified L1-TLB, so that simultaneous L1-TLB accesses are possible for address translation of AW and AR transactions.

Figure 17-3 illustrates conceptual block diagram of the Address Translation Unit.

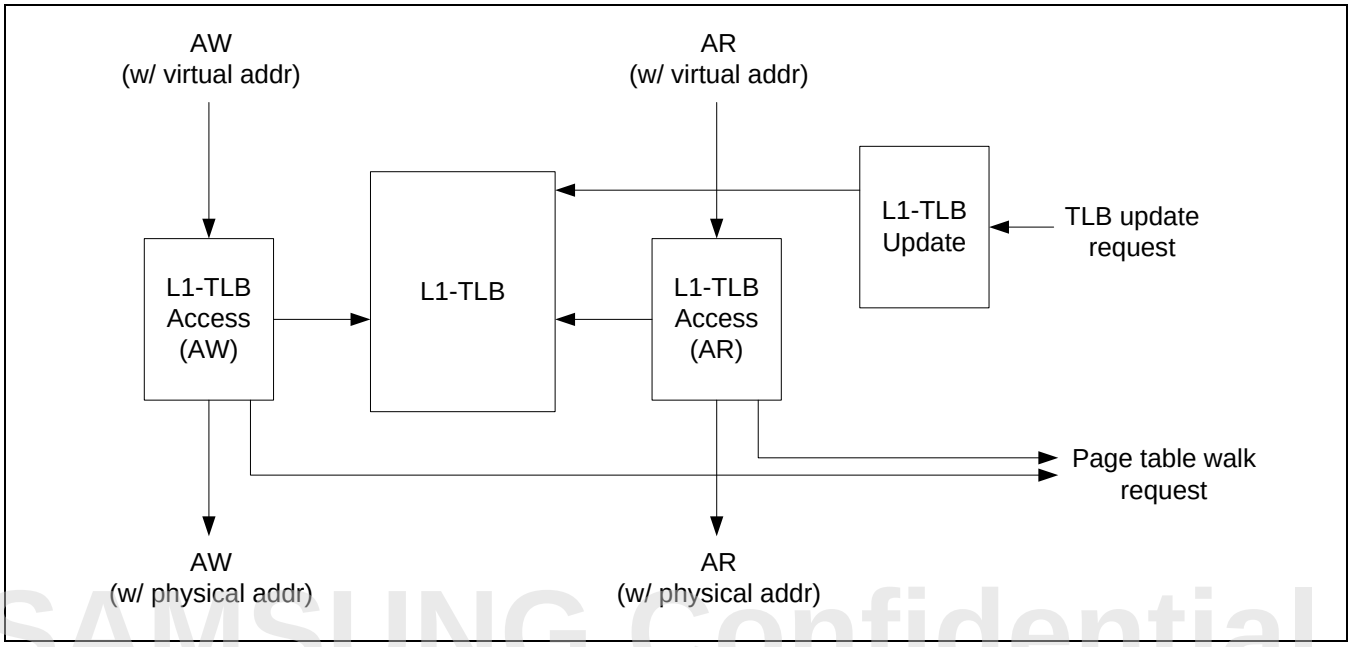


Figure 17-3 Conceptual Block Diagram of Address Translation Unit

Table 16-1 describes the structure of a Translation Lookaside Buffer (TLB) entry.

Table 17-1 Structure of a TLB Entry

Field	#Bit	Description
VPN	20	Virtual Page Number
V	1	Valid b0 = Not valid b1 = Valid
PS	2	Page Size b01 = 4 KB b00 = 64 KB b10 = 1 MB b11 = 16 MB
PPBA	20	Physical Page Base Address
NS	1	Non-secure b0 = Secure (security fault for non-secure transaction) b1 = Non-secure
AP	3	Access Permission • When MMU_CFG.AFE is 0, b000 = No-access (access fault) b001, b010, b011 = Read-write b100 = Reserved (access fault) b101, b110, b111 = Read-only • When MMU_CFG.AFE is 1, b0xx = Read-write b1xx = Read-only

An entry of L1-TLB contains address translation information. Table 16-1 describes the structure of an L1-TLB entry. The L1-TLB lookup translates Virtual Address (VA) into physical address (PA). To look up address translation information for VA, it compares the VA with Virtual Page Number (VPN) of all valid L1-TLB entries according to the Page Size (PS) field of the entries.

The compared bits for each page size are:

- For 4 KB PS, it compares VA[31:12] with VPN[19:0]
- For 64 KB PS, it compares VA[31:16] with VPN[19:4]
- For 1 MB PS, it compares VA[31:20] with VPN[19:8]
- For 16 MB PS, it compares VA[31:24] with VPN[19:12]

When an L1-TLB entry matches with VA (that is, L1-TLB hit), it should verify the protection attributes for the page according to the SysMMU configuration.

L1-TLB entry supports two kinds of protections. They are:

- Security protection
- Access protection

When MMU_CFG.Enable_security_prot is 1, the Address Translation Unit verifies the Non-secure (NS) field of the L1-TLB entry. If it violates the security permission, then it issues security fault.

When MMU_CFG.Enable_access_prot is 1, the Address Translation Unit verifies the Access Permission (AP) field of the TLB entry. If it violates the AP field, then it issues access fault. On the protection fault, it translates the VA into DEFAULT_SLAVE_ADDR to complete faulty transaction. When more than one L1-TLB entries matches with the VA, then it issues L1-TLB multi-hit fault.

On L1-TLB hit, the Address Translation Unit translates the VA into PA with the physical page base address (PPBA) of the L1-TLB entry:

- For 4 KB page, PA[31:0] = {PPBA[19:0], VA[11:0]}
- For 64 KB page, PA[31:0] = {PPBA[19:4], VA[15:0]}
- For 1 MB page, PA[31:0] = {PPBA[19:8], VA[19:0]}
- For 16 MB page, PA[31:0] = {PPBA[19:12], VA[23:0]}

When the Address Translation Unit fails to find corresponding address translation information in L1-TLB, it requests the Page Table Walk Unit to perform page table walking. The Page Table Walk Unit, before page table walking, requests the L2-TLB Access Unit to look up L2-TLB for the address translation information. If the L2-TLB Access Unit fails to find the information, then the Page Table Walk Unit looks up address translation information in the page table allocated in external memory by performing page table walking. It sends L1-TLB update request to the Address Translation Unit after completing L2-TLB look-up or page table walk. The Address Translation Unit updates or replacesan L1-TLB entry with requested address translation information according to the L1-TLB replacement policy.

The L1-TLB replacement policy can be either round-robin replacement or LRU replacement. MMU_CFG register configures the L1-TLB replacement policy. It should be configured before enabling the SysMMU.

17.2.3 Page Table Walk

In response to a page table walking request from the Address Translation Unit, the Page Table Walk Unit, firstly, requests the L2-TLB Access Unit to lookup L2-TLB. Refer to Section *17.2.4 L2-TLB Access Unit* for details of the L2-TLB Access Unit. In case of L2-TLB miss, the Page Table Walk Unit reads page table from external memory to load the missed address translation information into both L1-TLB and L2-TLB. It requests the AXI master interface to access external memory and read a page table entry. Page table is structured with two levels according to the page size. There is one first-level page table. Each entry in first-level page table contains first-level page descriptor (FLPD) which describes a second-level page table or 1MB/16MB page. Each entry in second-level page table contains second-level page descriptor (SLPD) which describes 4KB/64KB page. Therefore, to read descriptor for 1MB/16MB page, the Page Table Walk Unit should access external memory once, and for 4KB/64KB page, it should access external memory twice. To find the location of FLPD in external memory, the Page Table Walk Unit refers to PT_BASE_ADDR register.

The physical address (PA) of FLPD for a virtual address (VA) is:

- PA_of_FLPD[31:0] = {PT_BASE_ADDR[31:14-MMU_CFG.PTN], VA[31-MMU_CFG.PTN:20], 2'b0}

Figure 16-4 illustrates the format of FLPD. If the FLPD indicates page fault, the Page Table Walk Unit should issue page fault exception. If the FLPD describes 1 MB or 16 MB page, the Page Table Walk Unit requests the Address Translation Unit to update L1-TLB with the base physical address that is, the page size and the protection information of the FLPD.

If the FLPD describes second-level page table, the PA of SLPD is:

- PA_of_SLPD[31:0] = {FLPD[31:10], VA[19:12], 2'b0}

First-Level Descriptor Format

31:24	23:20	19	18	17:16	15	12:14	11:10	9:4	3	2	1	0	
X											0	0	Fault
Second-level page table base address, bits [31:10]								X	NS	X	0	1	4KB/64KB page
1MB page base address, PA[31:20]		NS	0	X	AP[2]	X	AP[1:0]	X			1	0	1MB page
16MB page base address, PA[31:24]	X	NS	1	X	AP[2]	X	AP[1:0]	X			1	0	16MB page
X											1	1	Fault

Figure 17-4 First-Level Page Descriptor Format

Figure 16-5 illustrates the format of SLPD. If the SLPD indicates page fault, the Page Table Walk Unit should issue page fault exception. Otherwise, it requests the Address Translation Unit to update L1-TLB with the base physical address that is, the page size and the protection information of the SLPD.

Software should write page table in external memory and set PT_BASE_ADDR register before enabling the SysMMU. As the top four bits of 64 KB page index region of the VA overlaps with the bottom four bits of the second-level table index, the descriptor of 64 KB page should occur first on a sixteen-word boundary. This should repeat in 16 consecutive memory locations in a second-level page table. Similarly, 16 MB page should occur first on a sixteen-word boundary and should repeat in 16 consecutive memory locations in the first-level page table.

Refer to ARMv7 architecture reference manual for more information.

Second-Level Descriptor Format									
31:16	15:12	11:10	9	8:6	5:4	3:2	1	0	
X							0	0	Fault
64KB page base address, PA[31:16]	X		AP[2]	X	AP[1:0]	X	0	1	64KB page
4KB page base address, PA[31:12]		X	AP[2]	X	AP[1:0]	X	1	x	4KB page

Figure 17-5 Second-Level Page Descriptor Format

17.2.4 L2-TLB Access Unit

Figure 17-6 illustrates the block diagram of L2-TLB.

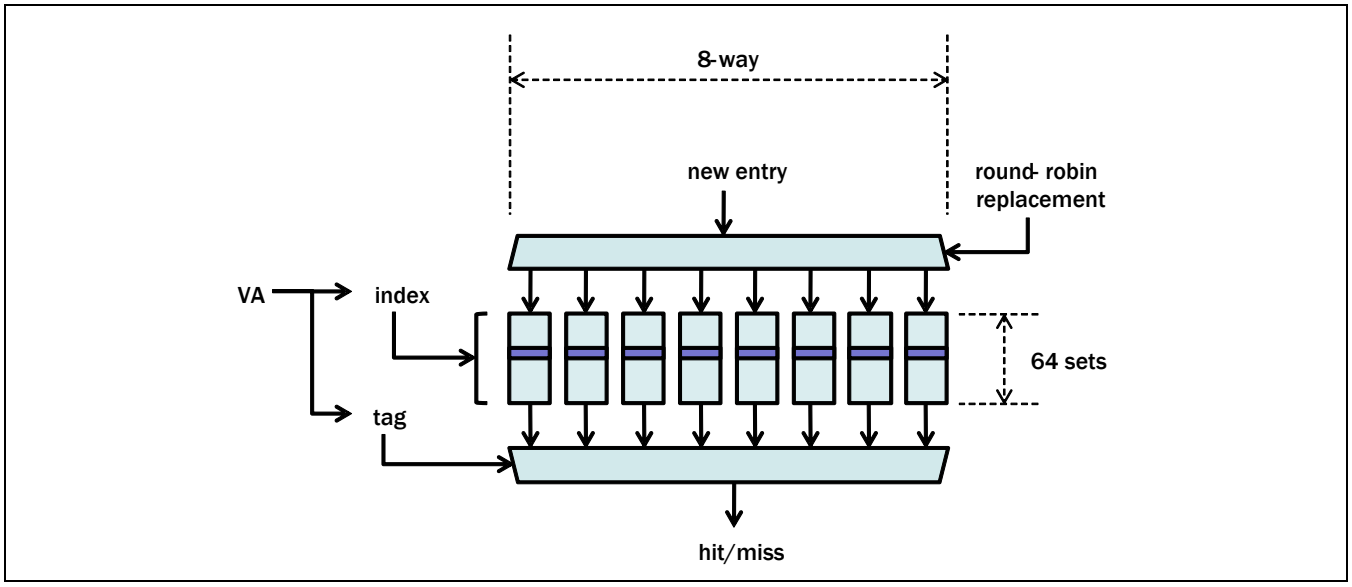


Figure 17-6 L2-TLB Block Diagram

L2-TLB has 8-way set-associative cache structure and has total 512 entries as illustrated in *Figure 17-6*. The index and tag are obtained from VA as following:

- index[5:0] = VA[17:12]
- tag[13:0] = VA[31:18]

The structure of L2-TLB entry is similar to the TLB entry except VPN field. An L2-TLB entry has 14-bit "tag" field instead of VPN field of TLB entry.

Tag-matching is performed according to the page size indicated by the entry as following:

- For 4 KB page size, tag[13:0] is compared with the entry tag[13:0]
- For 64 KB page size, tag[13:0] is compared with the entry tag[13:0]
- For 1 MB page size, tag[13:2] is compared with the entry tag[13:2]
- For 16 MB page size, tag[13:6] is compared with the entry tag[13:6]

On L2-TLB hit, the L2-TLB Access Unit sends the found entry data to the Address Translation Unit to update L1-TLB. Even in case of multi-hit on L2-TLB, the L2-TLB Access Unit selects one entry from the matched entries and sends it to the Address Translation Unit.

The L2-TLB is updated by round-robin replacement scheme.

17.3 Configuration and Programming View

This section provides information about programming the SysMMU and describes its registers used for control and configuration.

17.3.1 Disable Mode

On reset, the SysMMU is in disable mode, in which no address translation is performed and no interrupt is occurred. To enter disable mode from enable mode, set MMU_CTRL.Enable_MMU = 1'b0.

NOTE: It should be confirmed that there is no AXI transaction issued from master IP before the SysMMU enters disable mode. It is recommended that the SysMMU mode change between disable and enable mode be performed only during system booting time before the master IP activates.

17.3.2 Enable Mode

To enable the SysMMU, set MMU_CTRL.Enable_MMU = 1'b1.

NOTE: It should be confirmed that there is no AXI transaction issued from master IP before the SysMMU enters enable mode. It is recommended that the SysMMU mode change between disable and enable mode be performed only when system booting time before the master IP activates.

17.3.3 Block Mode

Block mode is sub mode of enable mode. In block mode, SysMMU blocks AW/AR request. To enter block mode, set MMU_CTRL.Block_MMU = 1'b1. To come back from block mode, set MMU_CTRL.Block_MMU = 1'b0.

As the SysMMU does not go to be blocked immediately when there are pending AW/AR transactions, MMU_STATUS.Blocked bit should be checked to see if the SysMMU actually goes to be blocked. In some cases, such as page table walking, the SysMMU goes to block mode (MMU_STATUS.Blocked = 1'b1) automatically.

Similarly, you should verify MMU_STATUS.Blocked bit to see if the SysMMU actually goes to unblock mode after unblocking (that is, set MMU_CTRL.Block_MMU = 1'b0).

17.3.4 L1-TLB Replacement Policy

The L1-TLB replacement policy can be either round-robin replacement or LRU replacement. The L1-TLB replacement policy is configured by L1TLB_replace bit of MMU_CFG register. It should be configured before enabling the SysMMU.

17.3.5 TLB Flush

To flush all L1/L2-TLB entries, you should set MMU_FLUSH. Flush_all_entry = 1'b1. To flush only the L1/L2-TLB entries that match a virtual address, you should use MMU_FLUSH_ENTRY register. You should perform TLB flush in block mode.

17.3.6 Fault Handling

The SysMMU verifies access protection fault, security protection fault, L1-TLB multi-hit fault, page fault, and bus error. When it enables the interrupt (MMU_CTRL.Enable_INT = 1'b1), it sends interrupt on the faults that is not masked by INT_MASK register. INT_STATUS register shows current fault status. To clear interrupt, use INT_CLEAR register. Clearing interrupt command should be issued after SysMMU goes to be blocked (MMU_STATUS.Blocked = 1'b1). Clearing interrupt command will not work unless the SysMMU is in block mode.

AW_FAULT_ADDR (AR_FAULT_ADDR) register holds the AW (AR) virtual address causing last access protection fault, security protection fault, or L1-TLB multi-hit fault. PAGE_FAULT_ADDR register holds the virtual address causing last page fault or bus error.

17.3.7 Access Protection Fault

When it configures MMU_CFG.Enable_access_prot bit to 1'b1, the SysMMU verifies access protection fault according to MMU_CFG.AFE bit as described in *Table 16-1*. Address Translation Unit issues the access protection fault when it violates the access permission for the corresponding page.

On access protection fault, the SysMMU forwards the faulty AXI AW/AR transaction to the address defined in DEFAUT_SLAVE_ADDR register and goes to block mode.

17.3.8 Security Protection Fault

When it configures MMU_CFG.Enable_security_prot to 1'b1, the SysMMU verifies security protection fault. Address Translation Unit issues security protection fault when a non-secure AXI transaction accesses a secure page.

On security protection fault, the SysMMU forwards the faulty AXI AW/AR transaction to the address defined in DEFAUT_SLAVE_ADDR register and goes to block mode.

17.3.9 L1-TLB Multi-hit Fault

L1-TLB multi-hit fault is issued when more than one L1-TLB entries are hit for an AXI AW/AR transaction.

On L1-TLB multi-hit fault, the SysMMU forwards the faulty AXI AW/AR transaction to the address defined in DEFAUT_SLAVE_ADDR register and goes to block mode.

17.3.10 Page Fault

Page fault is issued when the SysMMU reads invalid FLPD or SLPD during page table walk.

On page fault, the SysMMU goes to block mode. SysMMU retries the page table walk after it clears the page fault is cleared.

17.3.11 Bus Error

Bus error is issued when the SysMMU receives bus error response during page table walk. On bus error, the SysMMU goes to block mode. SysMMU retries the page table walk after it clears the bus error.

17.3.12 Miss Rate Measurement

The SysMMU provides event count values through APB access to measure AR/AW miss rate respectively. It counts total number of AR/AW transaction and number of AR/AW misses to measure AR/AW miss rate.

Table 17-2 describes the overview of PPMU_PPC registers.

Table 17-2 Overview of PPMU_PPC Registers

Control Function		Description
Enable/Disable [counter/adder]		PPC_CNTENS/PPC_CNTENC
Enable/Disable [counter/adder]'s interrupt		PPC_INTENS/PPC_INTENC
Start/Stop PPC	Start_mode == 0	Start and Stop by Register : PPC_PMNC[0] == 1 → Start PPC_PMNC[0] == 0 → Stop
	Start_mode == 1	Start and Stop by External Trigger Trigger == 1 → Start (In this case, PPC_PMNC[0] read value becomes 1) Trigger == 0 → Stop (In this case, PPC_PMNC[0] read value becomes 0)

PPC_CNTENC and PPC_CNTENS is a pair of related registers. Both registers only accept "writing 1" ignoring "writing 0". If you want to enable one counter, you should write "1" to its corresponding bit of PPC_CNTENS. After this writing, it changes the values you read from PPC_CNTENC and PPC_CNTENS that shows the corresponding bit of the enabled counter with a write of "1". If you want to disable one counter, you should write "1" to its corresponding bit of PPC_CNTENC. Even after this writing, it changes the values you read from PPC_CNTENC and PPC_CNTENS that shows the corresponding bit of the disabled counter with a write of "0". Their initial values are 0x0000_0000.

Use PPC_INTENS and PPC_INTENC to enable and disable the interrupt generation of the counters. Their setting rules are similar to PPC_CNTENC and PPC_CNTENS registers.

PPC_CCNT is an R/W register. Before start counting, you can set initial value of PPC_CCNT by writing some value to it. Read after this write, the read value should be similar to your written value. After start counting (suppose it enables the PPC_CCNT counter), it increases the PPC_CCNT by one every cycle from its initial value until you stop counting. Any value read or written from/to the counter registers during the counting is invalid.

An example of steps to measure miss rate is:

- Select events: for example, set MMU_CFG.Sel_PPMU_event = 0x0 for L1-TLB miss event.
- Enable events: set MMU_CTRL.Enable_PPC_event = 0x1
- Enable PPC interrupt: set PPC_INTENS = 0x8000_000F
- Enable PPC counters: set PPC_CNTENS = 0x8000_000F
- Clear PPC overflow flags: set PPC_FLAG = 0x8000_000F
- Reset all PPC counters: set PPC_PMNC = 0x0000_0006
- Setup sampling duration cycle time: for example, measure miss rate during 10,000 (0x2710) cycles, set PPC_CCNT = 0xFFFF_D8EF (0xFFFF_FFFF – 0x2710)
- Start all PPC counters: set PPC_PMNC = 0x0000_0001
- Generates interrupt from PPC due to overflow on PPC_CCNT
- Stop all PPC counters: set PPC_PMNC = 0x0000_0000
- Read PPC_FLAG to check if interrupt has been generated by PPC cycle counter overflow.
- Read PPC counter values:
  - PPC_PMCNT0: total number of AW transactions
  - PPC_PMCNT1: total number of AR transactions
  - PPC_PMCNT2: number of AW transactions causing TLB miss
  - PPC_PMCNT3: number of AR transactions causing TLB miss
- The formula to calculate AW/AR miss rates is:
  - AW miss rate = (value of PPC_PMCNT2)/(value of PPC_PMCNT0)
  - AR miss rate = (value of PPC_PMCNT3)/(value of PPC_PMCNT1)

17.4 Register Descriptions

Use the SysMMU registers to configure, control, or provide its operations and status of certain features.

17.4.1 Register Map Summary

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)

Register	Offset	Description	Reset Value
MMU_CTRL	0x0000	SysMMU control register	0x0000_0000
MMU_CFG	0x0004	SysMMU configuration register	0x0000_0000
MMU_STATUS	0x0008	SysMMU status register	0x0000_0000
MMU_FLUSH	0x000C	Flush all TLB entry	0x0000_0000
MMU_FLUSH_ENTRY	0x0010	Flush an entry by virtual address	0x0000_0000
PT_BASE_ADDR	0x0014	Page table base address	0x0000_0000
INT_STATUS	0x0018	Interrupt status	0x0000_0000
INT_CLEAR	0x001C	Clear interrupt	0x0000_0000
INT_MASK	0x0020	Masking interrupt	0x0000_0000
PAGE_FAULT_ADDR	0x0024	Virtual address causing last page fault or bus error	0x0000_0000
AW_FAULT_ADDR	0x0028	Virtual address causing last fault caused during AW address translation	0x0000_0000
AR_FAULT_ADDR	0x002C	Virtual address causing last fault caused during AR address translation	0x0000_0000
DEFAUT_SLAVE_ADDR	0x0030	Physical address of default slave	0x0000_0000
MMU_VERSION	0x0034	SysMMU version information	0x2000_0020 (NOTE)
PPC_PMNC	0x0800	Performance monitor control register	0x0000_0000
PPC_CNTENS	0x0810	Count enable set register	0x0000_0000
PPC_CNTENC	0x0820	Count enable clear register	0x0000_0000
PPC_INTENS	0x0830	Interrupt enable set register	0x0000_0000
PPC_INTENC	0x0840	Interrupt enable clear register	0x0000_0000
PPC_FLAG	0x0850	Overflow flag status register	0x0000_0000
PPC_CCNT	0x0900	Cycle count register	0x0000_0000
PPC_PMCNT0	0x0910	Performance monitor count0 registers (total number of AW transactions)	0x0000_0000
PPC_PMCNT1	0x0920	Performance monitor count1 registers (total number of AR transactions)	0x0000_0000
PPC_PMCNT2	0x0930	Performance monitor count2 registers (number of AW transactions causing TLB miss)	0x0000_0000
PPC_PMCNT3	0x0940	Performance monitor count3 registers (number of AR transactions causing TLB miss)	0x0000_0000

NOTE: Value is implementation dependant.

17.4.1.1 MMU_CTRL

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	RW	Reserved (Should be zero and read as zero)	28'b0
Enable_PPMU_EVENT	[3]	RW	0 = Disables PPMU event 1 = Enables PPMU event	1'b0
Enable_INT	[2]	RW	0 = Disables interrupt 1 = Enables interrupt	1'b0
Block_MMU	[1]	RW	This register is effective only when it enables MMU. 0 = Un-block MMU 1 = Block MMU	1'b0
Enable_MMU	[0]	RW	0 = Disables MMU (bypass mode) 1 = Enables MMU	1'b0

17.4.1.2 MMU_CFG

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	RW	Reserved (Should be zero and read as zero)	20'b0
Sel_PPMU_event	[11]	RW	0 = Generates L1-TLB miss event 1 = Generates L2-TLB miss event	1'b0
PTW_QOS	[10:7]	RW	PL301 r2p0 ARQOS value for PTW	4'b0
PTN	[6:4]	RW	Controls size of first level page table. It changes the virtual address range also according to the size shown in parentheses. b000 = 16 KB (0x0000_0000 to 0xffff_ffff) b001 = 8 KB (0x0000_0000 to 0x7fff_ffff) b010 = 4 KB (0x0000_0000 to 0x3fff_ffff) b011 = 2 KB (0x0000_0000 to 0x1fff_ffff) b100 = 1 KB (0x0000_0000 to 0x0fff_ffff) b101 = 512 B (0x0000_0000 to 0x07ff_ffff) b110 = 256 B (0x0000_0000 to 0x03ff_ffff) b111 = 128 B (0x0000_0000 to 0x01ff_ffff)	3'b0
Enable_security_prot	[3]	RW	0 = Disables security protection check 1 = Enables security protection check	1'b0
Enable_access_prot	[2]	RW	0 = Disables access protection check 1 = Enables access protection check	1'b0
AFE	[1]	RW	0 = Verifies access protection with AP[2:1] of page table entry 1 = Verifies access protection with AP[2:0] of page table entry	1'b0
L1TLB_replace	[0]	RW	0 = Round-robin 1 = LRU	1'b0

17.4.1.3 MMU_STATUS

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	R	Reserved. (Read as zero)	28'b0
AR_pend	[3]	R	1 = AR pending state	1'b0
AW_pend	[2]	R	1 = AW pending state	1'b0
L2TLB_PTW	[1]	R	1 = L2-TLB access or page table walk state	1'b0
Blocked	[0]	R	1 = Blocks MMU	1'b0

17.4.1.4 MMU_FLUSH

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	W	Reserved. (Should be zero)	31'b0
Flush_all_entry	[0]	W	1 = Flush all entries of both L1-TLB and L2-TLB	1'b0

17.4.1.5 MMU_FLUSH_ENTRY

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
VPN	[31:12]	W	Virtual Page Number	20'b0
RSVD	[11:1]	W	Reserved. (Should be zero)	11'b0
Flush_entry	[0]	W	1 = Flush entries of both L1-TLB and L2-TLB corresponding to the VPN field.	1'b0

17.4.1.6 PT_BASE_ADDR

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PT_base_addr	[31:7]	RW	Page table base address, bit[31:7]	25'b0
RSVD	[6:0]	RW	Reserved. (Should be zero and read as zero)	7'b0

17.4.1.7 INT_STATUS

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	R	Reserved. (Read as zero)	24'b0
AW_Access_fault	[7]	R	AW Access Protection Fault	1'b0
AW_Security_fault	[6]	R	AW Security Protection Fault	1'b0
AR_Access_fault	[5]	R	AR Access Protection Fault	1'b0
AR_Security_fault	[4]	R	AR Security Protection Fault	1'b0
Bus_error	[3]	R	Bus Error	1'b0
AW_MH_fault	[2]	R	AW Multi-hit Fault	1'b0
AR_MH_fault	[1]	R	AR Multi-hit Fault	1'b0
Page_fault	[0]	R	Page Fault	1'b0

17.4.1.8 INT_CLEAR

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	W	Reserved. (Should be zero)	24'b0
AW_access_fault	[7]	W	1 = Clears AW access protection fault	1'b0
AW_security_fault	[6]	W	1 = Clears AW security protection fault	1'b0
AR_access_fault	[5]	W	1 = Clears AR access protection fault	1'b0
AR_security_fault	[4]	W	1 = Clears AR security protection fault	1'b0
Bus_error	[3]	W	1 = Clears bus error	1'b0
AW_MH_fault	[2]	W	1 = Clears AW multi-hit fault	1'b0
AR_MH_fault	[1]	W	1 = Clears AR multi-hit fault	1'b0
Page_fault	[0]	W	1 = Clears page fault	1'b0

17.4.1.9 INT_MASK

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	RW	Reserved. (Should be zero and read as zero)	24'b0
AW_Access_fault	[7]	RW	1 = Masks AW access protection fault	1'b0
AW_Security_fault	[6]	RW	1 = Masks AW security protection fault	1'b0
AR_Access_fault	[5]	RW	1 = Masks AR access protection fault	1'b0
AR_Security_fault	[4]	RW	1 = Masks AR security protection fault	1'b0
Bus_error	[3]	RW	1 = Masks bus error	1'b0
AW_MH_fault	[2]	RW	1 = Masks AW multi-hit fault	1'b0
AR_MH_fault	[1]	RW	1 = Masks AR multi-hit fault	1'b0
Page_fault	[0]	RW	1 = Masks page fault	1'b0

17.4.1.10 PAGE_FAULT_ADDR

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Page_Fault_addr	[31:0]	R	Virtual address causing last page fault or bus error	32'b0

17.4.1.11 AW_FAULT_ADDR

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
AW_Fault_addr	[31:0]	R	Virtual address causing last AW multi-hit, access, or security fault	32'b0

17.4.1.12 AR_FAULT_ADDR

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
AR_Fault_addr	[31:0]	R	Virtual address causing last AR multi-hit, access, or security fault	32'b0

17.4.1.13 DEFAUT_SLAVE_ADDR

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Default_slave_addr	[31:4]	RW	Physical address of default slave[31:4]	28'b0
RSVD	[3:0]	RW	Reserved. (Should be zero and read as zero)	4'b0

17.4.1.14 MMU_VERSION

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0034, Reset Value = 0x2000_0020 (NOTE)

Name	Bit	Type	Description	Reset Value
Major_arch_version	[31:28]	R	Major Architecture Version	4'b2
Minor_arch_version	[27:21]	R	Minor Architecture Version	7'b0
RTL_version	[20:7]	R	RTL Revision Version	14'b0 (NOTE)
L1TLB_size	[6:0]	R	Number of L1-TLB entries	7'd32 (NOTE)

NOTE: Value is implementation dependant.

17.4.1.15 PPC_PMNC

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0800, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	RW	Reserved	12'b0
RSVD	[19:17]	RW	Reserved	3'b0
START MODE	[16]	RW	PPMU Start Mode 0 = Software (by CPU) 1 = Hardware (by SYSCON)	1'b0
RSVD	[15:5]	RW	Reserved	11'b0
RSVD	[4]	RW	Reserved	1'b0
CC DIVIDER	[3]	RW	Cycle Count Divider 0 = Counts every pRcessor clock cycle and reset value 1 = Counts every 64th pRcessor clock cycle	1'b0
CC RESET	[2]	W	Cycle Counter Reset 0 = No action 1 = Resets cycle counter and CCNT to zeR	1'b0
PPMU COUNTER RESET	[1]	W	Performance Counter Reset 0 = No action 1 = Resets all performance counters to zeR	1'b0
PPMU ENABLE	[0]	RW	Enable bit 0 = Disables all counters including CCNT 1 = Enables all counters including CCNT When you read it, 1 means it is counting and 0 means it is idle (stops counting). You can write it only when the start mode is set to be 0 (CPU starts the PPMU). Currrently, you can write this bit by 1 to start counting and write it by 0 to stop the counting. When the start mode is set to be 1 (SYSCON starts the PPMU), you only can read it and get the status of the PPMU. Currently, external trigger contRls the PPMU. When external trigger is 1, counting starts. When external trigger is 0, counting stops.	1'b0

17.4.1.16 PPC_CNTENS

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0810, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	Enables Cycle Counter	1'b0
RSVD	[30:4]	RW	Reserved	27'b0
PMCNT3	[3]	RW	Enables Counter 3	1'b0
PMCNT2	[2]	RW	Enables Counter 2	1'b0
PMCNT1	[1]	RW	Enables Counter 1	1'b0
PMCNT0	[0]	RW	Enables Counter 0	1'b0

17.4.1.17 PPC_CNTENC

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0820, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	Disables Cycle Counter	1'b0
RSVD	[30:4]	RW	Reserved	27'b0
PMCNT3	[3]	RW	Disables Counter 3	1'b0
PMCNT2	[2]	RW	Disables Counter 2	1'b0
PMCNT1	[1]	RW	Disables Counter 1	1'b0
PMCNT0	[0]	RW	Disables Counter 0	1'b0

17.4.1.18 PPC_INTENS

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0830, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	CCNT overflow interrupt enable	1'b0
RSVD	[30:4]	RW	Reserved	27'b0
PMCNT3	[3]	RW	Counter 3 overflow interrupt enable	1'b0
PMCNT2	[2]	RW	Counter 2 overflow interrupt enable	1'b0
PMCNT1	[1]	RW	Counter 1 overflow interrupt enable	1'b0
PMCNT0	[0]	RW	Counter 0 overflow interrupt enable	1'b0

17.4.1.19 PPC_INTENC

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0840, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	CCNT overflow interrupt disable	1'b0
RSVD	[30:4]	RW	Reserved	27'b0
PMCNT3	[3]	RW	Counter 3 overflow interrupt disable	1'b0
PMCNT2	[2]	RW	Counter 2 overflow interrupt disable	1'b0
PMCNT1	[1]	RW	Counter 1 overflow interrupt disable	1'b0
PMCNT0	[0]	RW	Counter 0 overflow interrupt disable	1'b0

17.4.1.20 PPC_FLAG

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0850, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	Cycle counter overflow flag	1'b0
RSVD	[30:4]	RW	Reserved	27'b0
PMCNT3	[3]	RW	Counter 3 overflow flag	1'b0
PMCNT2	[2]	RW	Counter 2 overflow flag	1'b0
PMCNT1	[1]	RW	Counter 1 overflow flag	1'b0
PMCNT0	[0]	RW	Counter 0 overflow flag	1'b0

17.4.1.21 PPC_CCNT

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0900, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31:0]	RW	CCNT register contains an event count	32'b0

17.4.1.22 PPC_PMCNT0

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0910, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PMCNT	[31:0]	RW	PMCNT register contains an event count (total number of AW transactions)	32'b0

17.4.1.23 PPC_PMCNT1

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0920, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PMCNT	[31:0]	RW	PMCNT register contains an event count (total number of AR transactions)	32'b0

17.4.1.24 PPC_PMCNT2

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0930, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PMCNT	[31:0]	RW	PMCNT register contains an event count (number of AW transactions causing L1-TLB or L2-TLB miss)	32'b0

17.4.1.25 PPC_PMCNT3

- Base Address: 0x1362_0000 (SMMUMFC_L)
- Base Address: 0x1363_0000 (SMMUMFC_R)
- Address = Base Address + 0x0940, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PMCNT	[31:0]	RW	PMCNT register contains an event count (number of AR transactions causing L1-TLB or L2-TLB miss)	32'b0

18 Dynamic Memory Controller

18.1 Overview

DMC is an Advanced Microcontroller Bus Architecture (AMBATM) AXI compliant slave to interface an external JEDEC DDR3 device. To support a high speed memory device, the controller uses the SEC SDRAM PHY interface. The controller includes an advanced embedded scheduler to utilize memory device efficiently and an optimized pipeline stage to minimize latency.

18.1.1 Features

Features of DMC are:

- Compatible with Joint Electron Device Engineering Council (JEDEC) DDR3 SDRAM specification.
- Dual slave ports compatible with AMBA3 AXI specification for memory accesses.
- Dual slave ports compatible with AMBA3 APB specification for programming registers.
- Uses the SEC SDRAM PHY interface to support high-speed memory devices.
- Supports dual memory channels with/without channel interleaving.
- Configures granularity of memory channel interleaving.
- Supports up to two external chip selects and 4/8 banks per a chip select.
- Supports 512 Mb, 1 Gb, 2 Gb, 4 Gb and 8 Gbit density per a chip select.
- Supports 16/32-bit wide memory data width.
- Optimized pipeline stage for low latency.
- Supports Quality of Service (QoS) scheme to ensure low latency for real-time applications.
- The advanced embedded scheduler enables out-of-order operations to utilize memory device efficiently.
- Detects AXI RAR/RAW/WAR/WAW hazards automatically.
- Supports early Write response.
- Supports excellent chip/bank interleaving and memory interrupting.
- Supports AMBA AXI low-power channel for systematic power control.
- Adapts to various low-power schemes to reduce the dynamic and static current of memory.
- Supports outstanding exclusive accesses.
- Supports bank selective precharge policy.
- Accommodates the embedded performance monitor.
- Supports 1:1/1:2 synchronous operations between AXI bus and memory clock domain.
- Supports clock frequency up to 200 MHz for AXI and 400 MHz for memory (SEC 32LP RVT).

18.2 Block Diagram

Figure 18-1 illustrates the overall block diagram of DMC.

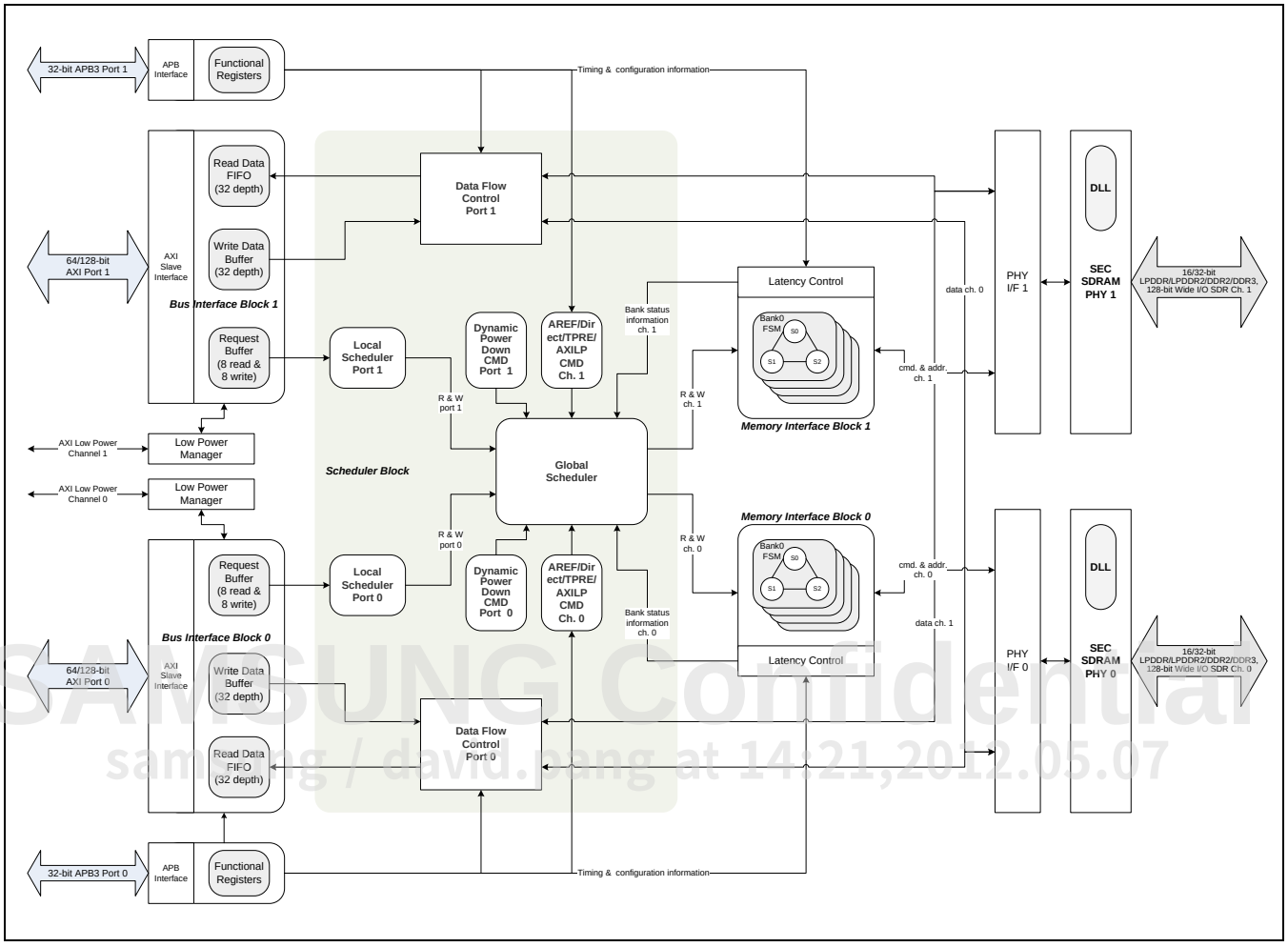


Figure 18-1 Block Diagram of DMC

The block diagram shows:

- Bus interface block
- Scheduler block
- Memory interface block. This block connects and interfaces with SEC SDRAM PHYs.

Bus interface block saves the bus transactions for memory access. It is derived from the AXI slave port to the command queue. Additionally, it saves the Write data to the Write buffer or sends the Read data to the Master through the AXI bus. It also acts as a Read First In First Out FIFO if AXI Masters are not ready or in 1:2 synchronous clock mode. It has an Advanced Peripheral Bus (APB) interface for special function registers/direct commands and an AXI low-power channel interface.

Scheduler block uses the memory bank Finite State Machine (FSM) information to arbitrate the bus transactions in the command queues. It transforms the commands into a memory command type, which is sent to the memory interface block. It also controls the Write and Read data flow between the memory and the AXI bus.

According to the memory command, the memory interface block updates each memory bank state. It is derived from the scheduler and returns the bank state to the scheduler. It creates a memory command depending on the memory latency and sends the command to SEC SDRAM PHYs through the PHY interface.

18.3 Initialization

You must power up and initialize SDRAM devices in a predefined manner. Other non-specific operational procedures may result in undefined operation. An initialization procedure consists of three procedures:

1. PHY DLL initialization
2. Setting controller register
3. Memory initialization

For initializing memory, refer to JEDEC specifications and datasheets of memory devices for more details. According to the memory types, initialization sequences are as follows.

18.3.1 DDR3

Use the sequence given here to initialize DDR3 devices. Unless specified otherwise, these steps are mandatory.

1. Apply power. RESET# needs to be maintained for minimum 200us with stable power. CKE is pulled “Low” anytime before RESET# being de-asserted (min. time 10ns)

2. If on die termination is required, enable PhyControl1.term_write_en, PhyControl1.term_read_en.

3. If ZQ calibration is required, disable PhyZQControl.ctrl_zq_mode_noterm and enable PhyZQCon-trol.ctrl_zq_start so that the PHY automatically calibrates the I/Os to match the driving and termination impedance by referencing resistor value of an external resistor and updates the matched value during auto re-fresh cycles.

4. Set the PhyControl0.ctrl_start_point and PhyControl0.ctrl_inc bit-fields to correct value according to clock frequency. Set the PhyControl0.ctrl_dll_on bit-field to ‘1’ to activate the PHY DLL.

5. DQS Cleaning: set the PhyControl1.ctrl_shiftc and PhyControl1.ctrl_offsetc bit-fields to the proper value according to clock frequency, board delay and memory tDQSCK parameter.

6. Set the PhyControl0.ctrl_start bit-field to "1".

7. Set the ConControl. At this moment, an auto refresh counter should be off.

8. Set the MemControl. At this moment, all power down modes and periodic ZQ(pzq_en) should be off.

9. Set the MemConfig0 register. If there are two external memory chips, also set the MemConfig1 register.

10. Set the PrechConfig and PwrdnConfig registers.

11. Set the TimingAref, TimingRow, TimingData and TimingPower registers according to memory AC parame-ters.

12. If QoS scheme is required, set the QosControl0~15 and QosConfig0~15 registers.

13. Wait for the PhyStatus0.ctrl_clock and PhyStatus0.ctrl_flock bit-fields to change to ‘1’. Check whether PHY DLL is locked.

14. PHY DLL compensates the changes of delay amount caused by PVT variation during memory operation. Therefore, it should not be off for reliable operation. It can be off except runs at low frequency. If off mode is used, set the PhyControl0.ctrl_force bit-field to the correct value according to the PhySta-tus0.ctrl_lock_value[9:2] bit-field for fix delay amount. Clear the PhyControl0.ctrl_dll_on bit-field to turn off PHY DLL.

15. Set the PhyControl1.fp_resync bit-field to ‘1’ to update DLL information.

16. Set the PhyControl1.fp_resync bit-field to ‘0’.

17. Confirm that after RESET# is de-asserted, 500 us have passed before CKE becomes active.

18. Confirm that clocks(CK, CK#) need to be started and stabilized for at least 10 ns or 5 tCK (which is larger) before CKE goes active.

19. Issue a NOP command using the DirectCmd register to assert and to hold CKE to a logic high level.

20. Wait for tXPR(max(5nCK,tRFC(min)+10ns)) or set tXP to tXPR value before step 17. If the system set tXP to tXPR, then the system must set tXP to proper value before normal memory operation.

21. Issue an EMRS2 command using the DirectCmd register to program the operating parameters. Dynamic ODT should be disabled. A10 and A9 should be low.

22. Issue an EMRS3 command using the DirectCmd register to program the operating parameters.

23. Issue an EMRS command using the DirectCmd register to enable the memory DLL.

24. Issue a MRS command using the DirectCmd register to reset the memory DLL.

25. Issues a MRS command using the DirectCmd register to program the operating parameters without resetting the memory DLL.

26. Issues a ZQINIT commands using the DirectCmd register.

27. If there are two external memory chips, perform steps 19 ~ 26 procedures for chip1 memory device.

28. Set the ConControl to turn on an auto refresh counter.

29. If power down modes or periodic ZQ(pzq_en) are required, set the MemControl register.

18.4 Address Mapping

DMC modifies the address of the AXI transaction that is derived from the AXI slave port into a memory address.

Memory address mapping consists of:

- Chip select
- Bank address
- Row address
- Column address
- Memory data width

To map chip select of memory device to a specific area of the address map, set the chip_base and chip_mask bit-fields of the MemConfig0 register. (Refer to *18.12 Register Description* for more details.) If chip1 of the memory device exists, ensure to set the MemConfig1 register. Then, the AXI masters that request the AXI address are divided into the AXI base and AXI offset addresses. The AXI base address activates the appropriate memory chip select and it maps the AXI offset address to a memory address according to the bank, row, column number, and data width set by the MemConfig0/1 and MemControl register.

There are two ways to map the AXI offset address

- Linear mapping
- Interleaved mapping

18.4.1 Linear Mapping

Figure 18-2 illustrates the linear address mapping.

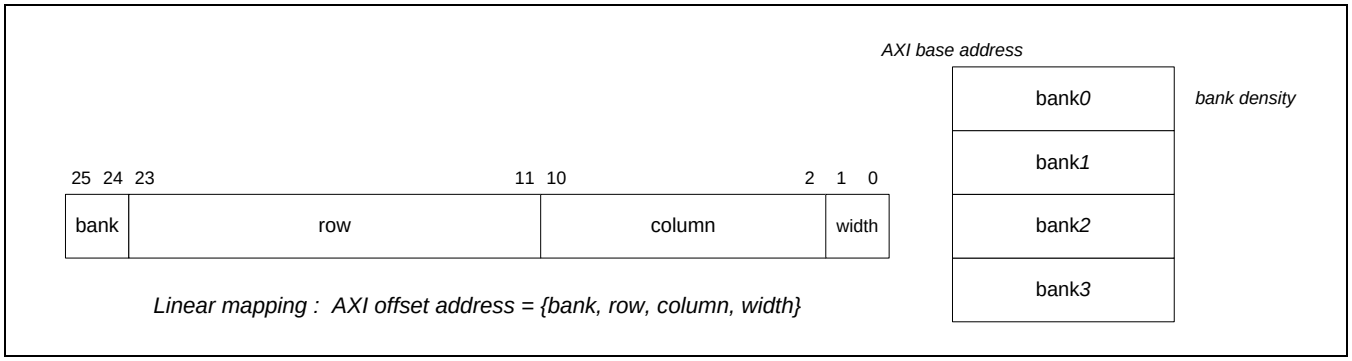


Figure 18-2 Linear Address Mapping

The linear mapping method maps the AXI address in the order of bank, row, column, and width. As the bank address does not change for at least one bank size, applications that use linear address mapping have a high possibility to access the same bank.

18.4.2 Interleaved Mapping

Figure 18-3 illustrates the interleaved address mapping.

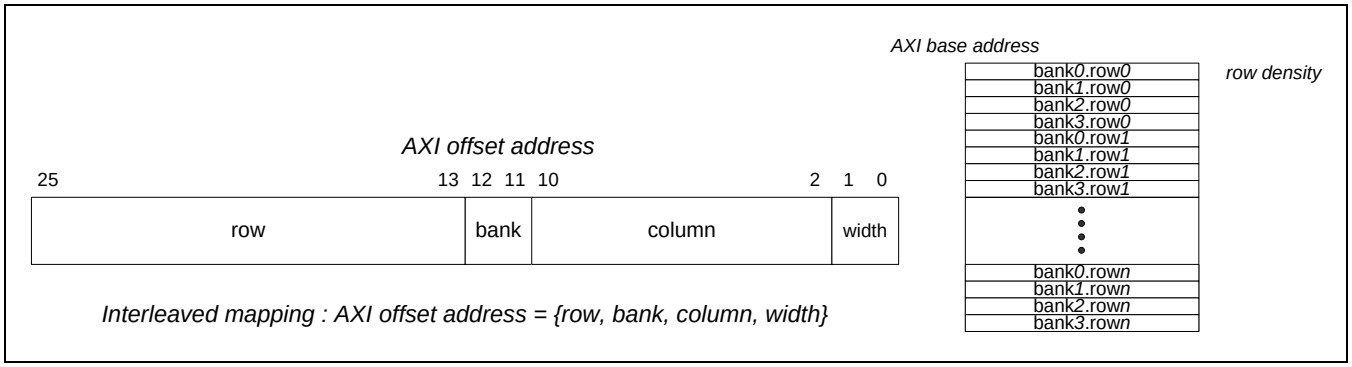


Figure 18-3 Interleaved Address Mapping

The interleaved mapping method also maps the AXI address in the order of row, bank, column, and width. The difference between linear and interleaved mapping methods is that the bank and row order are different. For accesses whose address exceeds a row size, interleaved mapping accesses a different bank. Therefore, applications that use interleaved mapping access numerous banks. By accessing numerous banks, the performance is better. However, the power consumption is more.

18.5 Low-Power Operation

The controller executes a low power-memory operation in five ways. They are:

- AXI low-power channel
- Dynamic power down
- Dynamic self refresh
- Clock stop
- Direct command

Each feature is independent of each other and executed at the same time.

18.5.1 AXI Low-Power Channel

The controller has an AXI low-power channel interface to communicate with low-power management units such as the system controller. This low-power channel interface makes the memory device enter into self-refresh mode. To request through the AXI low power channel, refer to the chip1_empty and chip0_empty bit-fields of ConControl register to verify if the command queue is currently empty.

18.5.2 Dynamic Power Down

The SDRAM device has an active/precharge power-down mode. The memory device enters this mode if CKE becomes LOW. To enter active power-down mode, at least one row of a bank must be open. To enter precharge power-down mode, CKE must be low.

If AXI transaction does not enter the controller and the command queue becomes empty for a specific number of cycles (PwrdnConfig.dpwrdn_cyc register), the controller changes the memory state of device to active/precharge power-down automatically. Then, there are two ways to enter the active/precharge power-down state. They are:

- Active/precharge power-down mode: Enter power down without considering whether there is a row open or not.
- Forced precharge power-down mode: Enter power down after closing all banks.

When a new AXI transaction enters the controller, the controller automatically wakes up the memory device from power-down state and executes in a normal operation state.

18.5.3 Dynamic Self Refresh

Similar to the dynamic power down feature, when the command queue is empty for a specific amount of cycles (PwrdnConfig.dsref_cyc register), the memory device enters self-refresh mode. As the exiting self-refresh mode requires many cycles, it is recommended that, preferring a greater cycle size for dynamic self-refresh entry to dynamic power down.

18.5.4 Clock Stop

To reduce the I/O power of the memory device and the controller, stop the clock when the LPDDR2-S4 is in idle mode or self-refresh mode. When you enable this feature, the controller automatically executes the clock stop feature.

18.5.5 Direct Command

Use the direct command feature to send a command to the memory device through the APB3 port. By doing this, you can force the memory device to enter active/precharge power down, self-refresh, or deep power-down mode.

18.6 Precharge Policy

The controller decides precharge policy in two ways. They are:

- Bank selective precharge policy
- Timeout precharge

18.6.1 Bank Selective Precharge Policy

As applications have different page policy preferences, it is difficult to decide on usage of policies: Open page policy and Close page (auto precharge policy). Instead of applying the page policy to all of the banks, the bank selects precharge policy that enables the user to choose a precharge policy for each bank (refer to PrechConfig.chip1_policy for more details). Therefore, you can assign certain applications to a bank that uses an open page policy, and other applications to a bank that uses a close page (auto precharge) policy.

Operations according to page policy are:

- Open page policy: After a READ or WRITE access, the controller leaves the accessed row open.
- Close page (auto precharge) policy: Immediately after issuing a READ or WRITE command, the controller issues an auto precharge to the bank.

18.6.2 Timeout Precharge

If a certain bank uses an open page policy, the controller leaves the accessed row open after a data access. When this happens and the bank that is left open is not scheduled for a specific number of cycles (PrechConfig.tp_cnt bit-field), the controller automatically issues a precharge command to close the bank.

Figure 18-4 illustrates the timing diagram of timeout precharge.

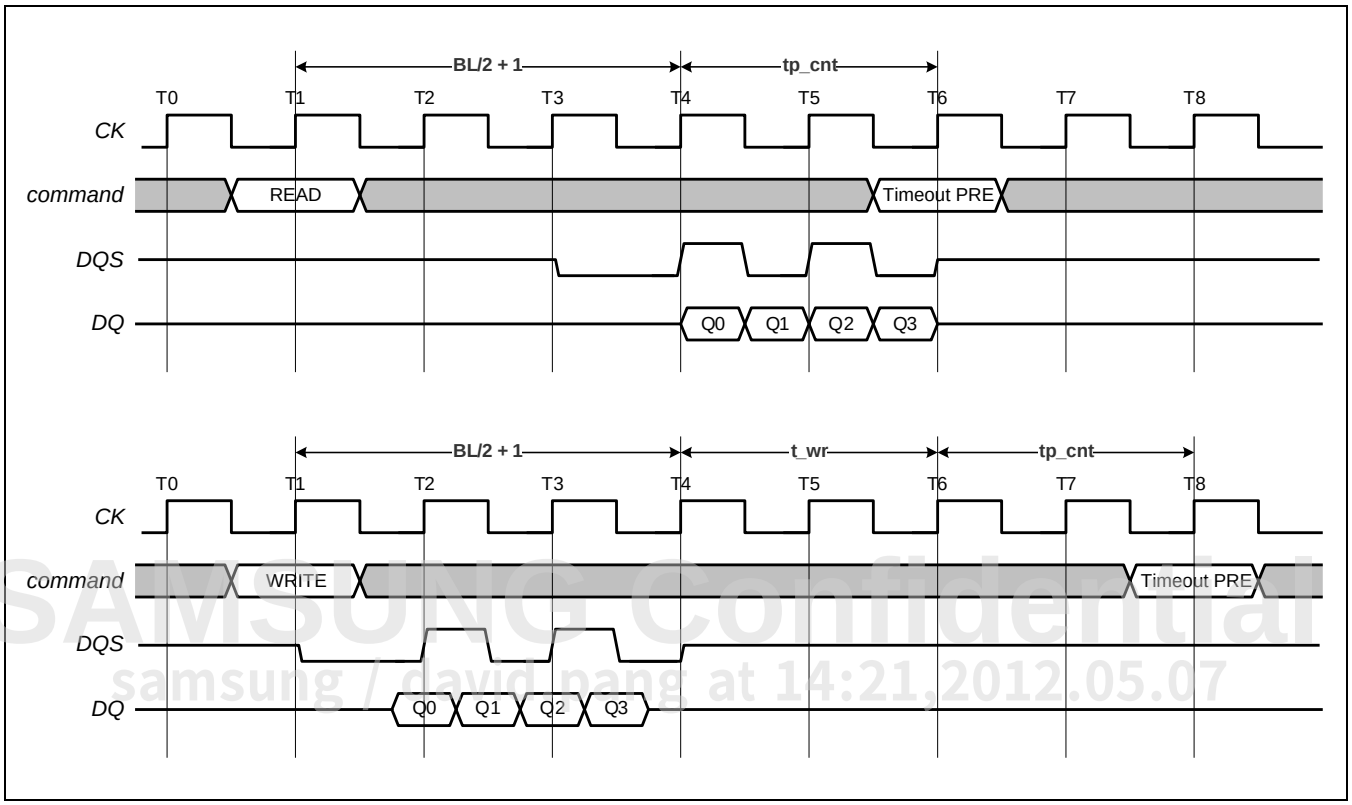


Figure 18-4 Timing Diagram Of Timeout Precharge

18.7 Quality of Service

DMC provides QoS feature to ensure low latency for real-time masters. Specifically, DMC uses timeout-based QoS enforcement scheme. When DMC receives an AXI transaction, it assigns a predefined QoS timeout value to the corresponding memory request for timeout. When the timer expires, the request is promoted to the highest priority for immediate selection during the arbitration stage.

DMC supports four types of QoS scheme. They are:

- AXI_ID-based QoS
- Fast QoS
- Default QoS
- AxQoS-based QoS

Note that DMC does not support combination of these schemes simultaneously. AXI_ID-based QoS scheme is always available However, users can choose only one of the {Fast QoS, Default QoS} or {AxQoS-based QoS} schemes by SFR setting. The subsections describe each QoS scheme in detail.

18.7.1 AXI_ID-Based QoS

There are 16 qos_cnt configuration registers (QoSControls) that have independent qos_masks. These configuration registers masks the AXID from one bit up to the AXID width. You can enable or disable each entry.

To set the timer:

If a transaction is received through the AXI bus, the qos_masks (QoSConfig(n).qos_mask) masks AXID from the 16 QoSControls that are enabled.

The controller compares the masked results to the qos_ids (QoSConfig(n).qos_id). When one of the results matches, it applies the qos_cnt (QoSControl(n).qos_cnt) value to the transaction. This happens before saving the value in the request buffer.

When results do not match, the controller applies either one of the Default QoS schemes or the AxQoS-based QoS scheme according to the SFR setting.

Table 18-1 QoS Bus Master ID

<table>
<tr><th></th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>F4D</td><td>0</td><td>0</td><td>0</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td><td>1</td></tr>
<tr><td>G2D</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>S</td><td>1</td><td>0</td></tr>
<tr><td>MFC(L)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>S</td><td>1</td><td>1</td><td rowspan="7">0</td></tr>
<tr><td>G3D</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>1</td><td>0</td></tr>
<tr><td>VP</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td><td rowspan="2">S</td><td rowspan="2">0</td><td rowspan="2">1</td></tr>
<tr><td>MIXER</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>1</td></tr>
<tr><td>Rotator</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td rowspan="2">S</td><td>0</td><td rowspan="2">0</td><td rowspan="2">0</td></tr>
<tr><td>MDMA2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>1</td></tr>
<tr><td>C2C</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>1</td></tr>
</table>

	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIMC0	0	x	x	x	x	S	0	0	0	0	0	0	0	0
FIMC1	0	x	x	x	x		0	0	1					
FIMC2	0	x	x	x	x		0	1	0					
FIMC3	0	x	x	x	x		0	1	1					
JPEG	0	x	x	x	x		1	0	0					
FIMD0M0	0	0	0	0	0	0	x	0	S	0	0	1		
FIMD0M1	0	0	0	0	0	x	x	1						
FIMC_LITE0	0	0	0	0	0	x	S	0	0	0	1	0		
FIMC_LITE1	0	0	0	0	0	x		0	1					
FIMC_ISPV3.1	0	0	x	x	x	x		1	0					
FIMC_DRCV3.1	x	x	x	x	S	0	0	1	1					
FIMC_FDV3.1	x	x	x	x		0	1	1	1					
AXI_ISP_CX	x	x	x	x		1	0	1	1					
GPS	x	x	x	x	x	x	1	0	1	0	1	1		
AUDI	0	0	0	0	0	0	1	0	0					
USB_HOST	0	0	0	0	0	0	0	1	1					
USB_OTG	0	0	0	0	0	0								
SDMMC	0	0	0	0	0	0	0	1	0					
TSI	0	0	0	0	0	0								
PDMA0	0	0	x	x	x	x	0	0	0					
PDMA1	0	0	x	x	x	x	0	0	1					
MFC(R)	0	0	0	0	x	x	x	x	S	1	0	0		
AudioSS	0	0	0	0	0	x	x	x	x	1	0	1		

NOTE: When the master access SysMMU, 'S' bit is '1'

18.7.2 Fast QoS

To serve the latency sensitive transactions faster, the controller enables an adaptive QoS scheme called Fast QoS. When you use Fast QoS scheme, a master observes its FIFO level. When the level crosses a threshold, Fast QoS asserts a predefined AXI side band signal (qos_fast). This signal promotes the priority of the corresponding AXI transaction to the highest level by ensuring low latency.

Here is the detailed usage scenario: For Read transactions, a master assumes that when the master's Read FIFO occupancy goes under the 1/4th of full FIFO size, the master may assume that the read FIFO can reach an underrun state. Therefore, the master signals DMC to inform the master's state and the qos_cnt_f value. The master signals the controller to request to apply the qos_cnt_f value specified for the faster transaction that enables to have higher priority in scheduling over other masters' transactions.

For Write transactions, a master assumes that when the master's Write FIFO occupancy goes over the 3/4th of full FIFO size, the master may assume that the Write FIFO can reach a full state. Therefore, the master signals DMC to inform the master's state and the qos_cnt_f value. The master signals the controller to request to apply the qos_cnt_f value specified for the faster transaction that enables to have higher priority in scheduling over other masters' transactions

Figure 18-5 illustrates the adaptive DRAM QoS scheme configuration in SoC.

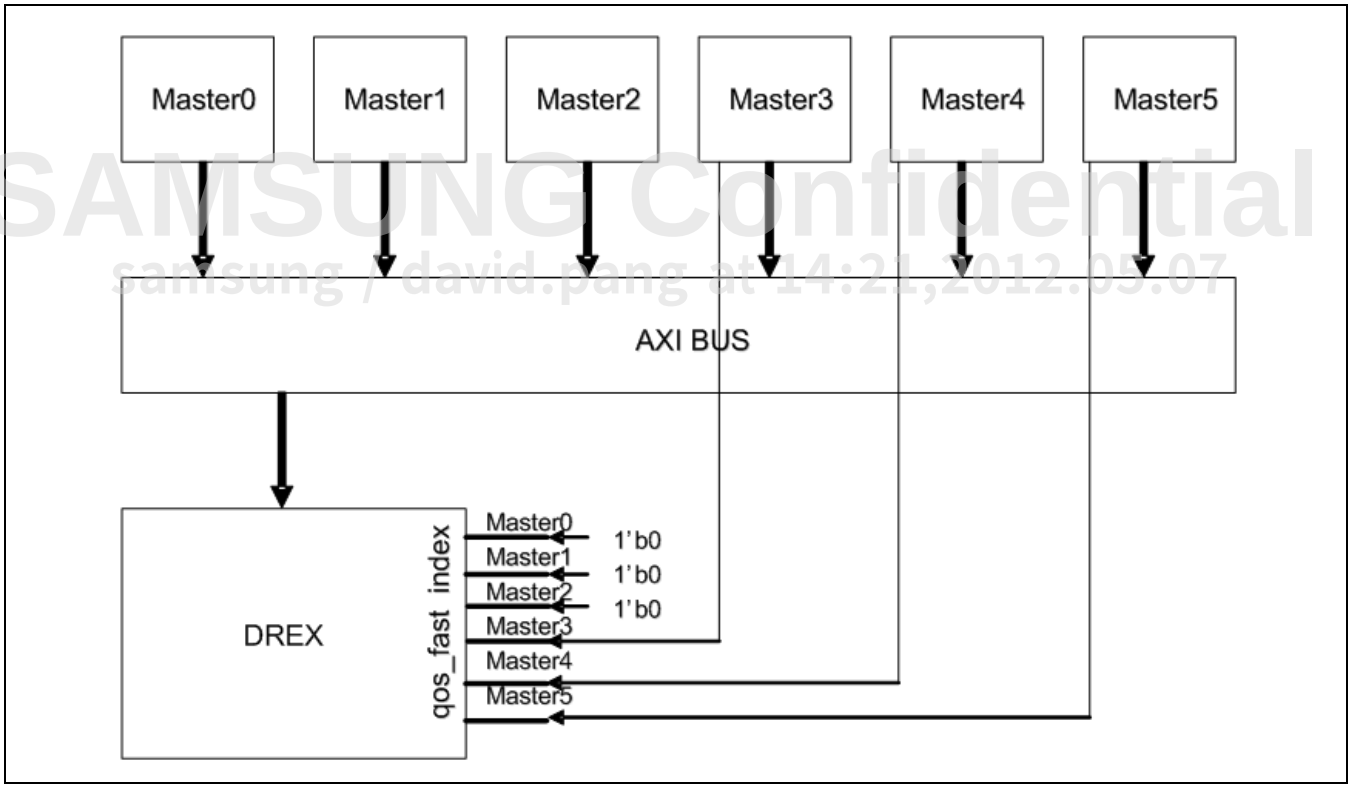


Figure 18-5 An Adaptive DRAM QoS Scheme Configuration

Each master is able to flag DMC by accessing a one-bit side-band channel. When the controller enables QoS fast (Concontrol.qos_fast_en), and the master that sent the signal raises the qos_fast flag, it applies qos_cnt_f instead of qos_cnt to the transaction.

18.7.3 Default QoS

When neither AXI_ID-based QoS scheme nor the Fast QoS scheme is applicable to a transaction, the controller assigns a default QoS counter (ConControl.timeout_cnt) to the transaction. The controller applies single default QoS counter to both Read and Write. Ensure that the default QoS scheme is unavailable when the QE enables AxQoS based QoS scheme.

18.7.4 AxQoS-Based QoS

When QE enables the AxQoS-based QoS scheme, DMC requires AxQoS value to accompany every incoming AXI transaction. This is urgent for each transaction. When DMC receives an AXI transaction, it decodes the AxQoS value and assigns a predefined timeout value (ConControl.timeout_cnt, QosTimeout0 and QosTimeout1) accordingly. DMC accepts AxQoS value ranging from 0 to 3, where 3 is the most urgent transaction.

18.8 Read Data Capture

A memory device that receives a READ command sends the data to the controller after Read latency (CAS latency). After clearing the DQS, the PHY uses the PHY DLL to phase shift the DQS 90 degrees. By using the shifted DQS, the PHY samples the Read data and saves the data into the Read data input FIFO, which is located inside the PHY. The controller then fetches the data from the PHY. While considering the Read latency and the Read fetch delay, and then the controller sends the data to the AXI read channel.

Figure 18-6 illustrates timing diagram of Read data capture (DDR3, Zero Delay, RL=3, rd_fetch=1).

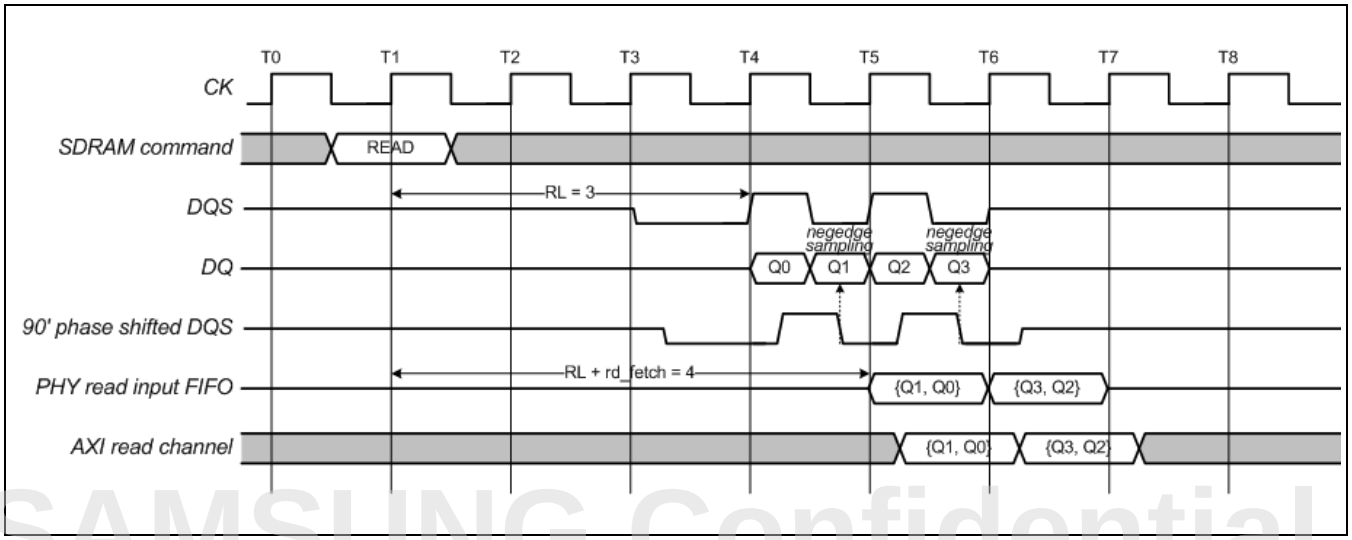


Figure 18-6 Timing Diagram Of Read data Capture (DDR3, zero delay, RL=3, rd_fetch=1)

DDR3 has an internal DLL which allows it to send the data after an exact amount of read latency. If you assume there are minimal or no board/PHY input delay, the controller sends the read DQs, sampled at a negative edge (Q1, Q3 sampling), from PHY read input FIFO to AXI read channel in 'read latency + 1(read fetch)' cycles. You can set the read fetch cycle using the **ConControl.rd_fetch** bit-field.

Figure 18-7 Timing Diagram Of Read Data Capture (DDR3, non-zero delay, RL=3, rd_fetch=2) illustrates timing diagram of Read data capture (DDR3, Non-zero Delay, RL=3, rd_fetch=1).

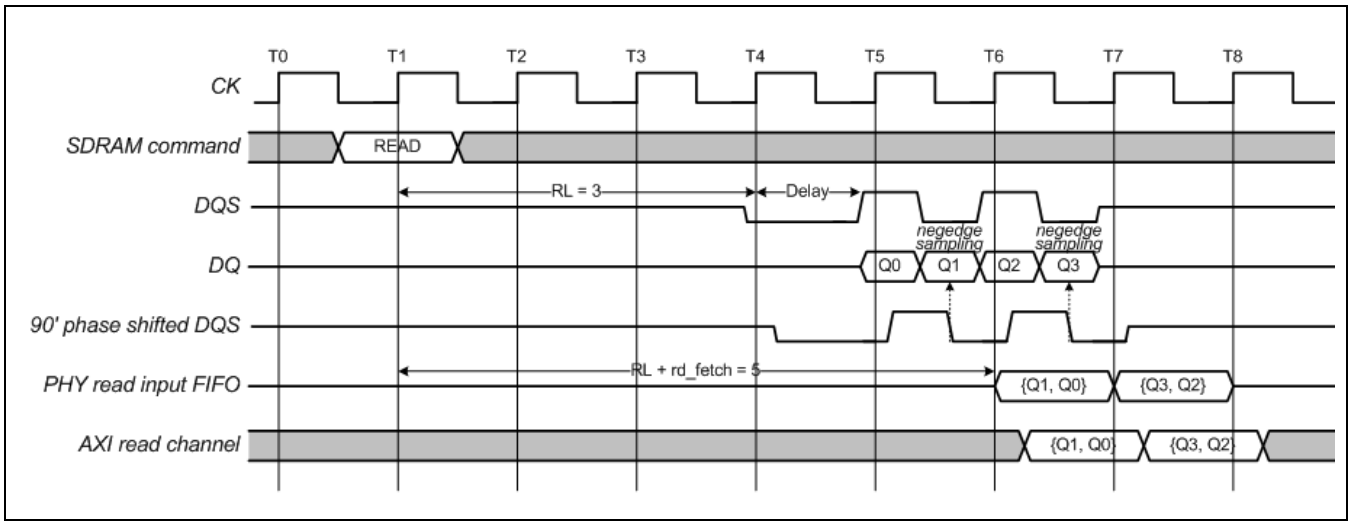


Figure 18-7 Timing Diagram Of Read Data Capture (DDR3, non-zero delay, RL=3, rd_fetch=2)

When there's an external delay, sampling at a negative edge happens at T5 and T6, which is one cycle slower than T4/T5 shown in *Figure 18-7* Timing Diagram Of Read Data Capture (DDR3, non-zero delay, RL=3, rd_fetch=2). Therefore, the read fetch cycle should be set to "2" since the sampled read data is saved into the read input FIFO slower.

Use this formula to calculate the DDR3 rd_fetch value:

rd_fetch (DDR3) = INT((Delay + 0.5T + 0.25T)/T) = INT(Delay/T + 0.75),
Delay: board delay + PHY input/output delay, T: clock period, INT(x): the rounded-up integer value of x

Figure 18-8 illustrates timing diagram of Read data capture (LPDDR2-S4, Zero Delay, RL=3, rd_fetch=1).

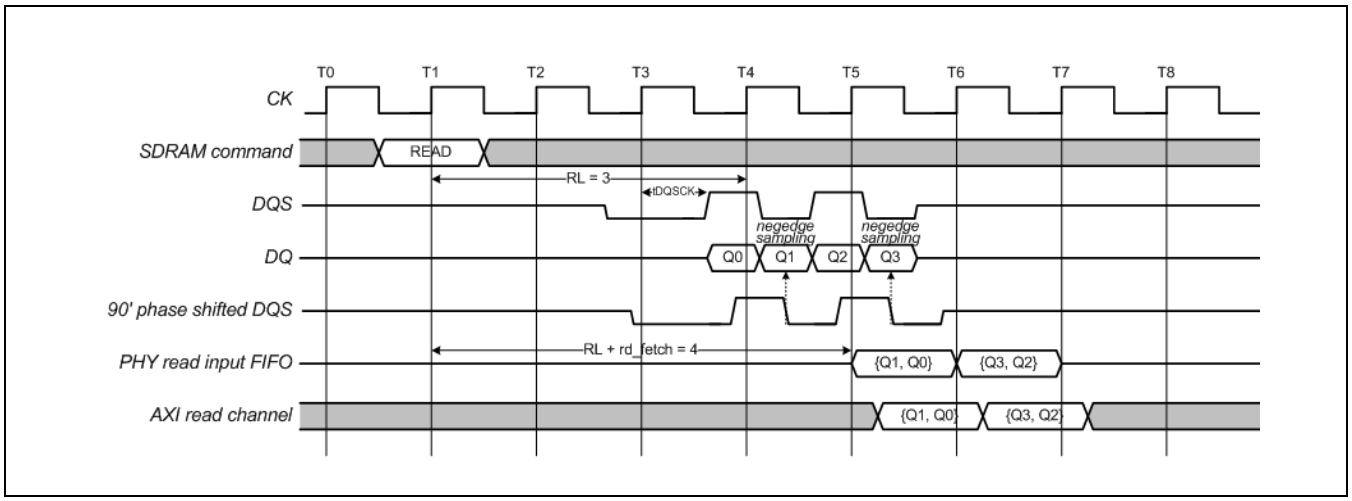


Figure 18-8 Timing Diagram Of Read Data Capture (LPDDR2-S4, Zero Delay, RL=3, rd_fetch=1)

An LPDDR2-S4 does not have an internal DLL. Without an internal DLL, it sends out the data after tDQSCK before the Read latency is complete. Even if the chip assumes zero delay, as tDQSCK becomes relatively large in

high frequencies, the Read fetch cycle should be set to 1.

Figure 18-9 illustrates timing diagram of Read data capture (LPDDR2-S4, Non-Zero Delay, RL=3, rd_fetch=2).

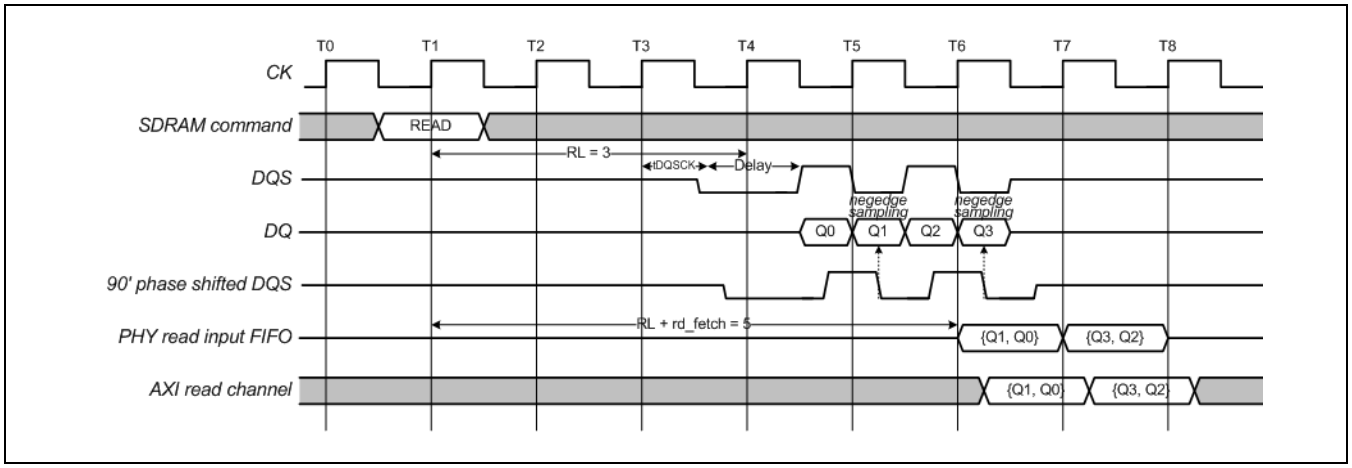


Figure 18-9 Timing Diagram Of Read Data Capture (LPDDR2-S4, Non-Zero Delay, RL=3, rd_fetch=2)

If a delay exists such as in Figure 18-9, a higher value should be assigned to rd_fetch.

Figure 18-10 illustrates timing diagram of Read data capture (LPDDR2-S4, Low Frequency, RL=3, rd_fetch=0).

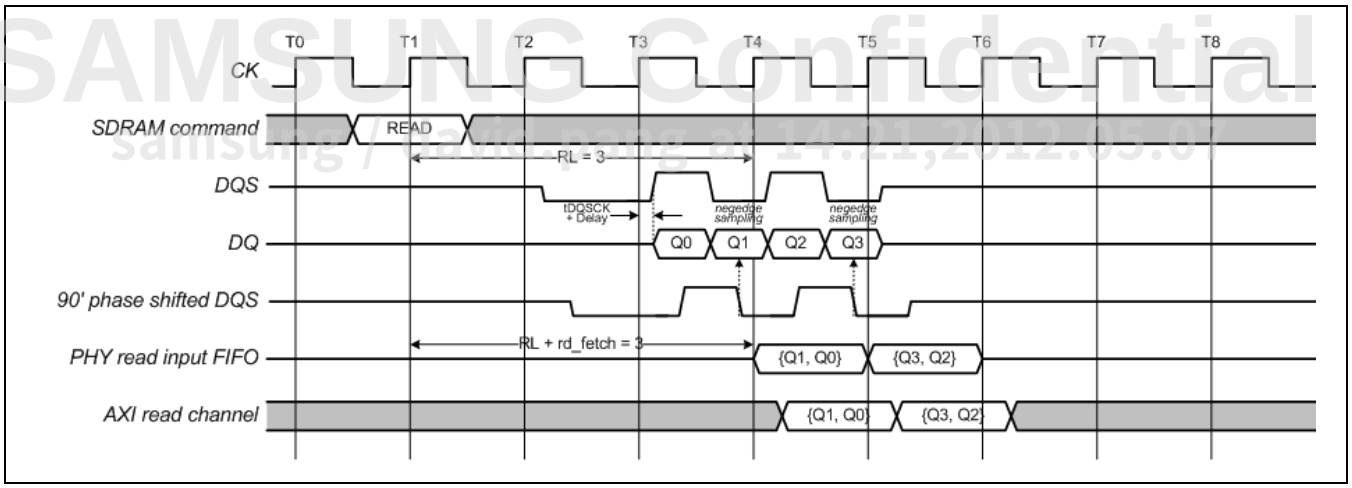


Figure 18-10 Timing Diagram Of Read Data Capture (LPDDR2-S4, Low Frequency, RL=3, rd_fetch=0)

tDQSCK + Delay is relatively smaller compared to the clock period during low frequencies as shown in Figure 18-10.

In this situation, negedge sampling occurs before Read latency and therefore Read fetch is set to 0.

Use this formula to calculate the LPDDR/LPDDR2-S4 rd_fetch value:

rd_fetch (LPDDR2-S4) = INT ((– 1 + Delay + 0.5T + 0.25T)/T) = INT (Delay/T – 0.25),
Delay: board delay + PHY input delay, T: clock period, INT(x): the rounded-up integer value of x

Therefore, if the value of Delay/T is less than 0.25, rd_fetch is set to 0.

18.9 Memory Channels Interleaving

As the capacity and bandwidth demand of SoCs exceed than what a single memory can offer, recent SoC designs have multiple memory devices and channels. The intertwining of demanding multimedia SOCs with its underlying memory subsystem is a problem. For example, multimedia IP playing with delicate allocation of data buffers among the multiple memories takes advantage of multiple memory channels for higher bandwidth. With the introduction of system Memory Management Unit (MMU), the multimedia IP might even lose control of physical address allocation for those data buffers.

DMC solves this complexity issue by logically combining the multiple memories into one bigger and faster memory. This is possible because of its memory channels interleaving capability. Also, DMC naturally improves the performance of master IPs, which previously utilized single memory channel by allowing them to access multiple memory channels.

When a master IP issues a memory request whose burst length is longer than an interleaving chunk size (which is configurable), DMC splits the burst into many small interleaving chunks and schedules the chunks for concurrent memory accesses. When you compare the case without interleaving, there is a chance of overall system performance speed-up by balanced usage of memory channels. However, there is a potential of higher bank conflicts that affects the performance negatively, which is an inherent property of interleaving.

Ensure that in modern DRAM devices, as the number of banks increases, the negative impact of bank conflicts decreases.

Figure 18-11 illustrates memory channels interleaving.

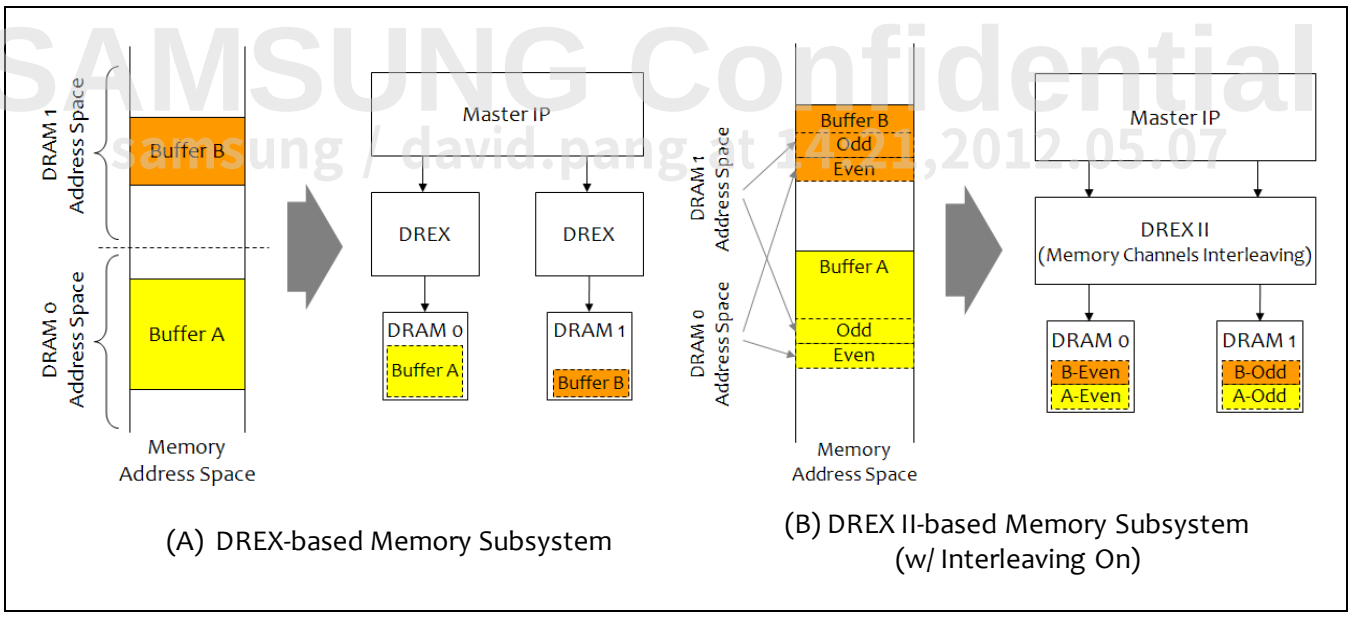


Figure 18-11 Memory Channels Interleaving

18.10 Performance Profiling

DMC provides performance monitoring capability based on event counters that are accessible through APB interface. Relevant registers are located on 0xFC to 0xF_1120.

DMC has two event counters: one for ACLK domain event counting (PPC_A) and the other for MCLK domain event counting (PPC_M).

Table 18-2 lists performance events in each domain.

It is possible to extract useful information from these event counts.

Table 18-2 List of Performance Events

Event Items		Clock Domain
1.	Total read requests counts	aclk
1-1.	Total read requests counts per bank	aclk
2.	Total write requests counts	aclk
2-1.	Total write requests counts per bank	aclk
3.	Splited read requests counts	aclk
4.	Splited write requests counts	aclk
5.	Read hit counts	mclk
5-1.	Read hit counts per bank	mclk
6.	Write hit counts	mclk
6-1.	Write hit counts per bank	mclk
7.	Read miss counts	mclk
7-1.	Read miss counts per bank	mclk
8.	Write miss counts	mclk
8-1.	Write miss counts per bank	mclk
9.	Timeout read miss counts	mclk
10.	Timeout write miss counts	mclk
11.	Hazard read miss counts	mclk
12.	Hazard write miss counts	mclk
13.	Power down mode cycle counts	mclk
14.	Self-refresh mode cycle counts	mclk
15.	DRAM data channel cycle used	mclk

Table 18-3 lists a few examples of performance events in each domain.

Table 18-3 List of Performance Information

Information	Event Expression
Total read requests	1
Total write requests	2
Total requests	1 + 2
Total splited read requests	3
Total splited write requests	4
Piority_level_timeout	Timeout read miss + Timeout write miss (9 + 10)
Piority_level_read_hit	5
Piority_level_write_hit	6
Piority_level_hazard_miss	(Hazard read miss + Hazard write miss) (11 + 12)
Piority_level_read_miss	(Read miss – Timeout read miss – Hazard read miss) (7 – 9 – 11)
Piority_level_write_miss	(Write miss – Timeout write miss – Hazard write miss) (8 – 10 – 12)
Powerdown and Self Refresh	13 + 14
Memory data transfer	15

For example, PL_read_hit can be obtained as shown in the steps here.It should read PPC_A and PPC_M.

Step1: Initialize State

- Mode Selection: Sets perev_en to 0x1, perev_mode_a to 0x0 and perev_mode_m to 0x5 in PerevConfig (Offset = 0x00FC). Then 4-bit events for PPC_A are {2'b0, read, write}. 4-bit events for PPC_M are {2'b0, readmiss, write miss}
- Interrupt Enable: Sets INTENS_PPC_A/M (Offset = 0xE030/0xF030) to 0x8000_000F
- Counter Enable: Sets CNTENS_PPC_A/M (Offset = 0xE010/0xF010) to 0x8000_000F

Step2: Start State

- Clear Overflow Flag Register: Sets FLAG_PPC_A/M (Offset = 0xE050, 0xF050) to 0x8000_000F
- All Counter Reset: Sets PMNC_PPC_A/M (Offset = 0xE000, 0xF000) to 0x0000_0006
- Sampling Duration Initial Value Setup: Sets CCNT_PPC_A/M (Offset = 0xE100, 0xF100) to some value. If 10000 (0x2710) cycles simulation run, then 0xFFFF_D8EF (0xFFFF_FFFF – 0x2710) should be set
- Start All Counters: Sets PMNC_PPC_A/M (Offset = 0xE000, 0xF000) to 0x0000_0001

Step3: Running & Interrupt

- Generates interrupt from PPC_A due to overflow on CCNT_PPC_A

Step4: Stop State

- Stop All Counters: Sets PMNC_PPC_A/M (Offset = 0xE000, 0xF000) to 0x0000_0000

Step5: Read State

- Get Overflow Flag Register: Reads FLAG_PPC_A (Offset = 0xE050) value to verify if CCNT_PPC_A has generated interrupt.
- Get Performance Counter Value: Reads PMCNT1_PPC_A (Offset = 0xE120) for total Read request number and Read PMCNT1_PPC_M (Offset = 0xF120) for Read miss count number. Then PL_read_hit is calculated by the control function in *Table 18-4* describes enable/disable and start/stop control.

- Table 18-4.

Table 18-4 describes enable/disable and start/stop control.

Table 18-4 Enable/Disable and Start/Stop Control

Control Function		Description
Enable/Disable[counter/adder]		CNTENS/CNTENC
Enable/Disable[counter/adder]'s interrupt		INTENS/INTENC
Start/Stop PPC	Start_mode == 0	Start and Stop by Register: PMNC[0] == 1 → Start, PMNC[0] == 0 → Stop
	Start_mode == 1	Start and Stop by External Trigger Trigger == 1 → Start (In this case, PMNC[0] read value becomes 1) Trigger == 0 → Stop (In this case, PMNC[0] read value becomes 0)

CNTENC and CNTENS

PPMU_CTENC and PPMU_CTENS is a pair of related registers.

If you want to enable a counter, write 1 to its corresponding bit of PPMU_CTENS. When you do this, the values you read from PPMU_CTENC and PPMU_CTENS changes, which show the enabled counter's corresponding bit with 1.

If you want to disable a counter, write 1 to its corresponding bit of PPMU_CTENC. When you do this, the values from PPMU_CTENC and PPMU_CTENS changes, which show the corresponding bit of the disabled counter with 0.

The initial values of these registers are 0x0000_0000.

INTENC and INTENS

PPMU_INTENS and PPMU_INTENC are used either enable or disable the interrupt generation of the counters. Their setting rules similar to the previous registers.

CCNT and PMCNTx

PPMU_CCNT is a Read/Write register.

Before counting set initial value of PPMU_CCNT by writing some value to it. Read after this Write should be the same value as the written value.

After counting (suppose if you enable the CCNT counter), the PPMU_CCNT increases every cycle from its initial value until you stop counting. Any value read or written from/to the counter registers during the counting is not significant.

18.11 I/O Description

Signal	I/O	Description	Pin Name	Type
SCLK	O	Memory Clock	Xm1CLK, m2CLK	Dedicated
nSCLK	O	Memory Negative Clock	Xm1CLKn, Xm2CLKn	Dedicated
RASn	O	Row Address Selection	Xm1RASn, Xm2RASn	Dedicated
CASn	O	Column Address Selection	Xm1CASn, Xm2CASn	Dedicated
WEn	O	Write Enable	Xm1WEn, Xm2WEn	Dedicated
DATA[31:0]	I/O	Memory Data Bus	Xm1DATA_0 to Xm1DATA_31, Xm2DATA_0 to Xm2DATA_31	Dedicated
DQM[3:0]	O	Write Masking Per Byte	Xm1DQM_0 to Xm1DQM_3, Xm2DQM_0 to Xm2DQM_3	Dedicated
DQSp[3:0]	I/O	Data Strobe Signal Per Byte	Xm1DQS_0 to Xm1DQS_3, Xm2DQS_0 to Xm2DQS_3	Dedicated
DQSn[3:0]	I/O	Data Strobe Negative Signal Per Byte	Xm1DQSn_0 to Xm1DQSn_3, Xm2DQSn_0 to Xm2DQSn_3	Dedicated
ADCT[24:0], CKE	O	Memory Address, Bank Address, CS, CKE signals	Refer to *Table 18-5* for details	Dedicated
ODT	I	On Die Termination	Xm1ODT, Xm2ODT	Dedicated
ZQ	I	Reference Pin for Output Drive Strength Calibration	Xm1ZQ, Xm2ZQ	Dedicated

Table 18-5 lists the PAD Mux for address configuration.

Table 18-5 PAD Mux for Address Configuration

Signal	DDR3	LPDDR2	PIN Name
ADCT[0]	ADDR_0	CA_9	Xm1ADDR_0, Xm2ADDR_0
ADCT[1]	ADDR_1	CA_8	Xm1ADDR_1, Xm2ADDR_1
ADCT[2]	ADDR_2	CA_7	Xm1ADDR_2, Xm2ADDR_2
ADCT[3]	ADDR_3	CA_6	Xm1ADDR_3, Xm2ADDR_3
ADCT[4]	ADDR_4	CA_5	Xm1ADDR_4, Xm2ADDR_4
ADCT[5]	ADDR_5	CA_4	Xm1ADDR_5, Xm2ADDR_5
ADCT[6]	ADDR_6	CA_3	Xm1ADDR_6, Xm2ADDR_6
ADCT[7]	ADDR_7	CA_2	Xm1ADDR_7, Xm2ADDR_7
ADCT[8]	ADDR_8	CA_1	Xm1ADDR_8, Xm2ADDR_8
ADCT[9]	ADDR_9	CA_0	Xm1ADDR_9, Xm2ADDR_9
ADCT[10]	ADDR_10	NC	Xm1ADDR_10, Xm2ADDR_10
ADCT[11]	ADDR_11	NC	Xm1ADDR_11, Xm2ADDR_11
ADCT[12]	ADDR_12	NC	Xm1ADDR_12, Xm2ADDR_12
ADCT[13]	ADDR_13	NC	Xm1ADDR_13, Xm2ADDR_13
ADCT[14]	ADDR_14	NC	Xm1ADDR_14, Xm2ADDR_14
ADCT[15]	ADDR_15	NC	Xm1ADDR_15, Xm2ADDR_15
ADCT[16]	BA_0	NC	Xm1/2BA[0]
ADCT[17]	BA_1	NC	Xm1/2BA[1]
ADCT[18]	BA_2	NC	Xm1/2BA[2]
ADCT[19]	CKE_1	CKE_1	Xm1CKE_1, Xm2CKE_1
ADCT[20]	ODT_0	ODT_0	Xm1ODT_0, Xm2ODT_0
ADCT[21]	ODT_1	ODT_1	Xm1ODT_1, Xm2ODT_1
ADCT[22]		–	–
ADCT[23]	CS_0	CS_0	Xm1CSn_0, Xm2CSn_0
ADCT[24]	CS_1	CS_1	Xm1CSn_1, Xm2CSn_1
CKE	CKE_0	CKE_0	Xm1CKE_0, Xm2CKE_0

18.12 Register Description

18.12.1 Register Map Summary

- Base Address: 0x1060_0000
- Base Address: 0x1061_0000

Register	Offset	Description	Reset Value
CONCONTROL	0x0000	Controller control register	0x0FFF_1308
MEMCONTROL	0x0004	Memory control register	0x0020_2400
MEMCONFIG0	0x0008	Memory chip0 configuration register	0x20F8_1312
MEMCONFIG1	0x000C	Memory chip1 configuration register	0x28F8_1312
DIRECTCMD	0x0010	Memory direct command register	0x0000_0000
PRECHCONFIG	0x0014	Precharge policy configuration register	0xFF00_0000
PHYCONTROL0	0x0018	PHY control0 register	0x0010_1000
PHYCONTROL1	0x001C	PHY control1 register	0x0000_0000
PHYCONTROL2	0x0020	PHY control2 register	0x0000_0000
PHYCONTROL3	0x0024	PHY control3 register	0x0000_0000
PWRDNCONFIG	0x0028	Dynamic power down configuration register	0xFFFF_00FF
TIMINGAREF	0x0030	AC Timing register for SDRAM auto refresh	0x0000_005D
TIMINGROW	0x0034	AC Timing register for SDRAM row	0x0F23_3286
TIMINGDATA	0x0038	AC Timing register for SDRAM data	0x1215_0405
TIMINGPOWER	0x003C	AC Timing register for power mode of SDRAM	0x381B_0422
PHYSTATUS	0x0040	PHY status register	0x0000_0000
PHYZQCONTROL	0x0044	PHY ZQ I/O control register	0xE385_5731
CHIP0STATUS	0x0048	Memory chip0 status register	0x0000_0000
CHIP1STATUS	0x004C	Memory chip1 status register	0x0000_0000
AREFSTATUS	0x0050	Counter status register for auto refresh	0x0000_0000
MRSTATUS	0x0054	Memory mode registers status register	0x0000_0000
PHYTEST0	0x0058	PHY test register 0	0x0000_0000
PHYTEST1	0x005C	PHY test register 1	0x0000_0000
QOSCONTROL0	0x0060	Quality of service control register 0	0x0000_0000
QOSCONFIG0	0x0064	Quality of service configuration register 0	0x0000_0000
QOSCONTROL1	0x0068	Quality service control register 1	0x0000_0000
QOSCONFIG1	0x006C	Quality of service configuration register 1	0x0000_0000
QOSCONTROL2	0x0070	Quality of service control register 2	0x0000_0000
QOSCONFIG2	0x0074	Quality of service configuration register 2	0x0000_0000
QOSCONTROL3	0x0078	Quality service control register 3	0x0000_0000
QOSCONFIG3	0x007C	Quality of service configuration register 3	0x0000_0000
QOSCONTROL4	0x0080	Quality of service control register 4	0x0000_0000

Register	Offset	Description	Reset Value
QOSCONFIG4	0x0084	Quality of service configuration register 4	0x0000_0000
QOSCONTROL5	0x0088	Quality of service control register 5	0x0000_0000
QOSCONFIG5	0x008C	Quality of service configuration register 5	0x0000_0000
QOSCONTROL6	0x0090	Quality of service control register 6	0x0000_0000
QOSCONFIG6	0x0094	Quality of service configuration register 6	0x0000_0000
QOSCONTROL7	0x0098	Quality of service control register 7	0x0000_0000
QOSCONFIG7	0x009C	Quality of service configuration register 7	0x0000_0000
QOSCONTROL8	0x00A0	Quality of service control register 8	0x0000_0000
QOSCONFIG8	0x00A4	Quality of service configuration register 8	0x0000_0000
QOSCONTROL9	0x00A8	Quality of service control register 9	0x0000_0000
QOSCONFIG9	0x00AC	Quality of service configuration register 9	0x0000_0000
QOSCONTROL10	0x00B0	Quality of service control register 10	0x0000_0000
QOSCONFIG10	0x00B4	Quality of service configuration register 10	0x0000_0000
QOSCONTROL11	0x00B8	Quality of service control register 11	0x0000_0000
QOSCONFIG11	0x00BC	Quality of service configuration register 11	0x0000_0000
QOSCONTROL12	0x00C0	Quality of service control register 12	0x0000_0000
QOSCONFIG12	0x00C4	Quality of service configuration register 12	0x0000_0000
QOSCONTROL13	0x00C8	Quality of service control register 13	0x0000_0000
QOSCONFIG13	0x00CC	Quality of service configuration register 13	0x0000_0000
QOSCONTROL14	0x00D0	Quality of service control register 14	0x0000_0000
QOSCONFIG14	0x00D4	Quality of service configuration register 14	0x0000_0000
QOSCONTROL15	0x00D8	Quality of service control register 15	0x0000_0000
QOSCONFIG15	0x00DC	Quality of service configuration register 15	0x0000_0000
QOSTIMEOUT0	0x00E0	Quality of service timeout register 0	0x0FFF_0FFF
QOSTIMEOUT1	0x00E4	Quality of service timeout register 1	0x0FFF_0FFF
IVCONTROL	0x00F0	Interleaving control register	0x8000_000C
PEREVCONFIG	0x00FC	Performance events configuration register	0x0000_0000
PMNC_PPC_A	0xE000	Performance monitor control register	0x0000_0000
CNTENS_PPC_A	0xE010	Count enable set register	0x0000_0000
CNTENC_PPC_A	0xE020	Count enable clear register	0x0000_0000
INTENS_PPC_A	0xE030	Interrupt enable set register	0x0000_0000
INTENC_PPC_A	0xE040	Interrupt enable clear register	0x0000_0000
FLAG_PPC_A	0xE050	Overflow flag status register	0x0000_0000
CCNT_PPC_A	0xE100	Cycle count register	0x0000_0000
PMCNT0_PPC_A	0xE110	Performance monitor count register	0x0000_0000
PMCNT1_PPC_A	0xE120	Performance monitor count register	0x0000_0000
PMCNT2_PPC_A	0xE130	Performance monitor count register	0x0000_0000

Register	Offset	Description	Reset Value
PMCNT3_PPC_A	0xE140	Performance monitor count register	0x0000_0000
PMNC_PPC_M	0xF000	Performance monitor control register	0x0000_0000
CNTENS_PPC_M	0xF010	Count enable set register	0x0000_0000
CNTENC_PPC_M	0xF020	Count enable clear register	0x0000_0000
INTENS_PPC_M	0xF030	Interrupt enable set register	0x0000_0000
INTENC_PPC_M	0xF040	Interrupt enable clear register	0x0000_0000
FLAG_PPC_M	0xF050	Overflow flag status register	0x0000_0000
CCNT_PPC_M	0xF100	Cycle count register	0x0000_0000
PMCNT0_PPC_M	0xF110	Performance monitor count register	0x0000_0000
PMCNT1_PPC_M	0xF120	Performance monitor count register	0x0000_0000
PMCNT2_PPC_M	0xF130	Performance monitor count register	0x0000_0000
PMCNT3_PPC_M	0xF140	Performance monitor count register	0x0000_0000

18.12.1.1 CONCONTROL

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0000, Reset Value = 0x0FFF_1308

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved (Should be zero.)	0x0
timeout_level0	[27:16]	RW	QoS Timeout Count 0 0xn = n aclk cycles (aclk: AXI clock) This counter prevents transactions in the command request buffer from starvation. This counter starts when a new AXI transaction comes into the request buffer. When the counter becomes zero, the corresponding transaction becomes the highest priority command of all the transactions in the command request buffer. This is used as a default timeout counter when AxQOS is 0. QoS counter can override if the ARID or AWID matches the QoS ID when it enters into the command queue. Refer to Chapter *18.7 Quality of Service* for more information.	0xFFF
rd_fetch	[15:12]	RW	Read Data Fetch Cycles 0xn = n mclk cycles (mclk: Memory clock) This register is for the unpredictable latency of Read data that enters from memory devices by tDQSCK variation or the board flying time. This parameter must control the Read fetch delay of PHY Read FIFO. The controller fetches Read data from PHY after read_latency + n mclk cycles. Refer to Chapter *18.8 Read Data Capture* for more information. In DDR3, If PhyControl1.term_read_en is enabled, add 1 cycle more to this value.	0x1
qos_fast_en	[11]	RW	Adaptive QoS Enable 0x0 = Disables 0x1 = Enables When you enable this, the controller loads QoS counter value from QoSControl.qos_cnt_f instead of QoSControl.qos_cnt when the corresponding input pin qos_fast is turned on. Refer to Chapter *18.7 Quality of Service* for more information.	0x0
dq_swap	[10]	RW	DQ Swap 0x0 = Disables 0x1 = Enables When you enable this, the controller reverses the bit order of memory data pins. (For example, DQ[31] <-> DQ[0], DQ[30]	0x0

Name	Bit	Type	Description	Reset Value
			<-> DQ[1])	
chip1_empty	[9]	R	Command Queue Status of Chip1 0x0 = Not Empty 0x1 = Empty There is no AXI transaction corresponding to chip1 memory in the command queue entries.	0x1
chip0_empty	[8]	R	Command Queue Status of Chip0 0x0 = Not Empty 0x1 = Empty There is no AXI transaction corresponding to chip0 memory in the command queue entries.	0x1
drv_type	[7:6]	RW	PHY Driving type 0x0 = Disables 0x1 = Reserved 0x2 = Dynamic pull down 0x3 = Static pull down During the high-Z state of the memory bidirectional pins, PHY drives these pins with the zeros, or pulls down these pins statically or dynamically to prevent current leakage. Set this value to 0x3 in LPDDR2-S4 over 333 MHz.	0x0
aref_en	[5]	RW	Auto Refresh Counter 0x0 = Disables 0x1 = Enables Enables this to decrease the auto refresh counter by 1 at the rising edge of the mclk.	0x0
pdn_dq_ disable	[4]	RW	DQ pulldown Disable Control 0x0 = Controls pulldown of dq according to drv_type of ConControl register 0x1 = Disables pulldown of dq When this value is set to 0x1, DMC disables pulldown of dq regardless of the drv_type field of ConControl register field. When this value is set to 0x0, the drv_type field of ConControl register defines control of dq pulldown.	0x0
io_pd_con	[3]	RW	I/O Powerdown Control 0x0 = Uses programmed ctrl_pd and pulldown control 0x1 = Controls ctrl_pd and pulldown control automatically When this value is set to 0x1, DMC automatically sets powerdown enable for input buffer of I/O and pulls down disable for dq and dqs in power-down mode. When this value is set to 0x0, DMC only sends programmed ctrl_pd value and pulldown control.	0x1
clk_ratio	[2:1]	RW	Clock Ratio of Bus Clock to Memory Clock 0x0 = freq.(aclk): freq.(mclk) = 1:1 0x1 = freq.(aclk): freq.(mclk) = 1:2 0x2 to 0x3 = Reserved	0x0

Name	Bit	Type	Description	Reset Value
RSVD	[0]	RW	Reserved. (Should be zero)	0x0

18.12.1.2 MEMCONTROL

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0004, Reset Value = 0x0020_2400

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved (Should be zero)	0x0
mrr_byte	[26:25]	RW	Mode Register Read byte lane location 0x0 = Memory dq[7:0] 0x1 = Memory dq[15:8] 0x2 = Memory dq[23:16] 0x3 = Memory dq[31:24]	0x0
pzq_en	[24]	RW	DDR3 periodic ZQ(ZQCS) enable Note that after exit from self refresh, ZQ function is required by the software. The controller does not support ZQ calibration command to be issued in parallel to DLL lock time when coming out of self refresh. Turn-on only when using DDR3.	0x0
otf_en	[23]	RW	DDR3 bc4/bl8 on the fly enable Turn-on only when using DDR3	0x0
bl	[22:20]	RW	Memory Burst Length 0x0 = Reserved 0x1 = 2 0x2 = 4 0x3 = 8 0x4 = 16 0x5 to 0x7 = Reserved In case of LPDDR2-S4, the controller only supports burst length 4. In case of DDR3, the controller only supports burst length 8.	0x2
num_chip	[19:16]	RW	Number of Memory Chips 0x0 = 1 chip 0x1 = 2 chips 0x2 to 0xf = Reserved	0x0
mem_width	[15:12]	RW	Width of Memory Data Bus 0x0 = Reserved 0x1 = 16-bit 0x2 = 32-bit 0x3 = Reserved 0x4 to 0xf = Reserved	0x2
mem_type	[11:8]	RW	Type of Memory 0x0 to 0x4 = Reserved 0x5 = LPDDR2-S4 0x6 = DDR3 0x7 to 0xf = Reserved	0x4
add_lat_pall	[7:6]	RW	Additional Latency for PALL 0x0 = 0 cycle	0x0

Name	Bit	Type	Description	Reset Value
			0x1 = 1 cycle 0x2 = 2 cycle 0x3 = Reserved When the controller issues all banks precharge command, the latency of precharging will be tRP + add_lat_pall	
dsref_en	[5]	RW	Dynamic Self Refresh 0x0 = Disables 0x1 = Enables Refer to Chapter 18.5.2 Dynamic Power Down for more information. In DDR3, this feature is not supported. This feature must be turn-off when using DDR3.	0x0
tp_en	[4]	RW	Timeout Precharge 0x0 = Disables timeout precharge 0x1 = Enables timeout precharge When you enable tp_en, it automatically precharges an open bank after a specified amount of mclk cycles (if no access has been made in between the cycles) in an open page policy. If PrechConfig.tp_cnt bit-field is set, it specifies the amount of mclk cycles to wait until timeout precharge precharges the open bank. Refer to Chapter 18.6.2 Timeout Precharge for more information.	0x0
dpwrdn_type	[3:2]	RW	Type of Dynamic Power Down 0x0 = Active/precharge power down 0x1 = Forced precharge power down 0x2 to 0x3 = Reserved Refer to Chapter 18.5.2 Dynamic Power Down for more information.	0x0
dpwrdn_en	[1]	RW	Dynamic Power Down 0x0 = Disables dynamic power down 0x1 = Enables dynamic power	0x0
clk_stop_en	[0]	RW	Dynamic Clock Control 0x0 = Always runs 0x1 = Stops during idle periods Refer to Chapter 18.5.4 Clock Stop for more information.	0x0

18.12.1.3 MEMORYCHIP0

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0008, Reset Value = 0x20F8_1312

Name	Bit	Type	Description	Reset Value
chip_base	[31:24]	RW	AXI Base Address AXI base address[31:24] = chip_base, For example, if chip_base = 0x20, then AXI base address of memory chip0 becomes 0x2000_0000.	0x20
chip_mask	[23:16]	RW	AXI Base Address Mask Upper address bit masks to determine AXI offset address of memory chip0. 0 = Does not use corresponding address bit for comparison 1 = Uses corresponding address bit for comparison For example, if chip_mask = 0xF8, then AXI offset address becomes 0x0000_0000 to 0x07FF_FFFF. If AXI base address of memory chip0 is 0x2000_0000, then memory chip0 has an address range of 0x2000_0000 to 0x27FF_FFFF.	0xF8
chip_map	[15:12]	RW	Address Mapping Method (AXI to Memory) 0x0 = Linear (bank, row, column, width) 0x1 = Interleaved (row, bank, column, width) 0x2 to 0xf = Reserved	0x1
chip_col	[11:8]	RW	Number of Column Address Bits 0x0 = 7 bits 0x1 = Reserved 0x2 = 9 bits 0x3 = 10 bits 0x4 = Reserved 0x5 to 0xf = Reserved	0x3
chip_row	[7:4]	RW	Number of Row Address Bits 0x0 = 12 bits 0x1 = 13 bits 0x2 = 14 bits 0x3 = 15 bits 0x4 = 16 bits 0x5 to 0xf = Reserved	0x1
chip_bank	[3:0]	RW	Number of Banks 0x0 = Reserved 0x1 = 2 banks 0x2 = 4 banks 0x3 = 8 banks 0x4 to 0xf = Reserved	0x2

18.12.1.4 MEMCONFIG1

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x000C, Reset Value = 0x28F8_1312

Name	Bit	Type	Description	Reset Value
chip_base	[31:24]	RW	AXI Base Address AXI base address[31:24] = chip_base, For example, if chip_base = 0x28, then AXI base address of chip1 becomes 0x2800_0000.	0x28
chip_mask	[23:16]	RW	AXI Base Address Mask Upper address bit masks to determine AXI offset address of memory chip1. 0 = Does not use corresponding address bit for comparison 1 = Uses corresponding address bit for comparison For example, if chip_mask = 0xF0, then AXI offset address becomes 0x0000_0000 to 0x0FFF_FFFF. If AXI base address of memory chip1 is 0x2800_0000, then memory chip1 has an address range of 0x2800_0000 to 0x37FF_FFFF.	0xF8
chip_map	[15:12]	RW	Address Mapping Method (AXI to Memory) 0x0 = Linear ({bank, row, column, width}) 0x1 = Interleaved ({row, bank, column, width}) 0x2 to 0xf = Reserved	0x1
chip_col	[11:8]	RW	Number of Column Address Bits 0x0 = 7 bits 0x1 = Reserved 0x2 = 9 bits 0x3 = 10 bits 0x4 = Reserved 0x5 to 0xf = Reserved	0x3
chip_row	[7:4]	RW	Number of Row Address Bits 0x0 = 12 bits 0x1 = 13 bits 0x2 = 14 bits 0x3 = 15 bits 0x4 = 16 bits 0x5 to 0xf = Reserved	0x1
chip_bank	[3:0]	RW	Number of Banks 0x0 = Reserved 0x1 = 2 banks 0x2 = 4 banks 0x3 = 8 banks 0x4 to 0xf = Reserved	0x2

18.12.1.5 DIRECTCMD

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved (Should be zero)	0x0
cmd_type	[27:24]	RW	Type of Direct Command 0x0 = MRS/EMRS (mode register setting) 0x1 = PALL (all banks precharge) 0x2 = PRE (per bank precharge) 0x3 = DPD (deep power down) 0x4 = REFS (self refresh) 0x5 = REFA (auto refresh) 0x6 = CKEL (active/precharge power down) 0x7 = NOP (exit from active/precharge power down or deep power down) 0x8 = REFSX (exit from self refresh) 0x9 = MRR (mode register reading) 0xa = ZQINIT (ZQ calibration init.) 0xb = ZQOPER (ZQ calibration long) 0xc = ZQCS (ZQ calibration short) 0xd to 0xf = Reserved When the controller issues a direct command, AXI masters must not access memory. You must verify the command queue's state by ConControl.chip0/1_empty and the bank FSM in the Chip0/1Status register before issuing a direct command. You must verify the bank FSM status before issuing a direct command. You must disable clk_stop_en, dynamic power down, dynamic self refresh, and force precharge function (MemControl register). You should issue MRS/EMRS or MRR commands if all banks are in idle state. If you issue MRS/EMRS or MRR commands to LPDDR2-S4, map the CA pins as: MA[7:0] = {cmd_addr[1:0], cmd_bank[2:0], cmd_addr[12:10]} OP[7:0] = cmd_addr[9:2] In DDR3, self refresh related timing such as tCKESR/tCKSRE/tCKSRX should be check by software.	0x0
RSVD	[23:21]	–	Reserved (Should be zero)	0x0
cmd_chip	[20]	RW	Select the chip number to which it sends the direct command. 0 = Chip 0 1 = Chip 1	0x0
RSVD	[19]	–	Reserved (Should be zero)	0x0
cmd_bank	[18:16]	RW	Related Bank Address that issues a direct command To send a direct command to a chip, it requires additional	0x0

Name	Bit	Type	Description	Reset Value
			information like blank address. It uses this register in such situation.	
cmd_addr	[15:0]	RW	Related Address value that uses a direct command To send a direct command to a chip, it requires additional information like blank address. It uses this register in such situation.	0x0

18.12.1.6 PRECHCONFIG

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0014, Reset Value = 0xFF00_0000

Name	Bit	Type	Description	Reset Value
tp_cnt	[31:24]	RW	Timeout Precharge Cycles 0xn = n mclk cycles When you enable the timeout precharge function (MemControl.tp_en) and the timeout precharge counter becomes zero, the controller forces the activated memory bank into the precharged state. Refer to Chapter *18.6.2 Timeout Precharge* for more information.	0xFF
RSVD	[23:16]	–	Reserved (Should be zero)	0x0
chip1_policy	[15:8]	RW	Memory Chip1 Precharge Bank Selective Policy 0x0 = Open page policy 0x1 = Close page (auto precharge) policy chip1_policy[n], where n is the bank number of chip1. Open Page Policy: After a READ or WRITE command, the row that the controller accesses is left open. Close Page (Auto Precharge) Policy: Right after a READ or WRITE command, memory devices automatically precharges the bank. This is a bank selective precharge policy. For example, when chip1_policy[2] is 0x0, bank2 of chip1 has an open page policy and if chip1_policy[6] is 0x1, bank6 of chip1 has a close page policy. Refer to Chapter *18.6.1 Bank Selective Precharge Policy* for more information.	0x0
chip0_policy	[7:0]	RW	Memory Chip0 Precharge Bank Selective Policy 0x0 = Open page policy 0x1 = Close page (auto precharge) policy Chip0_policy[n], where n is the bank number of chip0. This is for memory chip0.	0x0

18.12.1.7 PHYCONTROL0

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0018, Reset Value = 0x0010_1000

Name	Bit	Type	Description	Reset Value
ctrl_force	[31:24]	RW	DLL Force Delay Uses this field instead of PhyStatus.ctrl_lock_value[9:2] from the DLL only when PhyControl0.ctrl_dll_on is LOW. (That is, when the DLL is off, it uses this field to generate 270' clock and shift DQS by 90').	0x0
ctrl_inc	[23:16]	RW	DLL Delay Increment Increases the amount of start point This value should be 0x10	0x10
ctrl_start_point	[15:8]	RW	DLL Lock Start Point Initial DLL lock start point. This is the number of delay cells and is the start point where "DLL" starts tracing to lock. Calculates initial delay time by multiplying the unit delay of delay cell and this value. This value should be 0x10	0x10
dqs_delay	[7:4]	RW	Delay Cycles for DQS Cleaning This register enables PHY to clean incoming DQS signals. External circumstances delays DQS signals. If DQS is coming back with Read latency plus n mclk cycles, you should set this register value to have n mclk cycles.	0x0
ctrl_dfdqs	[3]	RW	Differential DQS If you enable this register, PHY generates differential DQS out signals for WRITE command and receives differential DQS input signals for READ command. In LPDDR2-S4 and DDR3, you must enable this field.	0x0
ctrl_half	[2]	RW	DLL Low Speed HIGH active signal turns on the low-speed mode for DLL. When this bit is set to HIGH, DLL can run at low speed (80 MHz to 166 MHz)	0x0
ctrl_dll_on	[1]	RW	DLL On HIGH active start signal turns on the DLL. This signal should be kept HIGH for normal operation. When this signal becomes LOW, it turns off DLL and ctrl_clock and ctrl_flock become HIGH. This bit should be set before ctrl_start turns on the DLL.	0x0
ctrl_start	[0]	RW	DLL Start HIGH active start signal makes DLL run and lock. This signal should be kept HIGH during normal operation. When this signal becomes LOW, DLL stops running. To re-run DLL, make this signal HIGH again. When you re-run, DLL loses previous lock information. Before ctrl_start is set, ensure that ctrl_dll_on is HIGH.	0x0

Figure 18-12 illustrates the PHY DLL lock procedure.

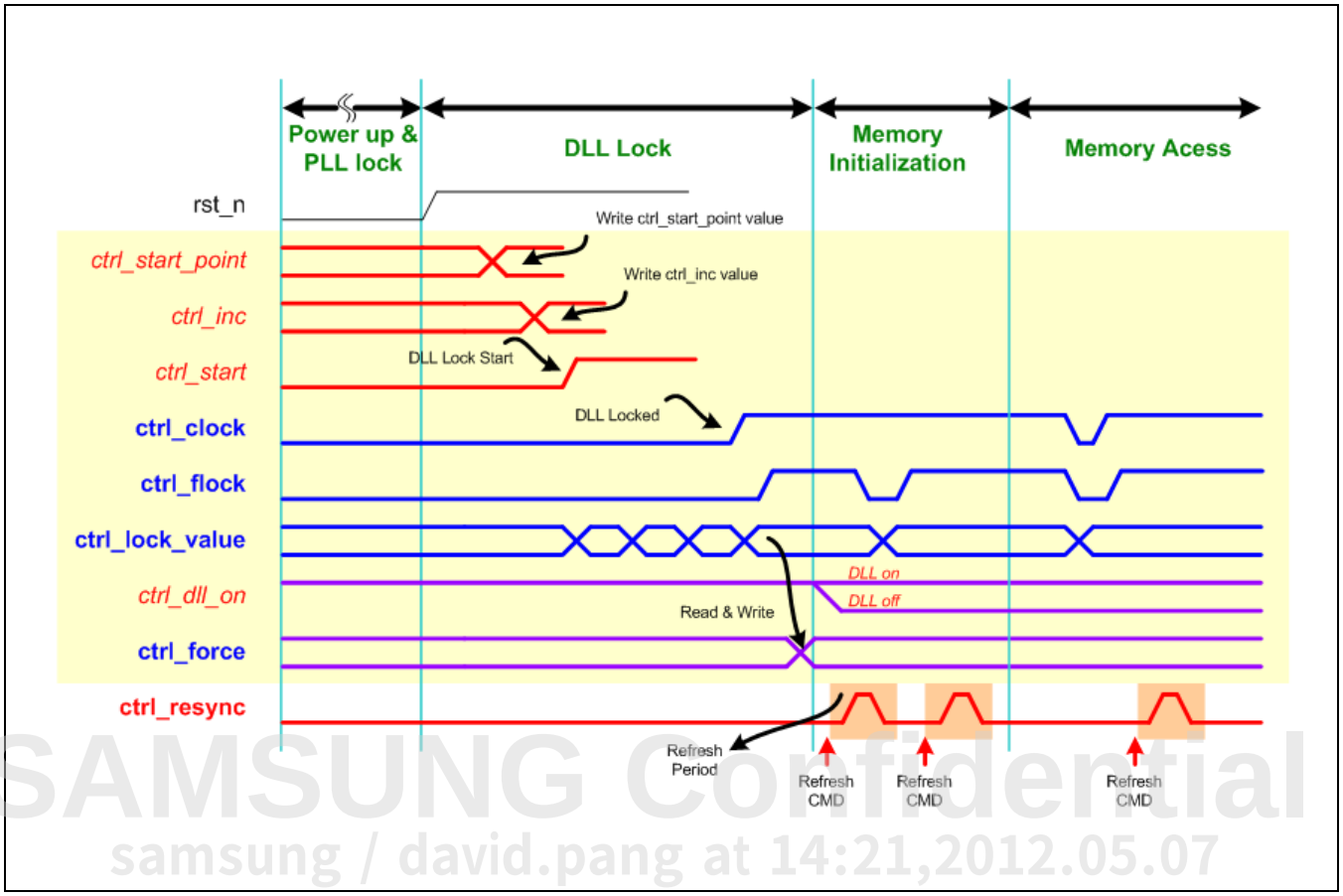


Figure 18-12 PHY DLL Lock Procedure

Use DLL to compensate Process, Voltage, and Temperature (PVT) condition. Therefore, do not turn off for reliable operation except for frequency scaling (Turning off DLL only permits lowering of frequency scaling).

18.12.1.8 PHYCONTROL1

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
mem_term_en	[31]	RW	Termination Enable for Memory At high-speed memory operation, it is necessary to turn on termination resistance in memory devices. This register controls an On-Die Termination (ODT) pin of a memory device.	0x0
phy_read_en	[30]	RW	Termination Enable for PHY At high-speed memory operation, it is necessary to turn on termination resistance in the PHY. This register controls an ODT pin of memory device.	0x0
ctrl_shgate	[29]	RW	Duration of DQS Gating Signal This field controls the gate control signal In LPDDR2-S4, this field should be set to 1'b0 regardless of clock frequency. In other memory types, this field should be: 1'b0 = (Gate signal length = "burst length/2" (<= 200 MHz)) 1'b1 = (Gate signal length = "burst length/2" – 1 (> 200 MHz))	0x0
ctrl_pd	[28:24]	RW	Input Gate for Power Down When this field is set, input buffer is turned off for power down. This field should be set to 0 for normal operation. ctrl_pd[3:0] = For each data slice ctrl_pd[4] = For control slice	0x0
ctrl_cmosrcv	[23]	RW	I/O Type This field controls the input mode of I/O 1'b0 = Differential receiver mode for high-speed operation 1'b1 = CMOS receiver mode for low-speed operation (< 200 MHz)	0x0
ctrl_offsetd	[22:16]	RW	This field is for debug purpose. When you fix this field, do not change it during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. Offset amount for 270' clock generation: • ctrl_offsetd[6]: 1 = (tFS: fine step delay) 270' delay amount – ctrl_offsetd[5:0] × tFS • ctrl_offsetd[6]: 0 = 270' delay amount + ctrl_offsetd[5:0] × tFS	0x0
RSVD	[15]	–	Reserved (Should be zero)	0x0
ctrl_offsetc	[14:8]	RW	Delay Offset for DQS Cleaning Gate offset amount for DDR. When you fix this field, do not change it during operation. This value is valid only after ctrl_resync becomes HIGH and LOW.	0x0

Name	Bit	Type	Description	Reset Value
			• ctrl_offsetc[6]: 1 = (tFS: fine step delay) GATEout delay amount - ctrl_offsetc[5:0] × tFS • ctrl_offsetc[6]: 0 = GATEout delay amount + ctrl_offsetc[5:0] × tFS	
ctrl_ref	[7:4]	RW	Reference Count for DLL Lock Confirmation	0x0
fp_resync	[3]	RW	Force DLL Resyncronization	0x0
ctrl_shiftc	[2:0]	RW	Phase Delay for DQS Cleaning GATEout signal delays amount for DDR When you fix this field, do not change it during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. 0x000 = T/128 (2.8125' shift) 0x001 = T/64 (5.625' shift) 0x010 = T/32 (11.25' shift) 0x011 = T/16 (22.5' shift) 0x100 = T/8 (45' shift) 0x101 = T/4 (90' shift) 0x110 = T/2 (180' shift) 0x111 = T (360' shift) Recommended values according to memory type: 0x100 when LPDDR2-S4, 0x110 when DDR2/DDR3	0x0

NOTE: DQS cleaning scheme: Use DQS cleaning to remove high-Z state of DQS.

Figure 18-13 illustrates the board-level connection diagram for DQS cleaning.

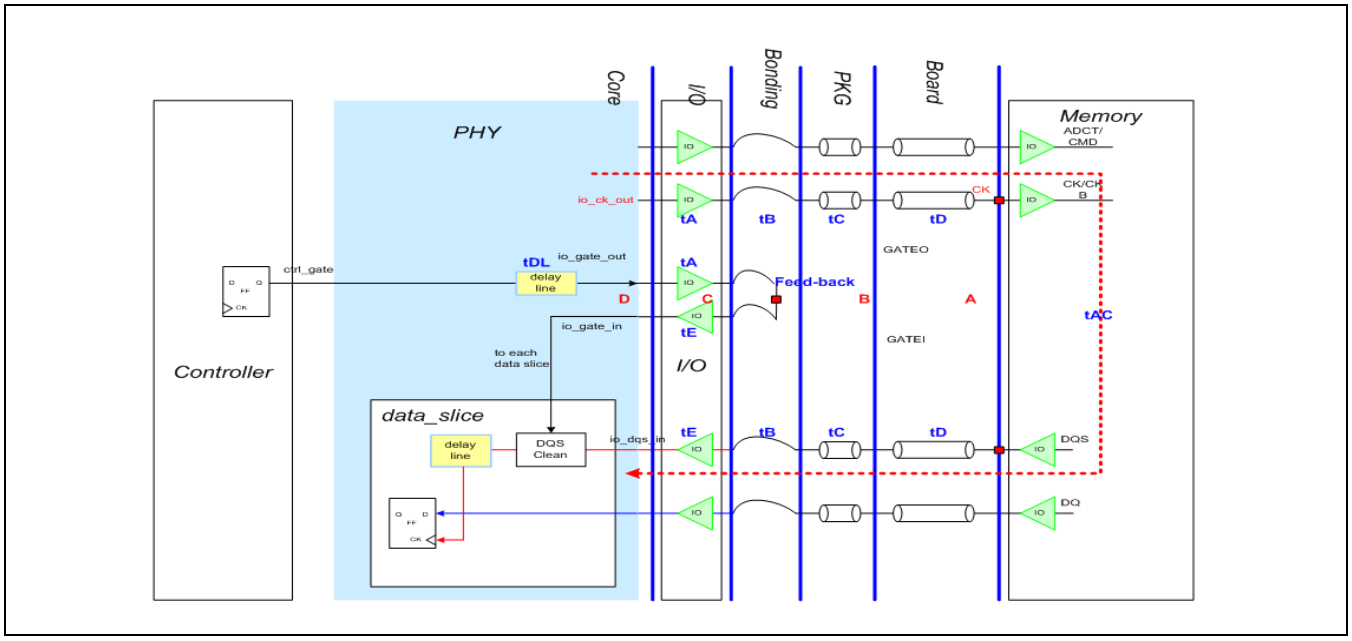


Figure 18-13 Board Level Connection Diagram for DQS Cleaning

DQS related timing parameters are:

- tA: I/O output delay
- tB: Package bonding wire delay
- tC: Package board delay
- tD: Board trace delay
- tE: I/O input delay
- tDL: Delay line delay
- tAC: Minimum CK-to-DQS timing.
- tFS: Fine step delay in DLL, From PhyStatus0.ctrl_lock_value[9:0], tFS is calculated.
  If ctrl_half = 0, tFS = tCK/ctrl_lock_value[9:0].
  If ctrl_half = 1, tFS = tCK × 0.5/ctrl_lock_value[9:0]

ctrl_shiftc controls PVT-independent delay amount (tF) and ctrl_offsetc controls PVT-dependent delay amount (tV).

Line delay calculation is:

- Delay line programming value; tDL ≈ tAC + 2 × (tB + tC + tD).
- tDL = tF (ctrl_shiftc[2:0]) + tV (ctrl_offsetc[6:0])
- If ctrl_shiftc[2:0] is 3'b100, tF is Tperiod/8 ≈ 0.9375 ns. (If tCK is 7.5 ns)
- If ctrl_offsetc[6:0] is 7'b00010_00, tV is 0.320 ns(40 ps × 8) @ worst case (if tFS = 40 ps)
- Therefore, tDL = tF + tV = 0.9375 ns + 0.320 ns = 1.2575 ns

Figure 18-14 illustrates the DQS cleaning for LPDDR2 if tAC Min.

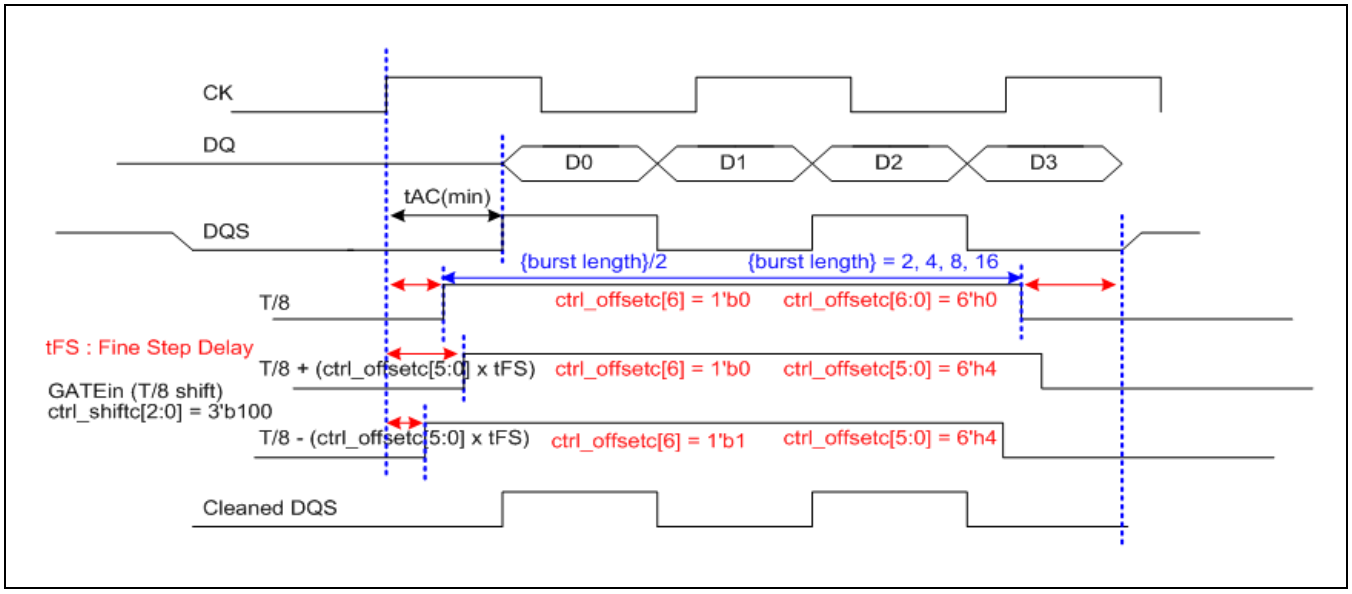


Figure 18-14 DQS Cleaning for LPDDR2 if tAC Min

Figure 18-15 illustrates the DQS cleaning for LPDDR2 if tAC Max.

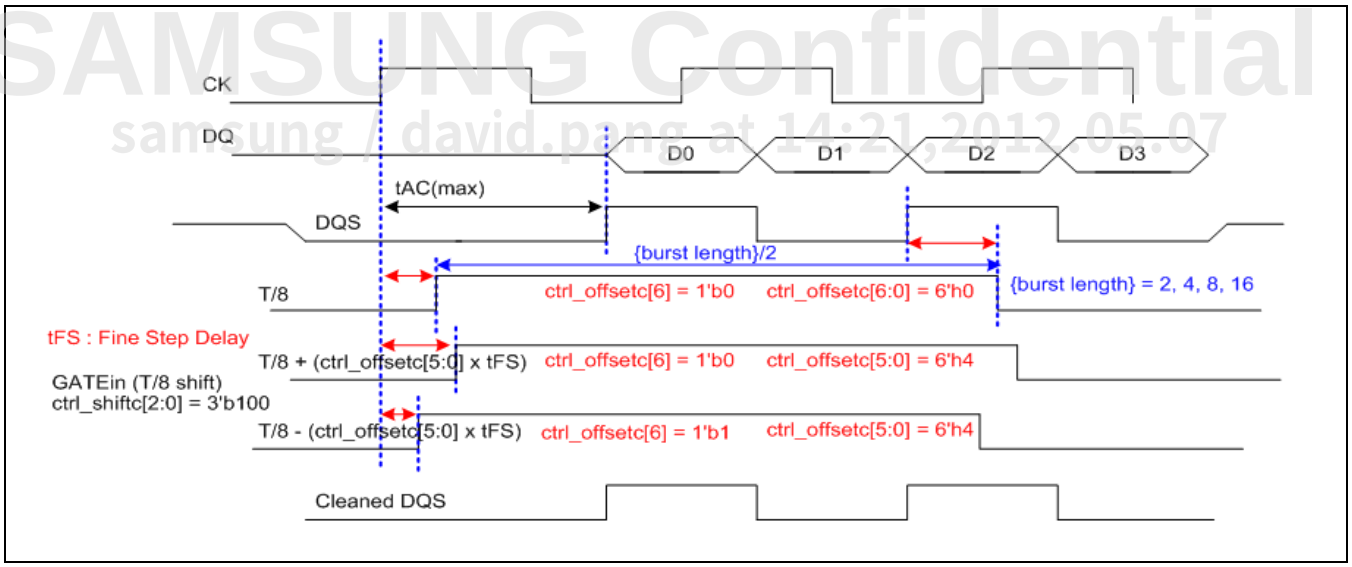


Figure 18-15 DQS Cleaning for LPDDR2 if tAC Max

Figure 18-15 illustrates the DQS cleaning for DDR3.

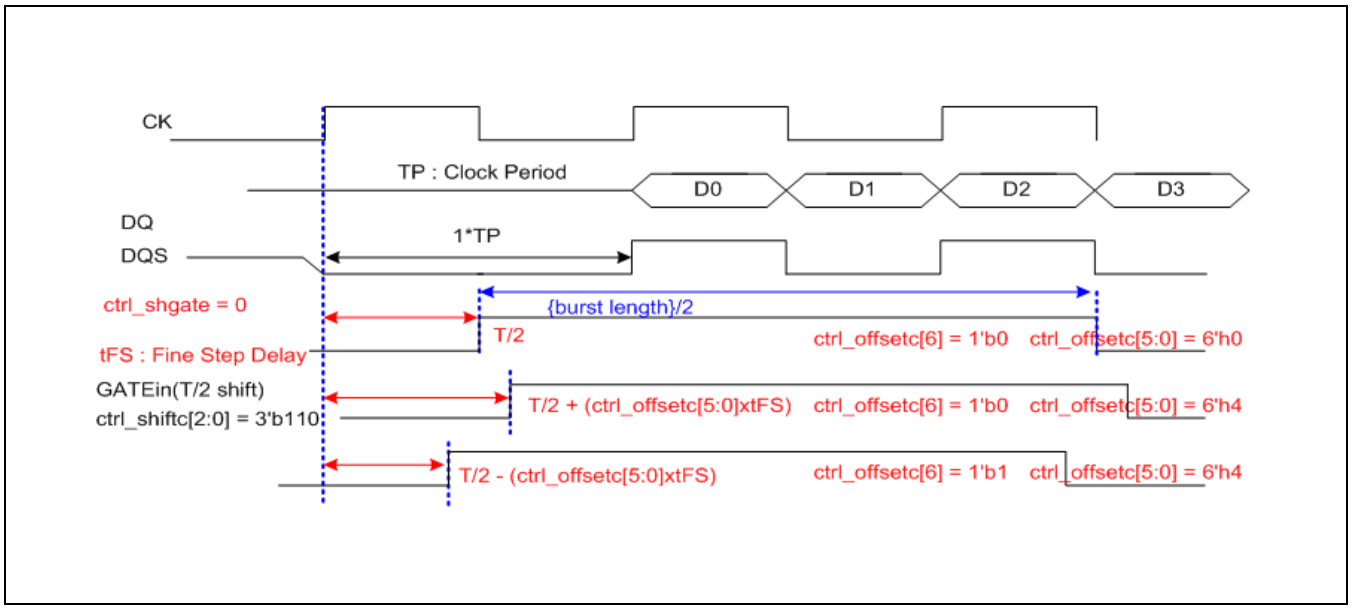


Figure 18-16 DQS cleaning for DDR3

18.12.1.9 PHYCONTROL2

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved (Should be zero)	0x0
ctrl_offsetr3	[30:24]	RW	Uses this field to give offset to Read DQS. When you fix this field, do not change it during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. Read DQS offset amount: • ctrl_offsetr3[6]: 1 = (tFS: fine step delay) Read DQS 90 delay amount – ctrl_offsetr3[5:0] × tFS • ctrl_offsetr3[6]: 0 = Read DQS 90 delay amount + ctrl_offsetr3[5:0] × tFS.	0x0
RSVD	[23]	–	Reserved (Should be zero)	0x0
ctrl_offsetr2	[22:16]	RW	Uses this field to give offset to Read DQS. When you fix this field, do not change it during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. Read DQS offset amount: • ctrl_offsetr2[6]: 1 = (tFS: fine step delay) Read DQS 90 delay amount – ctrl_offsetr2[5:0] × tFS • ctrl_offsetr2[6]: 0 = Read DQS 90 delay amount + ctrl_offsetr2[5:0] × tFS.	0x0
RSVD	[15]	–	Reserved (Should be zero)	0x0
ctrl_offsetr1	[14:8]	RW	Uses this field to give offset to Read DQS. When you fix this field, do not change it during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. Read DQS offset amount: • ctrl_offsetr1[6]: 1 = (tFS: fine step delay) Read DQS 90 delay amount – ctrl_offsetr1[5:0] × tFS • ctrl_offsetr1[6]: 0 = Read DQS 90 delay amount + ctrl_offsetr1[5:0] × tFS.	0x0
RSVD	[7]	–	Reserved (Should be zero)	0x0
ctrl_offsetr0	[6:0]	RW	Uses this field to give offset to Read DQS. When you fix this field, do not change it during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. Read DQS offset amount: • ctrl_offsetr0[6]: 1 = (tFS: fine step delay) Read DQS 90 delay amount – ctrl_offsetr0[5:0] × tFS • ctrl_offsetr0[6]: 0 = Read DQS 90 delay amount + ctrl_offsetr0[5:0] × tFS.	0x0

18.12.1.10 PHYCONTROL3

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved (Should be zero)	0x0
ctrl_offsetw3	[30:24]	RW	Uses this field to give offset to Write DQS. When you fix this field, do not change it during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. Write DQS offset amount: • ctrl_offsetw3[6]: 1 = (tFS: fine step delay) Write DQS 90 delay amount – ctrl_offsetw3[5:0] × tFS • ctrl_offsetw3[6]: 0 = Write DQS 90 delay amount + ctrl_offsetw3[5:0] × tFS.	0x0
RSVD	[23]	–	Reserved (Should be zero)	0x0
ctrl_offsetw2	[22:16]	RW	Uses this field to give offset to Write DQS. When you fix this field, do not change it during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. Write DQS offset amount: • ctrl_offsetw2[6]: 1 = (tFS: fine step delay) Write DQS 90 delay amount – ctrl_offsetw2[5:0] × tFS • ctrl_offsetw2[6]: 0 = Write DQS 90 delay amount + ctrl_offsetw2[5:0] × tFS.	0x0
RSVD	[15]	–	Reserved (Should be zero)	0x0
ctrl_offsetw1	[14:8]	RW	Uses this field to give offset to Write DQS. When you fix this field, do not change it during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. Write DQS offset amount: • ctrl_offsetw1[6]: 1 = (tFS: fine step delay) Write DQS 90 delay amount – ctrl_offsetw1[5:0] × tFS • ctrl_offsetw1[6]: 0 = Write DQS 90 delay amount + ctrl_offsetw1[5:0] × tFS.	0x0
RSVD	[7]	–	Reserved (Should be zero)	0x0
ctrl_offsetw0	[6:0]	RW	Uses this field to give offset to Write DQS. When you fix this field, do not change it during operation. This value is valid only after ctrl_resync becomes HIGH and LOW. Write DQS offset amount : • ctrl_offsetw0[6]: 1 = (tFS : fine step delay) Write DQS 90 delay amount – ctrl_offsetw0[5:0] × tFS • ctrl_offsetw0[6]: 0 = Write DQS 90 delay amount + ctrl_offsetw0[5:0] × tFS.	0x0

18.12.1.11 PWRDNCONFIG

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0028, Reset Value = 0xFFFF_00FF

Name	Bit	Type	Description	Reset Value
dsref_cyc	[31:16]	RW	Number of Cycles for dynamic self refresh entry 0xn = n aclk cycles If the command queue is empty for n + 1 cycles, the controller forces memory devices into self-refresh state. Refer to Chapter 18.5.3 Dynamic Self Refresh for more information.	0xFFFF
RSVD	[15:8]	–	Reserved (Should be zero)	0x0
dpwrdn_cyc	[7:0]	RW	Number of Cycles for dynamic power down entry 0xn = n aclk cycles When the command queue is empty for n + 1 cycles, the controller forces the memory device into active/precharge power down state. Refer to Chapter 18.5.2 Dynamic Power Down for more information.	0xFF

18.12.1.12 TIMINGPZQ

- Base Address = 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x002C, Reset Value = 0x0000_4084

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Should be zero	0x0
t_pzq	[23:0]	RW	Average Periodic ZQ Interval(Only in DDR3) tREFI(t_refi × T(rclk)) × t_pzq should be less than or equal to the minimum value of memory periodic ZQ interval, for example, if rclk frequency is 12MHz, t_refi is set to 93 and ZQ interval is 128ms then the following value should be programmed into it: 128 ms × 12 MHz / 93 = 16516	0x4084

18.12.1.13 TIMINGAREF

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_005D

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved (Should be zero)	0x0
t_refi	[15:0]	RW	Average Periodic Refresh Interval t_refi × T (rclk) should be less than or equal to the minimum value of memory tREFI (all bank). For example, for the all bank refresh period of 7.8 μs, and an rclk frequency of 12 MHz, the value that is given here should be programmed into it:	0x5D

			7.8 μs × 12 MHz = 93	

18.12.1.14 TIMINGROW

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0034, Reset Value = 0x0F23_3286

Name	Bit	Type	Description	Reset Value
t_rfc	[31:24]	RW	Auto refresh to Active/Auto refresh command period, in cycles t_rfc × T (mclk) should be greater than or equal to the minimum value of memory Trfc	0xF
t_rrd	[23:20]	RW	Active bank A to Active bank B delay, in cycles t_rrd × T (mclk) should be greater than or equal to the minimum value of memory tRRD	0x2
t_rp	[19:16]	RW	Precharge command period, in cycles t_rp × T (mclk) should be greater than or equal to the minimum value of memory tRP	0x3
t_rcd	[15:12]	RW	Active to Read or Write delay, in cycles t_rcd × T (mclk) should be greater than or equal to the minimum value of memory tRCD	0x3
t_rc	[11:6]	RW	Active to Active period, in cycles t_rc × T (mclk) should be greater than or equal to the minimum value of memory tRC	0xA
t_ras	[5:0]	RW	Active to Precharge command period, in cycles t_ras × T (mclk) should be greater than or equal to the minimum value of memory tRAS	0x6

18.12.1.15 TIMINGDATA

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0038, Reset Value = 0x1215_0405

Name	Bit	Type	Description	Reset Value
t_wtr	[31:28]	RW	Internal Write to Read command delay, in cycles t_wtr × T (mclk) should be greater than or equal to the minimum value of memory tWTR In LPDDR2-S4 t_wtr is max (2tCK, tWTR) In DDR3 t_wtr is max (4tCK, tWTR)	0x1
t_wr	[27:24]	RW	Write recovery time, in cycles t_wr × T (mclk) should be greater than or equal to the minimum value of memory tWR	0x2
t_rtp	[23:20]	RW	Internal Read to Precharge command delay, in cycles t_rtp × T (mclk) should be greater than or equal to the minimum value of memory tRTP In LPDDR2-S4 t_rtp is max (2tCK, tRTP) In DDR3 t_rtp is max (4tCK, tRTP)	0x1
cl	[19:16]	RW	CAS Latency in cycles cl should be greater than or equal to the minimum value of memory CL	0x5
RSVD	[15:12]	–	Reserved (Should be zero)	0x0
wl	[11:8]	RW	Write data latency (for LPDDR2-S4/DDR3), in cycles wl should be greater than or equal to the minimum value of memory WL	0x4
RSVD	[7:4]	–	Reserved (Should be zero)	0x0
rl	[3:0]	RW	Read data latency (for LPDDR2-S4/DDR3), in cycles rl should be greater than or equal to the minimum value of memory RL	0x5

NOTE: tDAL (Auto precharge Write recovery + precharge time) = t_wr + t_rp (automatically calculated).

18.12.1.16 TIMINGPOWER

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x003C, Reset Value = 0x381B_0422

Name	Bit	Type	Description	Reset Value
t_faw	[31:26]	RW	Four Active Window (for LPDDR2-S4/DDR3) t_faw × T (mclk) should be greater than or equal to the minimum value of memory tFAW	0xE
t_xsr	[25:16]	RW	Self refresh exit power down to next valid command delay, in cycles t_xsr × T (mclk) should be greater than or equal to the minimum value of memory tXSR	0x1B
t_xp	[15:8]	RW	Exit power down to next valid command delay, in cycles t_xp × T (mclk) should be greater than or equal to the minimum value of memory tXP In DDR3, this value should set to tXPDLL. In DDR3 even though “fast exit” is programmed in MRS, tXPDLL is applied.	0x4
t_cke	[7:4]	RW	CKE minimum pulse width (minimum power down mode duration), in cycles t_cke should be greater than or equal to the minimum value of memory tCKE	0x2
t_mrd	[3:0]	RW	Mode Register Set command period, in cycles t_mrd should be greater than or equal to the minimum value of memory tMRD In DDR3, this parameter should be set to tMOD value.	0x2

18.12.1.17 PHYSTATUS

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved (Should be zero)	0x0
ctrl_zq_pmon	[26:24]	R	Control Code Found by Auto Calibration for Pull-up	0x0
RSVD	[23]	–	Reserved (Should be zero)	0x0
ctrl_zq_nmon	[22:20]	R	Control Code Found by Auto Calibration for Pull-down	0x0
RSVD	[19:18]	–	Reserved (Should be zero)	0x0
ctrl_zq_error	[17]	R	Calibration Error When this register is set to 0x1, an error appears when the auto ZQ calibration is complete.	0x0
ctrl_zq_end	[16]	R	Calibration Completion When this register is set to 0x1, no error appears when the auto ZQ calibration is complete.	0x0
RSVD	[15:14]	–	Reserved (Should be zero)	0x0
ctrl_lock_value	[13:4]	R	Locked Delay Locked delay line encoding value ctrl_lock_value[9:2]: Number of delay cells for coarse lock ctrl_lock_value[1:0]: Control value for fine lock	0x0
RSVD	[3]	–	Reserved (Should be zero)	0x0
ctrl_locked	[2]	R	DLL Lock 0 = DLL is unlocked 1 = DLL is locked	0x0
ctrl_flock	[1]	R	Fine Lock Information	0x0
ctrl_clock	[0]	R	Coarse Lock Information	0x0

18.12.1.18 PHYZQCONTROL

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0044, Reset Value = 0xE385_5731

Name	Bit	Type	Description	Reset Value
ctrl_dcc	[31:20]	RW	Clock Duty Control PHY Duty Correction Circuit (DCC) shifts clock negative edge from the original position to the leaded or lagged position. This happens in compensation for the process variations. The value shown here are the delay value at the optimum condition. The maximum value that can be shifted by two pairs of DCC is about ± 60ps. ctrl_dcc[2:0]: 0x0 = Default value ctrl_dcc[2:0]: 0x1 = Negative edge is lagged by 20ps ctrl_dcc[2:0]: 0x2 = Negative edge is lagged by 26ps ctrl_dcc[2:0]: 0x3 = Negative edge is lagged by 29ps ctrl_dcc[2:0]: 0x4 = Negative edge is lagged by 31ps ctrl_dcc[2:0]: 0x5 = Negative edge is lagged by 33ps ctrl_dcc[2:0]: 0x6 = Negative edge is lagged by 34ps ctrl_dcc[2:0]: 0x7 = Negative edge is lagged by 35ps ctrl_dcc[5:3]: 0x7 = Default value ctrl_dcc[5:3]: 0x1 = Negative edge is leaded by 20ps ctrl_dcc[5:3]: 0x2 = Negative edge is leaded by 26ps ctrl_dcc[5:3]: 0x3 = Negative edge is leaded by 29ps ctrl_dcc[5:3]: 0x4 = Negative edge is leaded by 31ps ctrl_dcc[5:3]: 0x5 = Negative edge is leaded by 33ps ctrl_dcc[5:3]: 0x6 = Negative edge is leaded by 34ps ctrl_dcc[5:3]: 0x7 = Negative edge is leaded by 35ps ctrl_dcc[8:6]: Refer to ctrl_dcc[2:0] ctrl_dcc[11:9]: Refer to ctrl_dcc[5:3]	0xE38
ctrl_zq_force_impp	[19:17]	RW	Immediate Control Code for Pull-up	0x2
ctrl_zq_force_impn	[16:14]	RW	Immediate Control Code for Pull-down	0x5
ctrl_zq_mode_term	[13:11]	RW	On-die-termination Resistor Value Selection 0x0 = Reserved 0x1 = 120 ohm 0x2 = 60 ohm 0x3 = 40 ohm 0x4 = 30ohm 0x5~0x7 = Reserved	0x2
ctrl_zq_mode_dds	[10:8]	RW	Driver Strength Selection 0x0~0x3 = Reserved 0x4 = 48 ohm 0x5 = 40 ohm 0x6 = 34 ohm 0x7 = 30 ohm	0x7

Name	Bit	Type	Description	Reset Value
RSVD	[7]	–	Reserved (Should be zero)	0x0
ctrl_zq_div	[6:4]	RW	Calibration I/O Clock Selection Signal 0x0 = mclk/2 0x1 = mclk/4 0x2 = mclk/8 0x3 = mclk/16 0x4 = mclk/32 0x5 to 0x7 = Reserved	0x3
RSVD	[3]	–	Reserved (Should be zero)	0x0
ctrl_zq_force	[2]	RW	Force Calibration When this register is set, it uses ctrl_force_impp[2:0]/impn[2:0] instead of calibration control code found after auto calibration.	0x0
ctrl_zq_start	[1]	RW	Auto Calibration Start Signal ZQ I/O calibration starts by setting this register.	0x0
ctrl_zq_mode_noterm	[0]	RW	Termination Disable Selection 0x0 = Enables termination 0x1 = Disables termination	0x1

18.12.1.19 CHIP0STATUS

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0048, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
bank7_state	[31:28]	R	Current State of Bank 7 of Memory Chip0	0x0
bank6_state	[27:24]	R	Current State of Bank 6 of Memory Chip0	0x0
bank5_state	[23:20]	R	Current State of Bank 5 of Memory Chip0	0x0
bank4_state	[19:16]	R	Current State of Bank 4 of Memory Chip0	0x0
bank3_state	[15:12]	R	Current State of Bank 3 of Memory Chip0	0x0
bank2_state	[11:8]	R	Current State of Bank 2 of Memory Chip0	0x0
bank1_state	[7:4]	R	Current State of Bank 1 of Memory Chip0	0x0
bank0_state	[3:0]	R	Current State of Bank 0 of Memory Chip0 0x0 = Idle (precharged) 0x1 = MRS/EMRS/ZQOPER(DDR3)/ZQCS(DDR3) 0x2 = Deep power down 0x3 = Self refresh 0x4 = Auto refresh 0x5 = Precharge power down 0x6 = Row active 0x7 = Active power down 0x8 = Write 0x9 = Write with auto precharge 0xA = Read 0xB = Read with auto precharge 0xC = Burst stop 0xD = Precharging 0xE = MRR 0xF = Reserved	0x0

18.12.1.20 CHIP1STATUS

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x004C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
bank7_state	[31:28]	R	Current state of bank 7 of memory chip1	0x0
bank6_state	[27:24]	R	Current state of bank 6 of memory chip1	0x0
bank5_state	[23:20]	R	Current state of bank 5 of memory chip1	0x0
bank4_state	[19:16]	R	Current state of bank 4 of memory chip1	0x0
bank3_state	[15:12]	R	Current state of bank 3 of memory chip1	0x0
bank2_state	[11:8]	R	Current state of bank 2 of memory chip1	0x0
bank1_state	[7:4]	R	Current state of bank 1 of memory chip1	0x0
bank0_state	[3:0]	R	Current state of bank 0 of memory chip1 0x0 = Idle (precharged) 0x1 = MRS/EMRS/ZQOPER(DDR3)/ZQCS(DDR3) 0x2 = Deep power down 0x3 = Self refresh 0x4 = Auto refresh 0x5 = Precharge power down 0x6 = Row active 0x7 = Active power down 0x8 = Write 0x9 = Write with auto precharge 0xA = Read 0xB = Read with auto precharge 0xC = Burst stop 0xD = Precharging 0xE = MRR 0xF = Reserved	0x0

18.12.1.21 AREFSTATUS

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved (Should be zero)	0x0
aref_cnt	[15:0]	R	Current Value of Auto Refresh Counter Shows the current value of all bank auto refresh counter. It updates this when a new t_refi is programmed into the TimingAref register and decreases by 1 at the rising edge of mclk. An all bank auto refresh command is issued to memory device. This counter is reloaded with TimingAref.t_ref if it becomes zero.	0x0

18.12.1.22 MRSTATUS

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved (Should be zero)	0x0
mr_status	[7:0]	R	Mode registers status	0x0

18.12.1.23 PHYTEST0

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0058, Reset Value = 0x3FFF_1310

Name	Bit	Type	Description	Reset Value
ctrl_fb_cnt4	[31:24]	R	Count value for control channel	0x0
RSVD	[23:21]	–	Reserved (Should be zero)	0x0
ctrl_fb_oky	[20:16]	R	ctrl_fb_okay[4] = Okay for control ctrl_fb_okay[3:0] = Okay for data	0x0
ctrl_dis	[15]	RW	Clock Output Disable This field controls the CK/CKB 1'b0 = Enables clock output 1'b1 = Disables clock output	0x0
RSVD	[14:13]	–	Reserved (Should be zero)	0x0
ctrl_fb_err	[12:8]	R	ctrl_fb_err[4] = Error for control ctrl_fb_err[3:0] = Error for data	0x0
ctrl_fnc_fb	[7:5]	RW	Function feedback test Valid only when {mode_phy, mode_nand, mode_scan, mode_bypass, mode_mux} is set to 0. 0x0 = Normal operation mode 0x1 = Reserved 0x2 = External FNC feedback test mode 0x3 = Internal FNC feedback test mode 0x4 = Board PHY external read feedback 0x5 = Board PHY internal read feedback 0x6 = Board PHY internal write feedback 0x7 = Reserved	0x0
ctrl_fb_start	[4:0]	RW	ctrl_fb_start[4] = Start for control ctrl_fb_start[3:0] = Start for data	0x0

18.12.1.24 PHYTEST1

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x005C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ctrl_fb_cnt3	[31:24]	R	Count value for data3 channel	0x0
ctrl_fb_cnt2	[23:16]	R	Count value for data2 channel	0x0
ctrl_fb_cnt1	[15:8]	R	Count value for data1 channel	0x0
ctrl_fb_cnt0	[7:0]	R	Count value for data0 channel	0x0

18.12.1.25 QOSCONTROL

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000 (QOSCONTROL0)
- Address = Base Address + 0x0068, Reset Value = 0x0000_0000 (QOSCONTROL1)
- Address = Base Address + 0x0070, Reset Value = 0x0000_0000 (QOSCONTROL2)
- Address = Base Address + 0x0078, Reset Value = 0x0000_0000 (QOSCONTROL3)
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000 (QOSCONTROL4)
- Address = Base Address + 0x0088, Reset Value = 0x0000_0000 (QOSCONTROL5)
- Address = Base Address + 0x0090, Reset Value = 0x0000_0000 (QOSCONTROL6)
- Address = Base Address + 0x0098, Reset Value = 0x0000_0000 (QOSCONTROL7)
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000 (QOSCONTROL8)
- Address = Base Address + 0x00A8, Reset Value = 0x0000_0000 (QOSCONTROL9)
- Address = Base Address + 0x00B0, Reset Value = 0x0000_0000 (QOSCONTROL10)
- Address = Base Address + 0x00B8, Reset Value = 0x0000_0000 (QOSCONTROL11)
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000 (QOSCONTROL12)
- Address = Base Address + 0x00C8, Reset Value = 0x0000_0000 (QOSCONTROL13)
- Address = Base Address + 0x00D0, Reset Value = 0x0000_0000 (QOSCONTROL14)
- Address = Base Address + 0x00D8, Reset Value = 0x0000_0000 (QOSCONTROL15)

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved (Should be zero)	0x0
qos_cnt	[27:16]	RW	QoS Cycles 0xn = n aclk cycles The matched ARID uses this value for its timeout counters instead of ConControl.timeout_cnt.	0x0
qos_cnt_f	[15:4]	RW	QoS Cycles for Fast Request 0xn = n aclk cycles When you enable Concontrol.qos_fast_en and input pin qos_fast[n] bit is set to 1, this qos_cnt_f value is loaded to the timeout counter.	0x0
RSVD	[3:2]	–	Reserved (Should be zero)	0x0
qos_en	[1:0]	RW	QoS Enable 0x00 = Disables QoS 0x01 = Enables Read ID QoS 0x10 = Enables Write ID QoS 0x11 = Enables Read and Write ID QoS When you enable this function, its timeout counter works. AxID (ARID/AWID or both ARID & AWID) is masked with QoSConfig.qos_mask and compared with QoSConfig.qos_id.	0x0

18.12.1.26 QOSCONFIG

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x0064, Reset Value = 0x0000_0000 (QOSCONFIG0)
- Address = Base Address + 0x006C, Reset Value = 0x0000_0000 (QOSCONFIG1)
- Address = Base Address + 0x0074, Reset Value = 0x0000_0000 (QOSCONFIG2)
- Address = Base Address + 0x007C, Reset Value = 0x0000_0000 (QOSCONFIG3)
- Address = Base Address + 0x0084, Reset Value = 0x0000_0000 (QOSCONFIG4)
- Address = Base Address + 0x008C, Reset Value = 0x0000_0000 (QOSCONFIG5)
- Address = Base Address + 0x0094, Reset Value = 0x0000_0000 (QOSCONFIG6)
- Address = Base Address + 0x009C, Reset Value = 0x0000_0000 (QOSCONFIG7)
- Address = Base Address + 0x00A4, Reset Value = 0x0000_0000 (QOSCONFIG8)
- Address = Base Address + 0x00AC, Reset Value = 0x0000_0000 (QOSCONFIG9)
- Address = Base Address + 0x00B4, Reset Value = 0x0000_0000 (QOSCONFIG10)
- Address = Base Address + 0x00BC, Reset Value = 0x0000_0000 (QOSCONFIG11)
- Address = Base Address + 0x00C4, Reset Value = 0x0000_0000 (QOSCONFIG12)
- Address = Base Address + 0x00CC, Reset Value = 0x0000_0000 (QOSCONFIG13)
- Address = Base Address + 0x00D4, Reset Value = 0x0000_0000 (QOSCONFIG14)
- Address = Base Address + 0x00DC, Reset Value = 0x0000_0000 (QOSCONFIG15)

Name	Bit	Type	Description	Reset Value
qos_mask	[31:16]	RW	QoS Mask Bits You can use this to mask the incoming AxID to compare with the qos_id. For example, to have 0b00110XX000 IDs in the same QoS, it should mask the 4th and 5th bits. Therefore, qos_mask would be 0b1111100111.	0x0
qos_id	[15:0]	RW	QoS ID You can use this to compare with the masked AxID to verify whether its timeout counter should be used for QoS. After applying the qos_mask to the AxID, it is compared with qos_id. The qos_id would be 0b001100_0000 by using the previous example. When you compare the masked ID, if the result is equal to the qos_id, then the QoSControl0.qos_cnt is applied to this AxID transaction for timeout. Don't care bits must be assigned zeros.	0x0

18.12.1.27 QOSTIMEOUT0

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x00E0, Reset Value = 0x0FFF_0FFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved (Should be zero)	0x0
Timeout _cnt_level2	[27:16]	RW	QoS Timeout Count 2 0xn = n aclk cycles (aclk: AXI clock) This counter prevents transactions in the command request buffer from starvation. This counter starts when a new AXI transaction enters into the request buffer. When the counter becomes zero, the corresponding transaction becomes the highest priority command of all the transactions in the command request buffer. You can use this as a default timeout counter when AxQOS of a transaction coming in is 2. QoS counter overrides it if the ARID or AWID matches with the QoS ID when it enters into the command queue. Refer to Chapter *18.7 Quality of Service* for more information.	0xFFF
RSVD	[15:12]	–	Reserved (Should be zero)	0x0
Timeout _cnt_level1	[11:0]	RW	QoS Timeout Count 1 0xn = n aclk cycles (aclk: AXI clock) This counter prevents transactions in the command request buffer from starvation. This counter starts when a new AXI transaction enters into the request buffer. When the counter becomes zero, the corresponding transaction becomes the highest priority command of all the transactions in the command request buffer. You can use this as a default timeout counter when AxQOS of a transaction coming in is 1. QoS counter overrides it if the ARID or AWID matches with the QoS ID when it enters into the command queue. Refer to Chapter *18.7 Quality of Service* for more information.	0xFFF

18.12.1.28 QOSTIMEOUT1

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x00E4, Reset Value = 0x0FFF_0FFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved (Should be zero)	0x0_FFF0
Timeout _cnt_level3	[11:0]	RW	QoS Timeout Count 3 0xn = n aclk cycles (aclk: AXI clock) This counter prevents transactions in the command request buffer from starvation. This counter starts when a new AXI transaction enters into the request buffer. When the counter becomes zero, the corresponding transaction becomes the highest priority command of all the transactions in the command request buffer. You can use this as a default timeout counter when AxQOS of a transaction coming in is 3. QoS counter overrides it if the ARID or AWID matches with the QoS ID comes into the command queue. Refer to Chapter *18.7 Quality of Service* for more information.	0xFFF

18.12.1.29 IVCONTROL

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x00F0, Reset Value = 0x1000_000C

Name	Bit	Type	Description	Reset Value
iv_on	[31]	RW	Memory Channel Interleaving Enable 0x0 = Disables 0x1 = Enables When you disable the bit, AXI early Write response is also disabled	0x1
RSVD	[30:8]	–	Reserved (Should be zero)	0x0
iv_size	[7:0]	RW	Memory Channel Interleaving Size 0x00 to 0x03 = Reserved 0x04 = 16 byte 0x05 = 32 byte 0x06 = 64 byte 0x07 = 128 byte 0x08 = 256 byte 0x09 = 512 byte 0x0A = 1K byte 0x0B = 2K byte 0x0C = 4K byte 0x0D = 8K byte 0x0E = 16K byte 0x0F = 32K byte 0x10 = 64K byte 0x11 = 128K byte 0x12 = 256K byte 0x13 = 512K byte 0x14 = 1M byte 0x15 = 2M byte 0x16 = 4M byte 0x17 = 8M byte 0x18 = 16M byte 0x19 = 32M byte 0x1A = 64M byte 0x1B = 128M byte 0x1C = 256M byte 0x1D = 512M byte 0x1E = 1G byte 0x1F = 2G byte 0x20 to 0xFF = Reserved	0x0C

18.12.1.30 PEREVCONFIG

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0x00FC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved (Should be zero)	0x0
perev_ mode_m	[16:12]	RW	MCLK domain performance event 4 events represents arbitration that occur as follows:. 0x0 = {2'b0,read hit, write hit} to chip0 0x1 = Read hit to bank[3:0] of chip0 0x2 = Read hit to bank[7:4] of chip0 0x3 = Write hit to bank[3:0] of chip0 0x4 = Write hit to bank[7:4] of chip0 0x5 = {2'b0,read miss, write miss} to chip0 0x6 = Read miss to bank[3:0] of chip0 0x7 = Read miss to bank[7:4] of chip0 0x8 = Write miss to bank[3:0] of chip0 0x9 = Write miss to bank[7:4] of chip0 0xa = {Timeout read miss, timeout write miss, hazard read miss, hazard write miss} of chip0 0xb = {1'b0, powerdown, self-refresh, memory data transfer} of chip0 0xc = {2'b0,read hit, write hit} to chip1 0xd = Read hit to bank[3:0] of chip1 0xe = Read hit to bank[7:4] of chip1 0xf = Write hit to bank[3:0] of chip1 0x10 = Write hit to bank[7:4] of chip1 0x11 = {2'b0,read miss, write miss} to chip1 0x12 = Read miss to bank[3:0] of chip1 0x13 = Read miss to bank[7:4] of chip1 0x14 = Write miss to bank[3:0] of chip1 0x15 = Write miss to bank[7:4] of chip1 0x16 = {Timeout read miss, timeout write miss, hazard read miss, hazard write miss} of chip1 0x17 = {1'b0, powerdown, self-refresh, memory data transfer} of chip1 0x18 to 0x1f = Reserved	0x0
RSVD	[11:9]	–	Reserved (Should be zero)	0x0
perev_ mode_a	[8:4]	RW	ACLK domain performance event 4 events represents AXI request that occur as follows. 0x0 = {Chip1 read, chip1 write, chip0 read, chip0 write} 0x1 = {2'b0, read, write} to channel0, chip0 0x2 = Read to bank[3:0] of channel0, chip0 0x3 = Read to bank[7:0] of channel0, chip0 0x4 = Write to bank[3:0] of channel0, chip0 0x5 = Write to bank[7:0] of channel0, chip0 0x6 = {2'b0, read, write} to channel0, chip1 0x7 = Read to bank[3:0] of channel0, chip1 0x8 = Read to bank[7:0] of channel0, chip1 0x9 = Write to bank[3:0] of channel0, chip1	0x0

Name	Bit	Type	Description	Reset Value
			0xa = Write to bank[7:0] of channel0, chip1 0xb = {2'b0, read, write} to channel1, chip0 0xc = Read to bank[3:0] of channel1, chip0 0xd = Read to bank[7:0] of channel1, chip0 0xe = Write to bank[3:0] of channel1, chip0 0xf = Write to bank[7:0] of channel1, chip0 0x10 = {2'b0, read, write} to channel1, chip1 0x11 = Read to bank[3:0] of channel1, chip1 0x12 = Read to bank[7:0] of channel1, chip1 0x13 = Write to bank[3:0] of channel1, chip1 0x14 = Write to bank[7:0] of channel1, chip1 0x15 = {Chip1 split read, chip1 split write, chip0 split read, chip0 split write} 0x16 to 0x1f = Reserved	
RSVD	[3:1]	–	Reserved (Should be zero)	0x0
perev_en	[0]	RW	Performance event module enable 0x0 = Disables this module 0x1 = Enables this module	0x0

18.12.1.31 PMNC_PPC_A (M)

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0xE000 (PPC_A), 0xF0001 (PPC_B), Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:18]	–	Reserved (Should be zero)	0x0
START MODE	[16]	RW	PPC Start Mode 0x0 = SW (by CPU) 0x1 = HW (by SYSCON)	0x0
RSVD	[15:4]	–	Reserved (Should be zero)	0x0
CC DIVIDER	[3]	RW	Cycle count divider 0x0 = Counts every processor clock cycle, reset value 0x1 = Counts every 64 th processor clock cycle	0x0
CC RESET	[2]	W	Cycle counter reset 0x0 = No action 0x1 = Resets cycle counter, CCNT, to zero	0x0
PPC COUNTER RESET	[1]	W	Performance counter reset 0x0 = No action 0x1 = Resets all performance counters to zero	0x0
PPC ENABLE	[0]	RW	Enable bit 0x0 = Disables all counters including CCNT 0x1 = Enables all counters including CCNT When you read it, 1 means it is counting and 0 means it is idle (stop counting). You can write it only when the start mode is set to 0 (CPU starts PPMU). Now, you can write this bit by 1 to start counting and write it by 0 to stop the counting. When the start mode is set to 1 (SYSCON starts PPMU), you only can read it and get the status of the PPMU. Now, the external trigger controls PPMU. When external trigger is set to 1, counting starts, and when external trigger is set to 0, counting stops.	0x0

18.12.1.32 CNTENS_PPC_A (M)

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0xE010 (PPC_A), 0xF010 (PPC_B), Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	Enables cycle counter	0x0
RSVD	[30:4]	–	Reserved (Should be zero)	0x0
PMCNT3	[3]	RW	Enables Counter 3	0x0
PMCNT2	[2]	RW	Enables Counter 2	0x0
PMCNT1	[1]	RW	Enables Counter 1	0x0
PMCNT0	[0]	RW	Enables Counter 0	0x0

18.12.1.33 CNTENC_PPC_A (M)

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0xE020 (PPC_A), 0xF020 (PPC_B), Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	Disables cycle counter	0x0
RSVD	[30:4]	–	Reserved (Should be zero)	0x0
PMCNT3	[3]	RW	Disables Counter 3	0x0
PMCNT2	[2]	RW	Disables Counter 2	0x0
PMCNT1	[1]	RW	Disables Counter 1	0x0
PMCNT0	[0]	RW	Disables Counter 0	0x0

18.12.1.34 INTENS_PPC_A (M)

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0xE030 (PPC_A), 0xF030 (PPC_B), Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	CCNT overflow interrupt enable	0x0
RSVD	[30:4]	–	Reserved (Should be zero)	0x0
PMCNT3	[3]	RW	Counter 3 overflow interrupt enable	0x0
PMCNT2	[2]	RW	Counter 2 overflow interrupt enable	0x0
PMCNT1	[1]	RW	Counter 1 overflow interrupt enable	0x0
PMCNT0	[0]	RW	Counter 0 overflow interrupt enable	0x0

18.12.1.35 INTENC_PPC_A (M)

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0xE040 (PPC_A), 0xF040 (PPC_B), Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	CCNT overflow interrupt disable	0x0
RSVD	[30:4]	–	Reserved (Should be zero)	0x0
PMCNT3	[3]	RW	Counter 3 overflow interrupt disable	0x0
PMCNT2	[2]	RW	Counter 2 overflow interrupt disable	0x0
PMCNT1	[1]	RW	Counter 1 overflow interrupt disable	0x0
PMCNT0	[0]	RW	Counter 0 overflow interrupt disable	0x0

18.12.1.36 FLAG_PPC_A (M)

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0xE050 (PPC_A), 0xF050 (PPC_B), Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31]	RW	Cycle counter overflow flag	0x0
RSVD	[30:4]	–	Reserved (Should be zero)	0x0
PMCNT3	[3]	RW	Counter 3 overflow flag	0x0
PMCNT2	[2]	RW	Counter 2 overflow flag	0x0
PMCNT1	[1]	RW	Counter 1 overflow flag	0x0
PMCNT0	[0]	RW	Counter 0 overflow flag	0x0

18.12.1.37 CCNT_PPC_A (M)

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0xE100 (PPC_A), 0xF100 (PPC_B), Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CCNT	[31:0]	RW	CCNT Register includes an event count	0x0

18.12.1.38 PMCNT0_PPC_A (M)

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0xE110 (PPC_A), 0xF110 (PPC_B), Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PMCNT	[31:0]	RW	PMCNT Register includes an event count	0x0

18.12.1.39 PMCNT1_PPC_A (M)

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0xE120 (PPC_A), 0xF120 (PPC_B), Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PMCNT	[31:0]	RW	PMCNT Register includes an event count	0x0

18.12.1.40 PMCNT2_PPC_A (M)

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0xE130 (PPC_A), 0xF130 (PPC_B), Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PMCNT	[31:0]	RW	PMCNT Register includes an event count	0x0

18.12.1.41 PMCNT3_PPC_A (M)

- Base Address: 0x1060_0000, 0x1061_0000
- Address = Base Address + 0xE140 (PPC_A), 0xF140 (PPC_B), Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PMCNT	[31:0]	RW	PMCNT Register includes an event count	0x0

19 SROM Controller

19.1 Overview

Exynos 4412 SCP SROM Controller (SROMC) supports:

- External 8/16-bit NOR Flash/PROM/ SRAM memory.
- 4-bank memory up to maximum 16 Mbyte per bank.

19.1.1 Features of SROMC

The features of SROMC are:

- Supports SRAM, various ROMs, and NOR flash memory
- Supports only 8 or 16-bit data bus
- Address space: Up to 16 MB per bank
- Supports 4-bank.
- Fixed memory bank start address
- External wait to extend the bus cycle
- Supports byte and half-word access for external memory

19.1.2 Block Diagram

Figure 19-1 illustrates the block diagram of SROMC introduction.

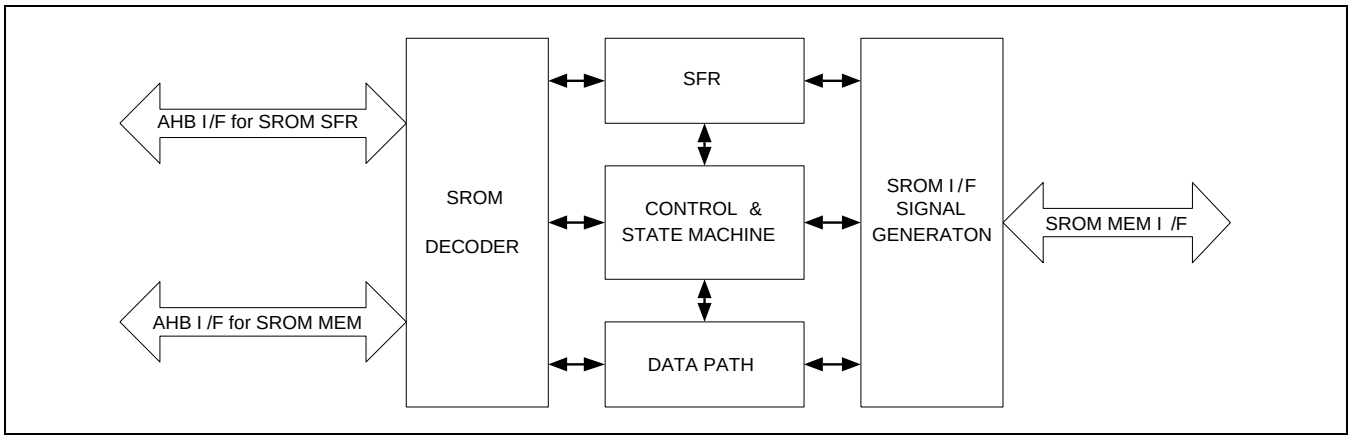


Figure 19-1 Block Diagram of SROMC Introduction

19.2 Functional Description

SROMC supports SROM interface for Bank 0 to Bank 3. This section includes:

- nWAIT Pin Operation
- Programmable Access Cycle

19.2.1 nWAIT Pin Operation

When it enables nWAIT signal corresponding to each memory bank, the external nWAIT pin should prolong the duration of nOE while the memory bank is active. It verifies the nWAIT from tacc-1 and deasserts the nOE at the next clock after sampling nWAIT is high. The nWE signal has the similar relation with nOE signal.

Figure 19-2 illustrates the SROMC nWAIT timing diagram.

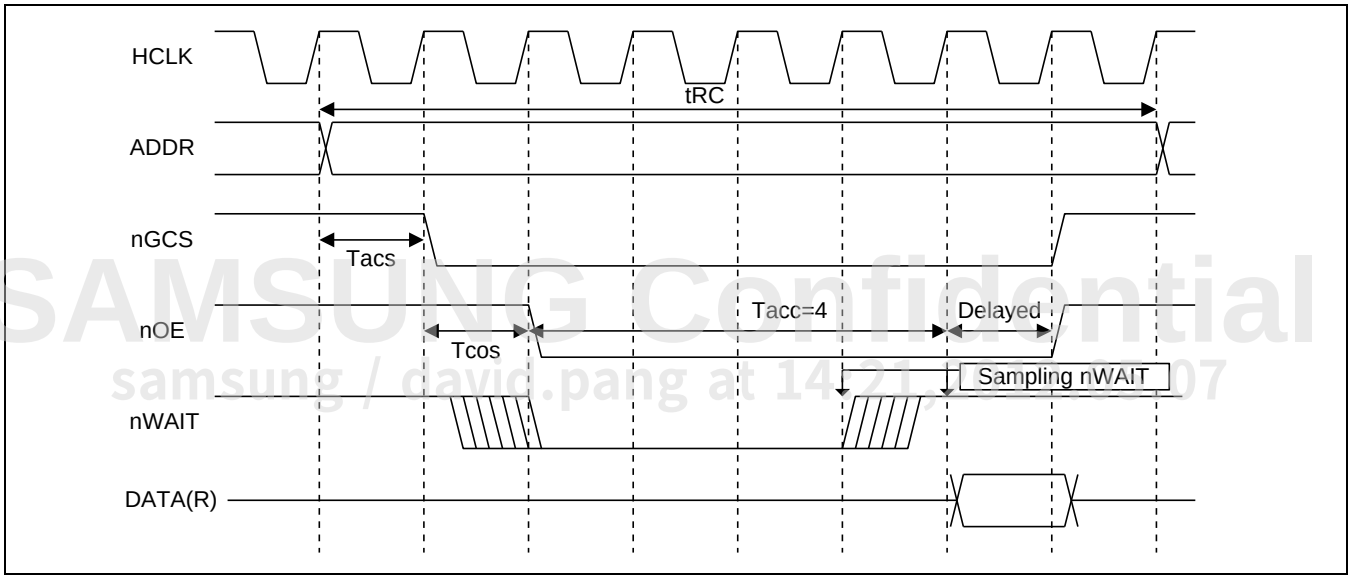


Figure 19-2 SROMC nWAIT Timing Diagram

19.2.2 Programmable Access Cycle

Figure 19-3 illustrates the SROMC read timing diagram.

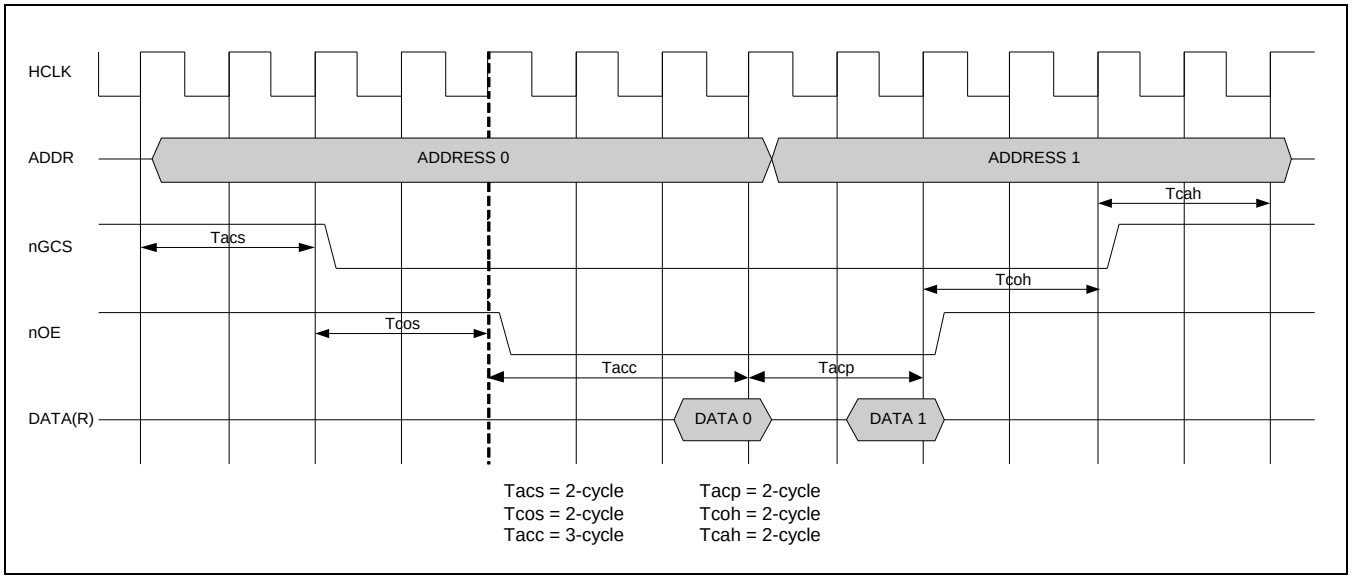


Figure 19-3 SROMC Read Timing Diagram

Figure 19-4 illustrates the SROMC write timing diagram.

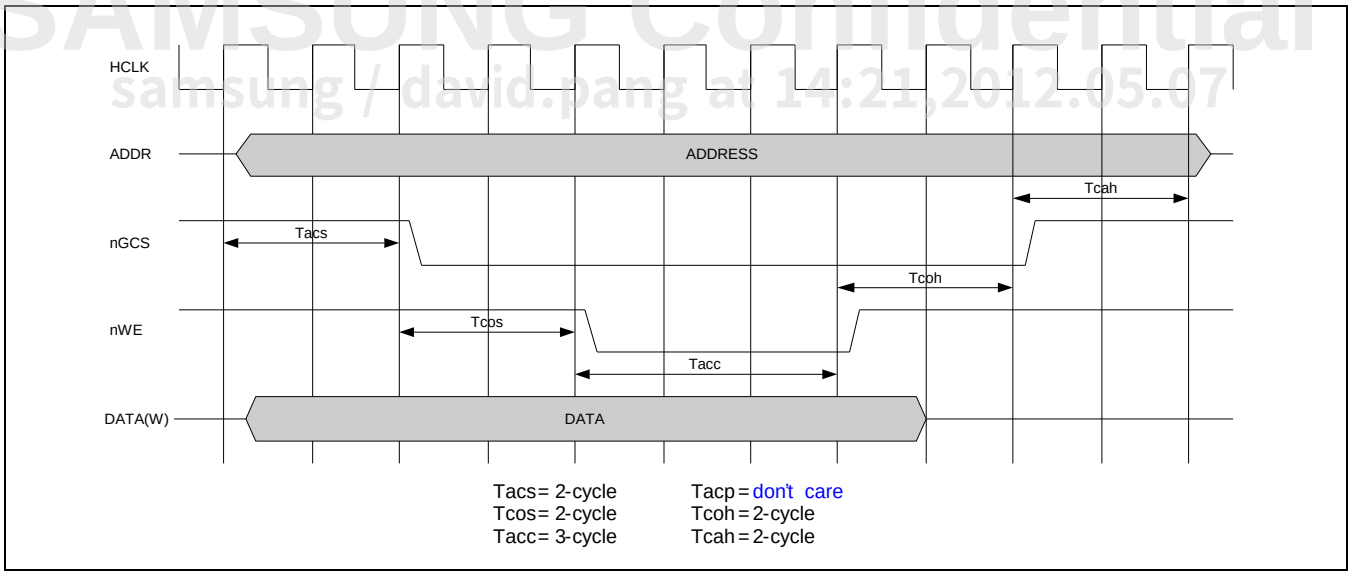


Figure 19-4 SROMC Write Timing Diagram

19.3 I/O Description

This section describes the I/O description of SROMC.

Signal	I/O	Description	Pad	Type
nGCS[3:0]	Output	Bank selection signal	Xm0CSn_x	muxed
ADDR[15:0]	Output	SROM address bus	Xm0ADDR_x	muxed
nOE	Output	SROM output enable	Xm0OEn	muxed
nWE	Output	SROM write enable	Xm0WEn	muxed
nWBE/nBE[1:0]	Output	SROM byte write enable/byte enable	Xm0BEn_x	muxed
DATA[15:0]	In/Out	SROM data bus	Xm0DATA_x	muxed
nWAIT	Input	SROM wait input	Xm0WAITn	muxed

19.4 Register Description

19.4.1 Register Map Summary

- Base Address: 0x1257_0000

Register	Offset	Description	Reset Value
SROM_BW	0x0000	Specifies the SROM bus width and wait control	0x0000_0009
SROM_BC0	0x0004	Specifies the SROM bank 0 control register	0x000F_0000
SROM_BC1	0x0008	Specifies the SROM bank 1 control register	0x000F_0000
SROM_BC2	0x000C	Specifies the SROM bank 2 control register	0x000F_0000
SROM_BC3	0x0010	Specifies the SROM bank 3 control register	0x000F_0000

19.4.1.1 SROM_BW

- Base Address: 0x1257_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0009

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
ByteEnable3	[15]	RW	nWBE/nBE (for UB/LB) control for memory bank 3 0 = Does not use UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Uses UB/LB (XrnWBE[1:0] is dedicated nBE[1:0])	0
WaitEnable3	[14]	RW	Wait enable control for memory bank 3 0 = Disables WAIT 1 = Enables WAIT	0
AddrMode3	[13]	RW	Select SROM ADDR base for memory bank 3 0 = SROM_ADDR is half-word base address. (SROM_ADDR[22:0] ← HADDR[23:1]) 1 = SROM_ADDR is byte base address (SROM_ADDR[22:0] ← HADDR[22:0]) NOTE: When DataWidth3 is "0", SROM_ADDR is byte base address. (It ignores this bit.)	0
DataWidth3	[12]	RW	Data bus width control for memory bank 3 0 = 8-bit 1 = 16-bit	0
ByteEnable2	[11]	RW	nWBE/nBE (for UB/LB) control for memory bank 2 0 = Does not use UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Uses UB/LB (XrnWBE[1:0] is dedicated nBE[1:0])	0
WaitEnable2	[10]	RW	Wait enable control for memory bank 2 0 = Disables WAIT 1 = Enables WAIT	0
AddrMode2	[9]	RW	Select SROM ADDR Base for memory bank 2 0 = SROM_ADDR is half-word base address. (SROM_ADDR[22:0] ← HADDR[23:1]) 1 = SROM_ADDR is byte base address (SROM_ADDR[22:0] ← HADDR[22:0]) NOTE: When DataWidth2 is "0", SROM_ADDR is byte base address. (It ignores this bit.)	0
DataWidth2	[8]	RW	Data bus width control for memory bank 2 0 = 8-bit 1 = 16-bit	0
ByteEnable1	[7]	RW	nWBE/nBE (for UB/LB) control for memory bank 1 0 = Does not use UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Uses UB/LB (XrnWBE[1:0] is dedicated nBE[1:0])	0
WaitEnable1	[6]	RW	Wait enable control for memory bank 1	0

Name	Bit	Type	Description	Reset Value
			0 = Disables WAIT 1 = Enables WAIT	
AddrMode1	[5]	RW	Select SROM ADDR base for memory bank 1 0 = SROM_ADDR is half-word base address. (SROM_ADDR[22:0] ← HADDR[23:1]) 1 = SROM_ADDR is byte base address (SROM_ADDR[22:0] ← HADDR[22:0]) NOTE: When DataWidth1 is "0", SROM_ADDR is byte base address. (It ignores this bit.)	0
DataWidth1	[4]	RW	Data bus width control for memory bank 1 0 = 8-bit 1 = 16-bit	0
ByteEnable0	[3]	RW	nWBE/nBE (for UB/LB) control for memory bank 0 0 = Does not use UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Uses UB/LB (XrnWBE[1:0] is dedicated nBE[1:0])	1
WaitEnable0	[2]	RW	Wait enable control for memory bank 0 0 = Disables WAIT 1 = Enables WAIT	0
AddrMode0	[1]	RW	Select SROM ADDR base for memory bank 0 0 = SROM_ADDR is half-word base address. (SROM_ADDR[22:0] ← HADDR[23:1]) 1 = SROM_ADDR is byte base address (SROM_ADDR[22:0] ← HADDR[22:0]) NOTE: When DataWidth0 is "0", SROM_ADDR is byte base address. (It ignores this bit.)	0
DataWidth0	[0]	RW	Data bus width control for memory bank 0 0 = 8-bit 1 = 16-bit	1

19.4.1.2 SROM_BCn (n = 0 to 3)

- Base Address: 0x1257_0000
- Address = Base Address + 0x0004, Reset Value = 0x000F_0000 (SROM_BC0)
- Address = Base Address + 0x0008, Reset Value = 0x000F_0000 (SROM_BC1)
- Address = Base Address + 0x000C, Reset Value = 0x000F_0000 (SROM_BC2)
- Address = Base Address + 0x0010, Reset Value = 0x000F_0000 (SROM_BC3)

Name	Bit	Type	Description	Reset Value
Tacs	[31:28]	RW	Address set-up before nGCS 0000 = 0 Clock 0001 = 1 Clocks 0010 = 2 Clocks 0011 = 3 Clocks …… 1100 = 12 Clocks 1101 = 13 Clocks 1110 = 14 Clocks 1111 = 15 Clocks NOTE: More 1-2 cycles according to bus i/f status	0000
Tcos	[27:24]	RW	Chip selection set-up before nOE 0000 = 0 Clock 0001 = 1 Clocks 0010 = 2 Clocks 0011 = 3 Clocks …… 1100 = 12 Clocks 1101 = 13 Clocks 1110 = 14 Clocks 1111 = 15 Clocks	0000
RSVD	[23:21]	–	Reserved	000
Tacc	[20:16]	RW	Access cycle 00000 = 1 Clock 00001 = 2 Clocks 00001 = 3 Clocks 00010 = 4 Clocks …… 11100 = 29 Clocks 11101 = 30 Clocks 11110 = 31 Clocks 11111 = 32 Clocks	01111
Tcoh	[15:12]	RW	Chip selection hold on nOE 0000 = 0 Clock 0001 = 1 Clocks 0010 = 2 Clocks 0011 = 3 Clocks …… 1100 = 12 Clocks 1101 = 13 Clocks	0000

Name	Bit	Type	Description	Reset Value
			1110 = 14 Clocks 1111 = 15 Clocks	
Tcah	[11:8]	RW	Address holding time after nGCSn 0000 = 0 Clock 0001 = 1 Clocks 0010 = 2 Clocks 0011 = 3 Clocks …… 1100 = 12 Clocks 1101 = 13 Clocks 1110 = 14 Clocks 1111 = 15 Clocks NOTE: More 1-2 cycles according to bus i/f status	0000
Tacp	[7:4]	RW	Page mode access cycle at Page mode 0000 = 0 Clock 0001 = 1 Clocks 0010 = 2 Clocks 0011 = 3 Clocks …… 1100 = 12 Clocks 1101 = 13 Clocks 1110 = 14 Clocks 1111 = 15 Clocks	0000
RSVD	[3:2]	–	Reserved	–
PMC	[1:0]	RW	Page mode configuration 00 = Normal (1 Data) 01 = 4 Data 10 = Reserved 11 = Reserved	00

20 NAND Flash Controller

20.1 Overview

Due to the recent increase in the prices of NOR flash memory and the moderately priced DRAM, and NAND flash, customers prefer to execute boot code on NAND flash and execute the main code on DRAM.

The boot code in Exynos 4412 SCP can be executed on external NAND flash. It copies NAND flash data to DRAM. To validate the NAND flash data, Exynos 4412 SCP includes hardware Error Correction Code (ECC). After the NAND flash content is copied to DRAM, main program will be executed on DRAM.

20.2 Features

The features of NAND flash controller are:

- Auto boot: The boot code is transferred to internal SRAM during reset. After the transfer, the boot code will be executed on the SRAM.
- NAND flash memory interface: Supports 512 Bytes, 2 KB, 4 KB, and 8 KB pages.
- Software mode: You can directly access NAND flash memory, for example, this feature can be used in read/erase/program NAND flash memory.
- Interface: Supports 8-bit NAND flash memory interface bus.
- Generates, detects, and indicates hardware ECC (software correction).
- Supports both Single Level Cell (SLC) and Multi Level Cell (MLC) NAND flash memories.
- ECC: Supports 1-/4-/8-/12-/16-bit ECC.
- SFR interface: Supports byte/half word/word access to Data and ECC data registers, and Word access to other registers.

20.2.1 Block Diagram

Figure 20-1 illustrates the NAND Flash Controller block diagram.

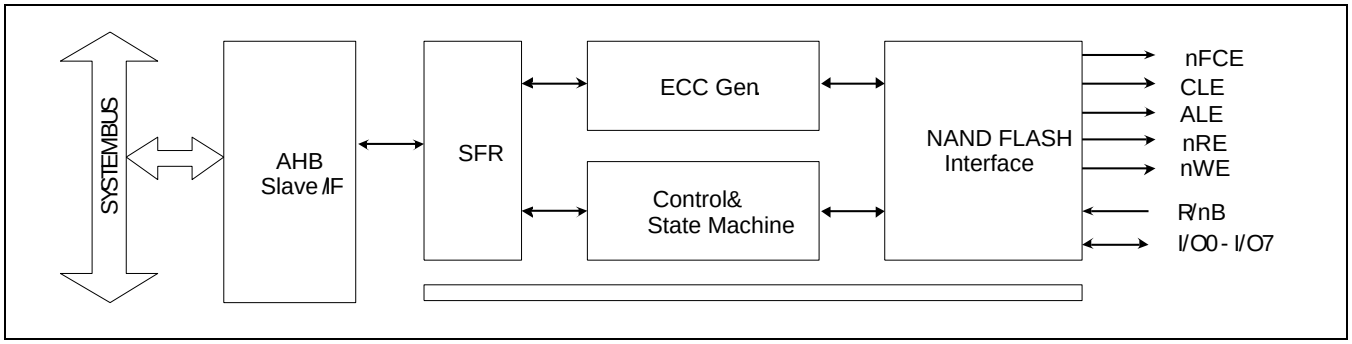


Figure 20-1 NAND Flash Controller Block Diagram

20.2.2 NAND Flash Memory Timing

Figure 20-2 illustrates the CLE and ALE timing (TACLS = 1, TWRPH0 = 0, TWRPH1 = 0).

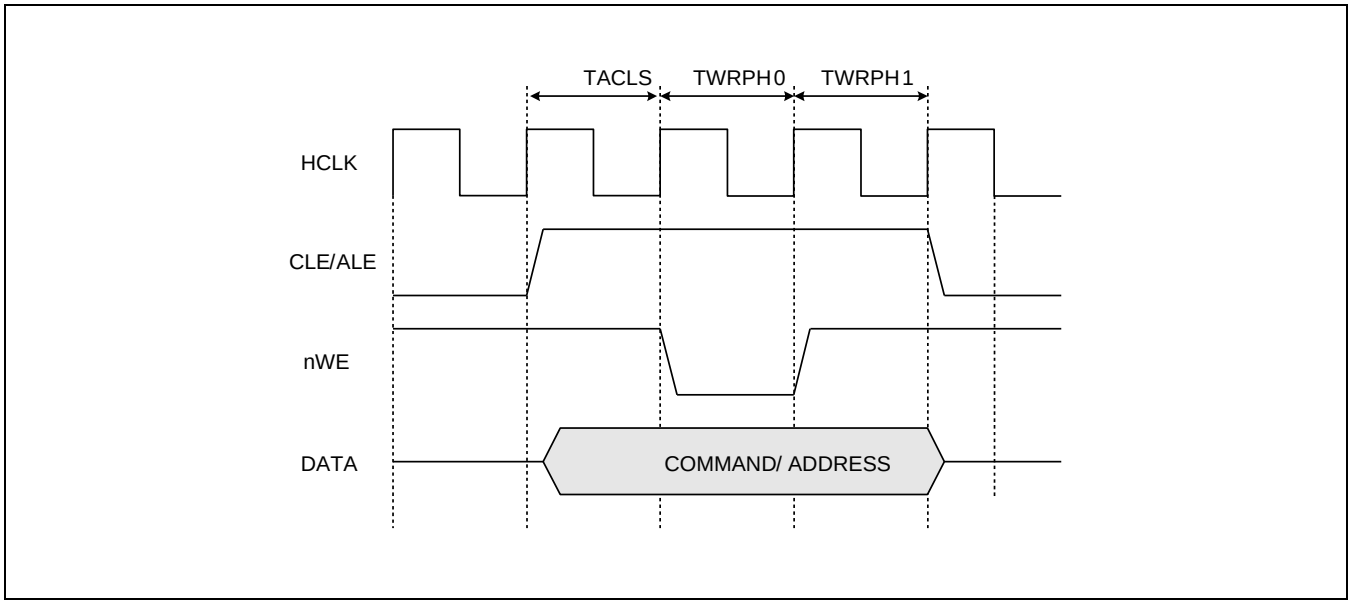


Figure 20-2 CLE and ALE Timing (TACLS = 1, TWRPH0 = 0, TWRPH1 = 0)

Figure 20-3 illustrates the nWE and nRE timing (TWRPH0 = 0, TWRPH1 = 0).

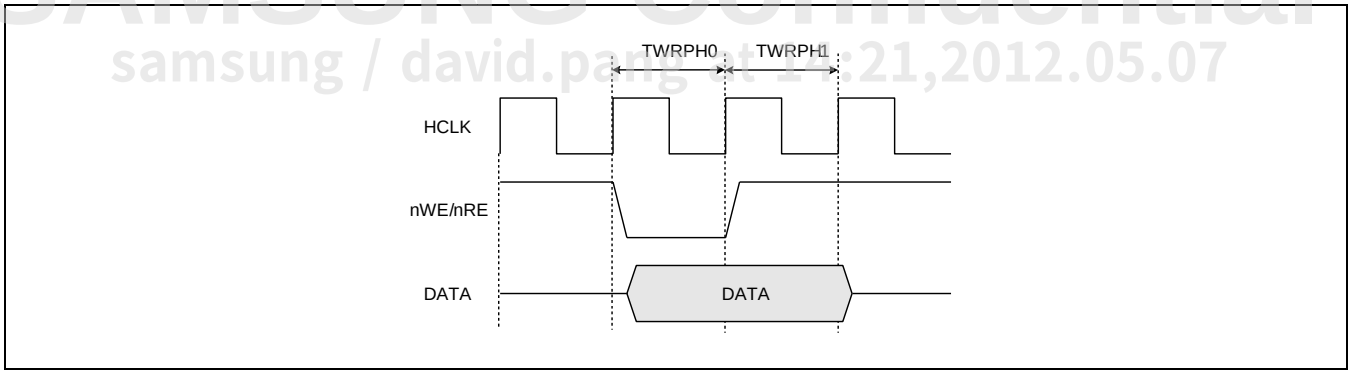


Figure 20-3 nWE and nRE Timing (TWRPH0 = 0, TWRPH1 = 0)

20.3 Software Mode

Exynos 4412 SCP supports only software mode access. Use this mode to access NAND flash memory. The NAND flash controller supports direct access to interface with the NAND flash memory.

- Writing to the command register (NFCMMD) specifies the NAND flash memory command cycle
- Writing to the address register (NFADDR) specifies the NAND flash memory address cycle
- Writing to the data register (NFDATA) specifies write data to the NAND flash memory (Write cycle)
- Reading from the data register (NFDATA) specifies read data from the NAND flash memory (Read cycle)
- Reading main ECC registers (NFMECCD0/NFMECCD1) and Spare ECC registers (NFSECCD) specifies read data from the NAND flash memory

NOTE: In the software mode, use polling or interrupt to verify the RnB status input pin.

20.3.1 Data Register Configuration

20.3.1.1 8-bit NAND Flash Memory Interface

1. Word Access

Register	Endian	Bit[31:24]	Bit[23:16]	Bit[15:8]	Bit[7:0]
NFDATA	Little	4th I/O[7:0]	3rd I/O[7:0]	2nd I/O[7:0]	1st I/O[7:0]

2. Half-word Access

Register	Endian	Bit[31:24]	Bit[23:16]	Bit[15:8]	Bit[7:0]
NFDATA	Little	Invalid value	Invalid value	2nd I/O[7:0]	1st I/O[7:0]

3. Byte Access

Register	Endian	Bit[31:24]	Bit[23:16]	Bit[15:8]	Bit[7:0]
NFDATA	Little	Invalid value	Invalid value	Invalid value	1st I/O[7:0]

20.3.2 1/4/8/12/16-bit ECC

NAND flash controller supports 1-/4-/8-/12-/16-bit ECC.

For 1-bit ECC, NAND flash controller includes ECC modules for main and spare (meta) data. Main data ECC module generates ECC parity code for 2048 bytes (maximum) data/message length, whereas spare (meta) data ECC module generates ECC parity code for 32 bytes (maximum).

For 4-bit ECC, NAND flash controller includes an ECC module. It generates 512 or 24 bytes of ECC parity code. Set MsgLength (NFCONF[25]) to select 512 or 24 bytes message length.

For 8-/12-/16-bit ECC, NAND flash controller includes ECC modules for each ECC. You can select data/message length for main and spare (meta) data length. Usually, the length of main data is 512 bytes and the length of spare (meta) data depends on user application.

Since these ECC modules support variable length of main and spare (meta) data, you should set the ECC parity conversion codes to handle free page. Refer to *20.3.11 ECC Parity Conversion Code Guide for 8/12/16-bit ECC* for more information on ECC parity conversion codes. Free page specifies an erased page. The value of erased page is "0xff". Therefore, set the ECC parity conversion codes to generate "0xff" ECC parity codes for all '0xff' data. This setting allows ECC module to detect errors on a free page.

ECC parity codes are:

- 28-bit ECC Parity Code = 22-bit Line parity + 6-bit Column Parity
- 10-bit ECC Parity Code = 4-bit Line parity + 6-bit Column Parity

Each 1-/4-/8-/12-/16-bit ECC module guarantees up to 1-/4-/8-/12-/16-bit errors, respectively. If the errors cross the number of guaranteed errors, it cannot guarantee the result.

20.3.3 2048 Byte 1-bit ECC Parity Code Assignment Table and *20.3.4 32 Byte 1-bit ECC Parity Code Assignment Table* describes 1-bit ECC parity code assignment.

20.3.3 2048 Byte 1-bit ECC Parity Code Assignment Table

	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
MECCn_0	– P64	– P64'	– P32	– P32'	– P16	– P16’	– P8	– P8'
MECCn_1	– P1024	– P1024'	– P512	– P512'	– P256	– P256'	– P128	– P128'
MECCn_2	– P4	– P4’	– P2	– P2'	– P1	– P1'	– P2048	– P2048'
MECCn_3	1	1	1	1	– P8192	– P8192’	– P4096	– P4096'

20.3.4 32 Byte 1-bit ECC Parity Code Assignment Table

	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
SECCn_0	– P2	– P2'	– P1	– P1'	– P16	– P16'	– P8	– P8'
SECCn_1	– P128	– P128'	– P64	– P64'	– P32	– P32'	– P4	– P4'

20.3.5 1-bit ECC Module Features

The ECC Lock (MainECCLock and SpareECCLock) bit of the control register generates the 1-bit ECC. If ECCLock is low, the hardware ECC modules generate the ECC codes.

1-bit ECC Register Configuration

The NAND Flash Memory interface table describes the configuration of 1-bit ECC value read from spare area of external NAND flash memory. The format of ECC read from memory is important to compare the ECC parity code that the hardware modules generate.

NOTE: 4-bit/8-bit/12-bit/16-bit ECC decoding scheme is different compared to 1-bit ECC.

NAND Flash Memory Interface

Register	Bit[31:24]	Bit[23:16]	Bit[15:8]	Bit[7:0]
NFMECCD0	Not used	2nd ECC	Not used	1st ECC
NFMECCD1	Not used	4th ECC	Not used	3rd ECC
NFSECCD	Not used	2nd ECC	Not used	1st ECC

20.3.6 1-bit ECC Programming Guide

1. To use SLC ECC in software mode, reset the ECCType to "0" (enable SLC ECC). ECC module generates ECC parity code for all Read/Write data when MainECCLock (NFCON[7]) and SpareECCLock (NFCON[6]) are unlocked ("0"). You should reset ECC value. To reset ECC value, write the InitMECC (NFCONT[5]) and InitSECC (NFCON[4]) bit as "1" and clear the MainECCLock (NFCONT[7]) bit to "0" (Unlock) before Reading or Writing data. MainECCLock (NFCONT[7]) and SpareECCLock (NFCONT[6]) bits control whether it generates ECC parity code or not.
2. The ECC module generates ECC parity code on register NFMECC0/1 whenever it Reads or Writes data.
3. After you complete Reading or Writing one page (excluding spare area data), set the MainECCLock bit to "1" (Lock). It locks ECC parity code and the value of the ECC status register does not change.
4. To generate spare area ECC parity code, clear SpareECCLock (NFCONT[6]) bit as "0" (Unlock).
5. The spare area ECC module generates ECC parity code on register NFSECC whenever it Reads or Writes data.
6. After you complete Reading or Writing spare area, set the SpareECCLock bit to "1" (Lock). It locks ECC parity code and it does not change the value of the ECC status register.
7. From now on, you can use these values to record to the spare area or verify the bit error.
8. For example, to verify the bit error of main data area on page Read operation, you should move the ECC parity codes (stored in spare area) to NFMECCD0 and NFMECCD1 after it generates ECC codes for main data area. From this point, the NFECCERR0 and NFECCERR1 have the valid error status values.

NOTE: NFSECCD is for ECC in the spare area. The main data area generates the spare area. (Usually, the user writes the ECC value generated from main data area to spare area. The value is similar to NFMECC0/1).

20.3.7 4-bit ECC Programming Guide (ENCODING)

1. To use 4-bit ECC in software mode, set the MsgLength to 0 (512 byte message length) and the ECCType to "1" (enable 4-bit ECC). ECC module generates ECC parity code for 512 byte read data. Therefore, to reset ECC value write the InitMECC (NFCONT[5]) bit as "1" and clear the MainECCLock (NFCONT[7]) bit to "0" (Unlock) before reading data.
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.

2. Whenever it writes data, the 4-bit ECC module generates ECC parity code internally.

3. After you complete writing 512 byte data (excluding spare area data) it updates the parity codes automatically to NFMECC0 and NFMECC1 registers. If you use 512 byte NAND Flash memory, you can program these values to spare area. However, if you use NAND Flash memory more than 512 byte page, you cannot program immediately. In this case, you have to copy these parity codes to other memory like DRAM. After writing all main data, you can write the copied ECC values to spare area.
   The parity codes have self-correctable information including parity code itself.

4. To generate spare area ECC parity code, set the MsgLength to "1" (24 byte message length) and the ECC Type to "1" (enable 4-bit ECC). ECC module generates ECC parity code for 24 byte write data. To reset ECC value write the InitMECC (NFCONT[5]) bit as "1" and clear the MainECCLock (NFCONT[7]) bit to "0" (unlock) before writing data. MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.

5. Whenever it writes data, the 4-bit ECC module generates ECC parity code internally.

6. When you complete writing 24 byte meta or extra data, it automatically updates the parity codes to NFMECC0 and NFMECC1 registers. You can program these parity codes to spare area.
   The parity codes have self-correctable information including parity code itself.

20.3.8 4-bit ECC Programming Guide (DECODING)

1. To use 4-bit ECC in software mode, set the MsgLength to "0" (512 byte message length) and the ECCType to "1" (enable 4-bit ECC). ECC module generates ECC parity code for 512 byte read data. Therefore, to reset ECC value, write the InitMECC (NFCONT[5]) bit as "1" and clear the MainECCLock (NFCONT[7]) bit to "0" (Unlock) before reading data.
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.

2. Whenever data is read, the 4-bit ECC module generates ECC parity code internally.

3. After you complete reading 512 byte (excluding spare area data), you should read parity codes. MLC ECC module needs parity codes to detect whether error bits have occurred or not. Therefore, you should read ECC parity code immediately after reading 512 byte. After reading ECC parity code, 4-bit ECC engine starts searching for error internally. 4-bit ECC error searching engine requires minimum of 155 cycles to find any error. During this time, you can continue reading main data from external NAND flash memory.
   Use ECCDecDone (NFSTAT[6]) to verify whether ECC decoding is completed or not.

4. When ECCDecDone (NFSTAT[6]) is set to "1", NFECCERR0 indicates whether error bit exists or not. If any error exists, refer NFECCERR0/1 and NFMLCBITPT registers to fix.

5. If you have more main data to Read, repeat step 1.

6. To verify meta data error, set the MsgLength to 1 (24-byte message length) and the ECCType to "1" (Enable 4-bit ECC). ECC module generates ECC parity code for 24-byte read data. Therefore, you must reset ECC value by writing the InitSECC (NFCONT[4]) bit as "1" and clear the SpareECCLock (NFCONT[6]) bit to"0" (Unlock) before reading data.
   SpareECCLock (NFCONT[6]) bit controls whether ECC Parity code is generated or not.

7. Whenever data is read, the 4-bit ECC module generates ECC parity code internally.

8. After you complete reading 24 byte (excluding spare area data), you should read parity codes. 4-bit ECC module needs parity codes to detect whether error bits have occurred or not. Therefore, ensure to read ECC parity codes immediately after reading 24 byte. After ECC parity code is read, 4-bit ECC engine starts searching for error internally. to verify whether ECC decoding is completed or not.

9. When ECCDecDone (NFSTAT[6]) is set ("1"), NFECCERR0 indicates whether error bit exists or not. If any error exists, you can fix it by referring to NFECCERR0/1 and NFMLCBITPT registers.

20.3.9 8/12/16-bit ECC Programming Guide (ENCODING)

1. To use 8/12/16-bit ECC in software mode, set the MsgLength (NFECCCONF[25:16]) to 511 (512 byte message length) and the ECCType to "001/100/101" (enable 8/12/16-bit ECC, respectively). ECC module generates ECC parity code for 512 byte write data. Therefore, reset ECC value by writing the InitMECC (NFECCCONT[2]) bit as "1" before writing data and clear the MainECCLock (NFCONT[7]) bit to "0" (unlock) before writing data.

2. Whenever data is written, the corresponding 8/12/16-bit ECC module generates ECC parity code internally.

3. After you complete writing 512 byte data (excluding spare area data), the parity codes are automatically updated to the NFECCPRG0 – NFECCPRGECC6 registers. If you use a NAND flash memory that contains 512 byte page, you can program these values to spare area. However, if you use a NAND flash memory more than 512 byte page, you cannot program immediately. In this case, you should copy these ECC parity codes to other memory like DRAM. After writing all main data, you can write the copied ECC values to spare area.

   The parity codes have self-correctable information including parity code itself.

   The following table describes the ECC parity size.

ECC Type	Size of ECC Parity Codes
8-bit ECC	13 byte
12-bit ECC	20 byte
16-bit ECC	26 byte

4. To generate spare area ECC parity code for meta data, the steps are similar (from 1 – 3), except setting the MsgLenght (NFECCCONF[25:16]) to the size that you prefer. When you set InitMECC (NFECCCONT[2]), all ECC parity codes generated for main data are cleared. Therefore, you should copy the ECC parity codes for main data.

NOTE: You should set the ECC parity conversion codes to verify free page error. Refer to *20.3.11 ECC Parity Conversion Code Guide for 8/12/16-bit ECC* for more information.

20.3.10 8/12/16-bit ECC Programming Guide (DECODING)

1. To use 8/12/16-bit ECC in software mode, set the MsgLength (NFECCCONF[25:16] to 511 (512 byte message length) and the ECCType to "001/100/101" (enable 8/12/16-bit ECC, respectively). ECC module generates ECC parity code for 512 byte read data. Therefore, you should reset ECC value by writing the InitMECC (NFECCCONT[2]) bit as "1" and clear the MainECCLock (NFCONT[7]) bit to "0" (unlock) before read data.

2. Whenever data is read, the 8/12/16-bit ECC module generates ECC parity code internally.

3. After you complete reading 512 byte (excluding spare area data), ensure to read the corresponding parity codes. ECC module requires parity codes to detect whether error bits have occurred or not. Therefore, you should read ECC parity code immediately after reading 512 byte. After reading the ECC parity code, the 8/12/16-bit ECC engine searches for error internally. 8/12/16-bit ECC search engine requires minimum of 155 cycles to find any errors. DecodeDone (NFECCSTAT[24]) can be used to check whether ECC decoding is completed or not.

4. When DecodeDone (NFECCSTAT[24]) is set ("1"), ECCError (NFECCSECSTAT[4:0]) indicates whether error bit exists or not. If any error exists, you can fix it by referencing NFECCERL0 to NFECCERL7 and NFECCERP0 to NFECCERP3 registers.

5. If you have additional main data to Read, repeat the steps 1 – 4.

6. To verify spare area data (meta data) error, the sequences are similar (steps 1 – 4), except setting the MsgLenght (NFECCCONF[25:16]) to the size that you want.

NOTE: You should set the ECC parity conversion codes to check free page error.
Refer to *20.3.11 ECC Parity Conversion Code Guide for 8/12/16-bit ECC*, for more information.

20.3.11 ECC Parity Conversion Code Guide for 8/12/16-bit ECC

The ECC parity conversion codes are there to fix errors, which occur when reading a free page. Free page means the erased page. The 8/12/16-bit ECC modules support variable message size for meta data stored in spare area. Generally, the size of main data (sector) is 512 byte and user should set the corresponding ECC parity conversion codes as the Table describes:

ECC Type	ECC Parity Conversion Codes
8-bit ECC	Here, 13 byte ECC parity conversion codes
12-bit ECC	Here, 20 byte ECC parity conversion codes
16-bit ECC	Here, 26 byte ECC parity conversion codes

Depending on the requirements of users, the message size for meta data stored spare area might differ. Therefore, you can change the size of meta data by changing MsgLength (NFECCCONF[25:16]) and change ECC parity conversion codes.

Steps to determine ECC parity conversion codes according to the size of message length are:

1. Clear all ECC parity conversion registers (NFECCCONECC0 to NFECCCONECC6) as all zero.
2. Set all registers for page program.
3. Reset InitMECC (NFECCCONT[2] bit as "1".
4. Write "0xff" data as much as the size of meta data.
5. After you write data as MsgLength (NFECCCONF[25:16]), the EncodeDone (NFECCSTAT[25]) is set to "1". It generates the corresponding ECC parity codes.
6. Set ECC parity conversion registers as inverted values of ECC parity codes generated. To ensure ECC parity conversion codes work properly, repeat step 3 – 5. After you set ECC parity conversion codes, if the generated ECC parity codes are all "0xff", then it is working correctly.

Constraints to support free page function are:

1. Free page check is for only data area (512 byte)
2. If there is an error during reading a page erased (free page), then free page engine indicates that the page is not free page.
3. To detect errors on free page, you should set corresponding conversion codes.

20.3.12 Lock Scheme for Data Protection

NFCON provides a lock scheme to protect data stored in external NAND flash memories from malicious program.

For this scheme, the NFSBLK and NFEBLK registers are used to provide access control methods. Only the memory area between NFSBLK and NFEBLK is erasable and programmable. However, the read access is available to entire memory area.

This lock scheme is only available when you enable LockTight (NFCONT[17]) and LOCK(NFCONT[16]).

1. **Unlock Mode**
   In unlock mode, user can access entire NAND flash memory; there are no constraints to access memory.

2. **Soft Lock Mode**
   In soft lock mode, you can access NAND block area between NFSBLK and NFEBLK.
   When you try to program or erase the locked area, an illegal access error occurs (NFSTAT [5] bit will be set).

3. **Lock-Tight Mode**
   In lock-tight mode, you can access NAND block area between NFSBLK and NFEBLK as soft lock mode. The difference is that you cannot change NFSBLK and NFEBLK registers. You cannot change LockTight (NFCONT[17]) bits also.

When you try to program or erase the locked area, an illegal access error occurs (it sets NFSTAT[5] bit).

The LockTight (NFCONT[17]) bit is only cleared when reset or wake up from sleep mode (It is impossible to clear it by software).

Figure 20-4 illustrates the accessibility of NAND area.

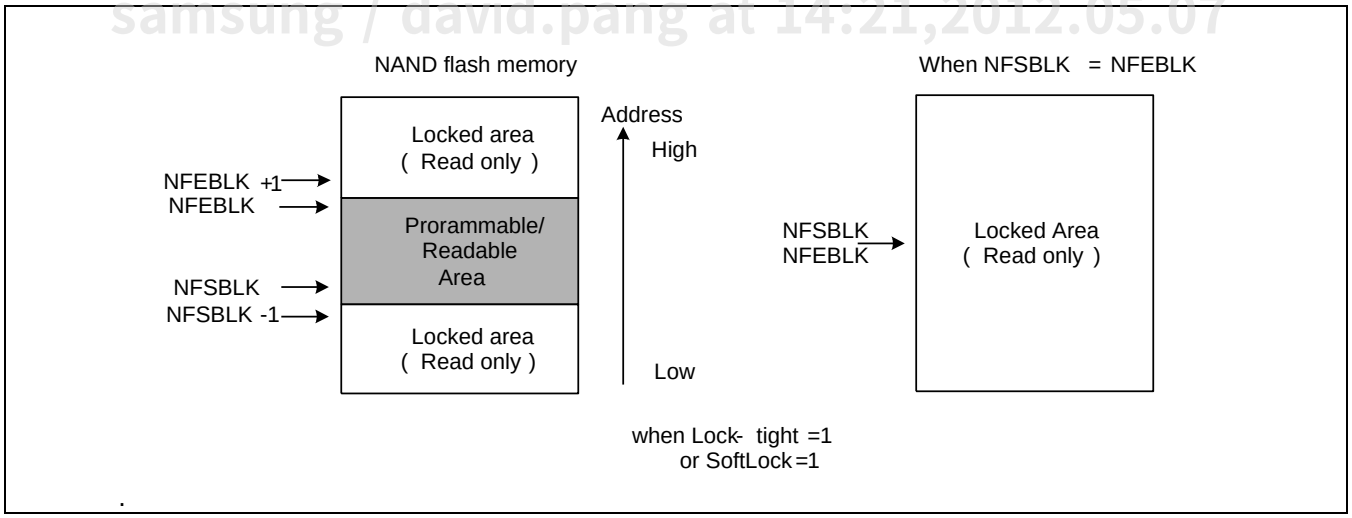


Figure 20-4 Accessibility of NAND Area

NOTE: If the address of NFSBLK and NFEBLK are similar, then it does not allow the erase and program to all NAND memory.

20.4 Programming Constraints

NFCON has a constraint to access an external NAND flash memory. NFCON accesses NAND flash memory through External Bus Interface (EBI) which uses two different clocks source. The constraint occurs because EBI operates using OneNAND external interface clock and EBI interface between NFCON and EBI is handled as asynchronous interface so that a few clock latencies consume for bus handshaking. The clock of NFCON should be set lower than EBI internal operation clock. Refer to the EBI and Clock Management Unit (CMU) manual, for more information.

20.5 I/O Description

Signal	I/O	Description	Pad	Type
Xm0DATA[7:0]	Input/Output	Address/data bus	Xm0DATA[7:0]	muxed
Xm0FRnB[3:0]	Input	Ready and busy	Xm0FRnB[3:0]	muxed
Xm0FCLE	Output	Command latch enable	Xm0FCLE	muxed
Xm0FALE	Output	Address latch enable	Xm0FALE	muxed
Xm0CSn[3:0]	Output	Chip enable	Xm0CSn[3:0]	muxed
Xm0REn	Output	Read enable	Xm0OEn	muxed
Xm0WEn	Output	Write enable	Xm0WEn	muxed

20.6 Register Description

20.6.1 Register Map Summary

- Base Address: 0x0CE0_0000

Register	Offset	Description	Reset Value
1/4-bit ECC Register			
NFCONF	0x0000	Configuration register	0x0000_1000
NFCONT	0x0004	Control register	0x00C1_00C6
NFCMMD	0x0008	Command register	0x0000_0000
NFADDR	0x000C	Address register	0x0000_0000
NFDATA	0x0010	Data register	0x0000_0000
NFMECCD0	0x0014	1st and 2nd main ECC data register	0x0000_0000
NFMECCD1	0x0018	3rd and 4th main ECC data register	0x0000_0000
NFSECCD	0x001C	Spare ECC read register	0xFFFF_FFFF
NFSBLK	0x0020	Programmable start block address register	0x0000_0000
NFEBLK	0x0024	Programmable end block address register	0x0000_0000
NFSTAT	0x0028	NAND status register	0xF080_0F0D
NFECCERR0	0x002C	ECC error status0 register	0x0003_FFF2
NFECCERR1	0x0030	ECC error status1 register	0x0000_0000
NFMECC0	0x0034	Generated ECC status0 register	0xFFFF_FFFF
NFMECC1	0x0038	Generated ECC status1 register	0xFFFF_FFFF
NFSECC	0x003C	Generated spare area ECC status register	0xFFFF_FFFF
NFMLCBITPT	0x0040	4-bit ECC error bit pattern register	0x0000_0000

- Base Address: 0x0CE2_0000

Register	Offset	Description	Reset Value
8/12/16-bit ECC Register			
NFECCCONF	0000	ECC configuration register	0x0000_0000
NFECCCONT	0020	ECC control register	0x0000_0000
NFECCSTAT	0030	ECC status register	0x0000_0000
NFECCSECSTAT	0040	ECC sector status register	0x0000_0000
NFECCPRGECC0	0090	ECC parity code0 register for page program	0x0000_0000
NFECCPRGECC1	0094	ECC parity code1 register for page program	0x0000_0000
NFECCPRGECC2	0098	ECC parity code2 register for page program	0x0000_0000
NFECCPRGECC3	009C	ECC parity code3 register for page program	0x0000_0000
NFECCPRGECC4	00A0	ECC parity code4 register for page program	0x0000_0000
NFECCPRGECC5	00A4	ECC parity code5 register for page program	0x0000_0000
NFECCPRGECC6	00A8	ECC parity code6 register for page program	0x0000_0000
NFECCERL0	00C0	ECC error byte location0 register	0x0000_0000
NFECCERL1	00C4	ECC error byte location1 register	0x0000_0000
NFECCERL2	00C8	ECC error byte location2 register	0x0000_0000
NFECCERL3	00CC	ECC error byte location3 register	0x0000_0000
NFECCERL4	00D0	ECC error byte location4 register	0x0000_0000
NFECCERL5	00D4	ECC error byte location5 register	0x0000_0000
NFECCERL6	00D8	ECC error byte location6 register	0x0000_0000
NFECCERL7	00DC	ECC error byte location7 register	0x0000_0000
NFECCERP0	00F0	ECC error bit pattern0 register	0x0000_0000
NFECCERP1	00F4	ECC error bit pattern1 register	0x0000_0000
NFECCERP2	00F8	ECC error bit pattern2 register	0x0000_0000
NFECCERP3	00FC	ECC error bit pattern3 register	0x0000_0000
NFECCCONECC0	0110	ECC parity conversion code0 register	0x0000_0000
NFECCCONECC1	0114	ECC parity conversion code1 register	0x0000_0000
NFECCCONECC2	0118	ECC parity conversion code2 register	0x0000_0000
NFECCCONECC3	011C	ECC parity conversion code3 register	0x0000_0000
NFECCCONECC4	0120	ECC parity conversion code4 register	0x0000_0000
NFECCCONECC5	0124	ECC parity conversion code5 register	0x0000_0000
NFECCCONECC6	0128	ECC parity conversion code6 register	0x0000_0000

20.6.2 NAND Flash Interface and 1/4-bit ECC Registers

20.6.2.1 NFCONF

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_1000

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	0
MsgLength	[25]	RW	0 = 512 byte message length 1 = 24 byte message length	0
ECCType0	[24:23]	RW	This bit indicates the type of ECC to use. 00 = 1-bit ECC 10 = 4-bit ECC 01 = 11 = Disables 1-bit and 4-bit ECC	0
RSVD	[22:16]	–	Reserved	0000000
TACLS	[15:12]	RW	CLE and ALE duration setting value (0 – 15) • Duration = HCLK × TACLS	0x1
TWRPH0	[11:8]	RW	TWRPH0 duration setting value (0 – 15) • Duration = HCLK × (TWRPH0 + 1) NOTE: You should add additional cycles about 10ns for page read because of additional signal delay on PCB pattern.	0x0
TWRPH1	[7:4]	RW	TWRPH1 duration setting value (0 – 15) • Duration = HCLK × (TWRPH1 + 1)	0x0
PageSize	[3:2]	RW	This bit indicates the page size of NAND flash memory 00 = 2048 byte 01 = 512 byte 10 = 4096 byte 11 = 2048 byte NOTE: Using 1-bit ECC it determines the message length. It does not determine the message length using MsgLength (NFCONF[25] field. NFCON does not consider the actual page size of external NAND. Software handles the page size.	0
AddrCycle	[1]	RW	This bit indicates the number of address cycle of NAND flash memory. When Page Size is 512 Bytes: 0 = 3 Address cycle 1 = 4 Address cycle When page size is 2 K or 4 K: 0 = 4 Address cycle 1 = 5 Address cycle NOTE: It is only used for Lock scheme. Refer to section *20.3.12 Lock Scheme for Data Protection*, for more information.	0
RSVD	[0]	–	Reserved	0

20.6.2.2 NFCONT

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0004, Reset Value = 0x00C1_00C6

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0
Reg_nCE3	[23]	RW	NAND flash memory nRCS[3] signal control 0 = Force nRCS[3] to low (Enables chip select) 1 = Force nRCS[3] to high (Disables chip select)	1
Reg_nCE2	[22]	RW	NAND flash memory nRCS[2] signal control 0 = Force nRCS[2] to low (Enables chip select) 1 = Force nRCS[2] to high (Disables chip select)	1
RSVD	[21:19]	–	Reserved	0
MLCEccDirection	[18]	RW	4-bit, ECC encoding/decoding control 0 = Decoding 4-bit ECC. It is used for page read 1 = Encoding 4-bit ECC. It is used for page program	0
LockTight	[17]	RW	Lock-tight configuration 0 = Disables lock-tight 1 = Enables lock-tight If this bit is set to 1, you cannot clear this bit. Refer to *20.3.12 Lock Scheme for Data Protection*, for more information.	0
LOCK	[16]	RW	Soft Lock configuration 0 = Disables lock 1 = Enables lock Software can modify soft lock area any time. Refer to *20.3.12 Lock Scheme for Data Protection*, for more information.	1
RSVD	[15:14]	–	Reserved	00
EnbMLCEncInt	[13]	RW	4-bit ECC encoding completion interrupt control 0 = Disables interrupt 1 = Enables interrupt	0
EnbMLCDecInt	[12]	RW	4-bit ECC decoding completion interrupt control 0 = Disables interrupt 1 = Enables interrupt	0
RSVD	[11]	–	Reserved	0
EnbIllegalAccINT	[10]	RW	Illegal access interrupt control 0 = Disables interrupt 1 = Enables interrupt Illegal access interrupt occurs when CPU tries to program or erase locking area (the area setting in NFSBLK (0xB0E0_0020) to NFEBLK (0xB0E0_0024) – 1.	0
EnbRnBINT	[9]	RW	RnB status input signal transition interrupt control 0 = Disables RnB interrupt	0

Name	Bit	Type	Description	Reset Value
			1 = Enables RnB interrupt	
RnB_TransMode	[8]	RW	RnB transition detection configuration 0 = Detects rising edge 1 = Detects falling edge	0
MECCLock	[7]	RW	Lock Main area ECC generation 0 = Unlocks Main area ECC 1 = Locks Main area ECC Main area ECC status register is NFMECC0/NFMECC1 (0xB0E0_0034/0xB0E0_0038),	1
SECCLock	[6]	RW	Lock Spare area ECC generation 0 = Unlocks Spare ECC 1 = Locks Spare ECC Spare area ECC status register is NFSECC (0xB0E0_003C)	1
InitMECC	[5]	RW	1 = Initializes main area ECC decoder/encoder (Write-only)	0
InitSECC	[4]	RW	1 = Initializes spare area ECC decoder/encoder (Write-only)	0
HW_nCE	[3]	RW	Reserved (HW_nCE)	0
Reg_nCE1	[2]	RW	NAND flash memory nRCS[1] signal control	1
Reg_nCE0	[1]	RW	NAND flash memory nRCS[0] signal control 0 = Force nRCS[0] to low (enables chip select) 1 = Force nRCS[0] to high (disables chip select) NOTE: The setting all nCE[3:0] zero can not be allowed. Only one nCE can be asserted to enable external NAND flash memory. The lower bit has more priority when user set all nCE[3:0] zeros.	1
MODE	[0]	RW	NAND flash controller operating mode 0 = Disables NAND flash controller 1 = Enables NAND flash controller	0

20.6.2.3 NFCMMD

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
REG_CMMD	[7:0]	RW	NAND flash memory command value	0x00

20.6.2.4 NFADDR

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x000000
REG_ADDR	[7:0]	RW	NAND flash memory address value	0x00

20.6.2.5 NFDATA

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
NFDATA	[31:0]	RW	NAND flash Read/program data value for I/O NOTE: Refer to *20.3.1 Data Register Configuration*, for more information.	0x00000000

20.6.2.6 NFMECCD

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
ECCData1 (ECC1)	[23:16]	RW	2nd ECC NOTE: In software mode, read this register when you need to read 2nd ECC value from NAND flash memory	0x00
RSVD	[15:8]	–	Reserved	0x00
ECCData0 (ECC0)	[7:0]	RW	1st ECC. NOTE: In software mode, read this register when you need to read 1st ECC value from NAND flash memory. This register has the similar Read function as NFDATA.	0x00

NOTE: It allows only word access.

20.6.2.7 NFMECCD1

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
ECCData3 (ECC3)	[23:16]	RW	4th ECC. NOTE: In software mode, read this register when you need to read 4th ECC value from NAND flash memory	0x00
RSVD	[15:8]	–	Reserved	0x00
ECCData2 (ECC2)	[7:0]	RW	3rd ECC. NOTE: In software mode, read this register when you need to read 3rd ECC value from NAND flash memory. This register has the similar Read function as NFDATA.	0x00

20.6.2.8 NFSECCD

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x001C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
SECCData1	[23:16]	RW	2nd ECC. NOTE: In software mode, read this register when you need to read 2nd ECC value from NAND flash memory	0xFF
RSVD	[15:8]	–	Reserved	0x00
SECCData0	[7:0]	RW	1st ECC. NOTE: In software mode, read this register when you need to read 1st ECC value from NAND flash memory. This register has the similar Read function as NFDATA.	0xFF

NOTE: It allows only word access.

20.6.2.9 NFSBLK

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
SBLK_ADDR2	[23:16]	RW	The 3rd block address of the block erase operation	0x00
SBLK_ADDR1	[15:8]	RW	The 2nd block address of the block erase operation	0x00
SBLK_ADDR0	[7:0]	RW	The 1st block address of the block erase operation (Only bit[7:5] are valid).	0x00

NOTE: Address of Advance Flash block starts from 3-address cycle. So block address register only requires 3-bytes. Refer to *20.3.12 Lock Scheme for Data Protection*, for more information on lock scheme.

20.6.2.10 NFEBLK

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
EBLK_ADDR2	[23:16]	RW	The 3rd block address of the block erase operation	0x00
EBLK_ADDR1	[15:8]	RW	The 2nd block address of the block erase operation	0x00
EBLK_ADDR0	[7:0]	RW	The 1st block address of the block erase operation (Only bit[7:5] are valid)	0x00

NOTE: Address of Advance Flash block starts from 3-address cycle. So block address register only requires 3-bytes. Refer to *20.3.12 Lock Scheme for Data Protection* for more information on lock scheme.

20.6.2.11 NFSTAT

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0028, Reset Value = 0xF080_0F0D

Name	Bit	Type	Description	Reset Value
Flash_RnB_GRP	[31:28]	RW	The status of RnB[3:0] input pin 0 = NAND flash memory busy 1 = NAND flash memory ready to operate	0xF
RnB_TransDetect_GRP	[27:24]	RW	When RnB[3:0] low to high transition occurs, this bit is set and an interrupt is issued if RnB_TransDetect_GRP is enabled. To clear this, write "1". 0 = RnB transition is not detected 1 = RnB transition is detected Transition configuration is set in RnB_TransMode(NFCONT[8]).	–
RSVD	[23:12]	–	Reserved	0x800
Flash_nCE[3:0] (Read-only)	[11:8]	RW	The status of nCE[3:0] output pin	0xF
MLCEncodeDone	[7]	RW	When it completes 4-bit ECC encoding, this bit is set and it issues an interrupt if it enables MLCEncodeDone. The NFMLCECC0 and NFMLCECC1 have valid values. To clear this, write "1". 1 = It completes 4-bit ECC encoding	0
MLCDecodeDone	[6]	RW	When it completes 4-bit ECC decoding, this bit is set and it issues an interrupt if it enables MLCDecodeDone. The NFMLCBITPT, NFMLCL0, and NFMLCEL1 have valid values. To clear this, write "1". 1 = Completes 4-bit ECC decoding	0
IllegalAccess	[5]	RW	Once Soft Lock or Lock-tight is enabled and any illegal access (program, erase) to the memory takes place, then this bit is set. 0 = It does not detect illegal access 1 = It detects illegal access To clear this value, write 1 to this bit.	0
RnB_TransDetect	[4]	RW	When RnB[0] low to high transition occurs, this bit is set and an interrupt is issued if RnB_TransDetect is enabled. To clear this, write "1". 0 = It does not detect RnB transition 1 = It detects RnB transition Transition configuration is set in RnB_TransMode(NFCONT[8]).	0
Flash_nCE[1] (Read-only)	[3]	RW	The status of nCE[1] output pin	1
Flash_nCE[0] (Read-only)	[2]	RW	The status of nCE[0] output pin	1
RSVD	[1]	–	Reserved	0

Name	Bit	Type	Description	Reset Value
Flash_RnB (Read-only)	[0]	RW	The status of RnB[0] input pin 0 = NAND flash memory busy 1 = NAND flash memory ready to operate	1

20.6.2.12 NFECCERR0

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x002C, Reset Value = 0x0003_FFF2

When ECC Type is 1-bit ECC

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x00
ECCSDataAddr	[24:21]	R	In spare area, indicates which number data is error	0x0
ECCSBitAddr	[20:18]	R	In spare area, indicates which bit is error	000
ECCDataAddr	[17:7]	R	In main data area, indicates which number data is error	0x7FF
ECCBitAddr	[6:4]	R	In main data area, indicates which bit is error	111
ECCSprErrNo	[3:2]	R	Indicates whether spare area bit fail error occurred 00 = No Error 01 = 1-bit error (correctable) 10 = Multiple error 11 = ECC area error	00
ECCMainErrNo	[1:0]	R	Indicates whether main data area bit fail error occurred 00 = No Error 01 = 1-bit error (Correctable) 10 = Multiple error 11 = ECC area error	10

NOTE: The above values are valid only when both ECC register and ECC status register have valid value.

When ECC Type is 4-bit ECC

Name	Bit	Type	Description	Reset Value
MLCECCBusy	[31]	R	Indicates the 4-bit ECC decoding engine is searching whether a error exists or not 0 = Idle 1 = Busy	0
MLCECCReady	[30]	R	ECC Ready bit	1
MLCFreePage	[29]	R	Indicates the page data read from NAND flash has all "FF" value.	0
MLCECCError	[28:26]	R	4-bit ECC decoding result 000 = No error 001 = 1-bit error 010 = 2-bit error 011 = 3-bit error 100 = 4-bit error 101 = Uncorrectable 11x = Reserved	000
MLCErrLocation2	[25:16]	R	Error byte location of 2nd bit error	0x000
RSVD	[15:10]	–	Reserved	0x00
MLCErrLocation1	[9:0]	R	Error byte location of 1st bit error	0x000

NOTE: These values are updated when ECCDecodeDone (NFSTAT[6]) is set ("1").

20.6.2.13 NFECCERR1

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

When ECC Type is 4-bit ECC

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	0x00
MLCErrLocation4	[25:16]	R	Error byte location of 4th bit error	0x00
RSVD	[15:10]	–	Reserved	0x00
MLCErrLocation3	[9:0]	R	Error byte location of 3rd bit error	0x000

NOTE: These values are updated when ECCDecodeDone (NFSTAT[6]) is set ("1").

20.6.2.14 NFMECC0

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0034, Reset Value = 0xFFFF_FFFF

When ECC Type is 1-bit ECC

Name	Bit	Type	Description	Reset Value
MECC3	[31:24]	R	ECC3 for data	0xFF
MECC2	[23:16]	R	ECC2 for data	0xFF
MECC1	[15:8]	R	ECC1 for data	0xFF
MECC0	[7:0]	R	ECC0 for data	0xFF

NOTE: The NAND flash controller generate NFMECC0/1 when read or write main area data while the MainECCLock (NFCONT[7]) bit is "0" (Unlock).

When ECC Type is 4-bit ECC

Name	Bit	Type	Description	Reset Value
4th Parity	[31:24]	R	4th Check parity generated from main area (512 byte)	0x00
3rd Parity	[23:16]	R	3rd Check parity generated from main area (512 byte)	0x00
2nd Parity	[15:8]	R	2nd Check parity generated from main area (512 byte)	0x00
1st Parity	[7:0]	R	1st Check parity generated from main area (512 byte)	0x00

NOTE: The NAND flash controller generates these ECC parity codes when write main area data while the MainECCLock (NFCON[7]) bit is "0" (unlock).

20.6.2.15 NFMECC1

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0038, Reset Value = 0xFFFF_FFFF

When ECC Type is 4-bit ECC

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
7th Parity	[23:16]	R	7th Check parity generated from main area (512 byte)	0x00
6th Parity	[15:8]	R	6th Check parity generated from main area (512 byte)	0x00
5th Parity	[7:0]	R	5th Check parity generated from main area (512 byte)	0x00

NOTE: The NAND flash controller generates these ECC parity codes when write main area data while the MainECCLock (NFCON[7]) bit is "0" (unlock).

20.6.2.16 NFSECC

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x003C, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0xFFFF
SECC1	[15:8]	R	Spare area ECC1 Status	0xFF
SECC0	[7:0]	R	Spare area ECC0 Status	0xFF

NOTE: The NAND flash controller generates NFSECC when Read or Write spare area data while the SpareECCLock (NFCONT[6]) bit is "0" (unlock).

20.6.2.17 NFMLCBITPT

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
4th Error bit pattern	[31:24]	R	4th Error bit pattern	0x00
3rd Error bit pattern	[23:16]	R	3rd Error bit pattern	0x00
2nd Error bit pattern	[15:8]	R	2nd Error bit pattern	0x00
1st Error bit pattern	[7:0]	R	1st Error bit pattern	0x00

20.6.3 ECC Registers for 8, 12 and 16-bit ECC

20.6.3.1 NFECCCONF

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
RSVD	[28]	–	Reserved	0
MsgLength	[25:16]	RW	The ECC message size For 512 byte message, you should set to 511.	
RSVD	[15:4]	–	Reserved	0
ECCType	[3:0]	RW	These bits indicate what type of ECC is used. 000 = Disables 8/12/16-bit ECC 001 = Reserved 010 = Reserved 011 = 8-bit ECC/512B 100 = 12-bit ECC 101 = 16-bit ECC/512B 110 = Reserved 111 = Reserved	0x0

20.6.3.2 NFECCCONT

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	0x00
EnbMLCEncInt	[25]	RW	MLC ECC encoding completion interrupt control 0 = Disables interrupt 1 = Enables interrupt	0
EnbMLCDecInt	[24]	RW	MLC ECC decoding completion interrupt control 0 = Disables interrupt 1 = Enables interrupt	0
EccDirection	[16]	RW	MLC ECC encoding/decoding control 0 = Decoding, used for page read 1 = Encoding, used for page program	0
RSVD	[15:3]	–	Reserved	0x0
InitMECC	[2]	RW	1 = Initialize main area ECC decoder/encoder (Write-only)	0
RSVD	[1]	–	Reserved	0
ResetECC	[0]	RW	1 = Reset ECC logic (Write-only)	0

20.6.3.3 NFECCSTAT

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ECCBusy	[31]	R	Indicates the 8-bit ECC decoding engine is searching whether a error exists or not 0 = Idle 1 = Busy	0
RSVD	[30]	–	Reserved	1
EncodeDone	[25]	RWX	When MLC ECC encoding is finished, this value set and issue interrupt if EncodeDone is enabled. The NFMLCECC0 and NFMLCECC1 have valid values. To clear this, write "1". 1 = It completes MLC ECC encoding	0
DecodeDone	[24]	RWX	When MLC ECC decoding is finished, this value set and issue interrupt if DecodeDone is enabled. The NFMLCBITPT, NFMLCL0, and NFMLCEL1 have valid values. To clear this, write "1". 1 = It completes MLC ECC decoding	0
RSVD	[23:9]	–	Reserved	0x0000
FreePageStat	[8]	R	It indicates whether the sector is free page or not.	0
RSVD	[7:0]	–	Reserved	0x00

20.6.3.4 NFECCSECSTAT

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ValdErrorStat	[31:8]	R	Each bit indicates which ERL and ERP are valid.	0x0000_00
ECCErrorNo	[4:0]	R	ECC decoding result when page read 00000 = No error 00001 = 1-bit error 00010 = 2-bit error 00011 = 3-bit error .... 01110 = 14-bit error 01111 = 15-bit error 10000 = 16-bit error NOTE: If it uses 8-bit ECC, the valid number of error is until 8. If the number exceeds the supported error number, it means that uncorrectable error occurs.	0x00

20.6.3.5 NFECCPRGECCn (n = 0 to 6)

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000_0000
- Address = Base Address + 0x0094, Reset Value = 0x0000_0000
- Address = Base Address + 0x0098, Reset Value = 0x0000_0000
- Address = Base Address + 0x009C, Reset Value = 0x0000_0000
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000
- Address = Base Address + 0x00A4, Reset Value = 0x0000_0000
- Address = Base Address + 0x00A8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
4th Parity	[31:24]	R	4th check parity for page program from main area	0x00
3rd Parity	[23:16]	R	3rd check parity for page program from main area	0x00
2nd Parity	[15:8]	R	2nd check parity for page program from main area	0x00
1st Parity	[7:0]	R	1st check parity for page program from main area	0x00
8th Parity	[31:24]	R	8th check parity generated from main area	0x00
7th Parity	[23:16]	R	7th check parity generated from main area	0x00
6th Parity	[15:8]	R	6th check parity generated from main area	0x00
5th Parity	[7:0]	R	5th check parity generated from main area	0x00
12th Parity	[31:24]	R	12th check parity generated from main area	0x00
11th Parity	[23:16]	R	11th check parity generated from main area	0x00
10th Parity	[15:8]	R	10th check parity generated from main area	0x00
9th Parity	[7:0]	R	9th check parity generated from main area	0x00
16th Parity	[31:24]	R	16th check parity generated from main area	0x00
15th Parity	[23:16]	R	15th check parity generated from main area	0x00
14th Parity	[15:8]	R	14th check parity generated from main area	0x00
13th Parity	[7:0]	R	13th check parity generated from main area	0x00
20th Parity	[31:24]	R	20th check parity generated from main area	0x00
19th Parity	[23:16]	R	19th check parity generated from main area	0x00
18th Parity	[15:8]	R	18th check parity generated from main area	0x00
17th Parity	[7:0]	R	17th check parity generated from main area	0x00
24th Parity	[31:24]	R	24th check parity generated from main area	0x00
23rd Parity	[23:16]	R	23rd check parity generated from main area	0x00
22th Parity	[15:8]	R	22th check parity generated from main area	0x00
21th Parity	[7:0]	R	21th check parity generated from main area	0x00
RSVD	[31:16]	–	Reserved	–
26th Parity	[15:8]	R	26th check parity generated from main area	0x00
25th Parity	[7:0]	R	25th check parity generated from main area	0x00

NOTE: The NAND flash controller generates these ECC parity codes when write main area data while the MainECCLock (NFCON[7]) bit is "0" (unlock).

20.6.3.6 NFECCERLn (n = 0 to 7)

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000
- Address = Base Address + 0x00C4, Reset Value = 0x0000_0000
- Address = Base Address + 0x00C8, Reset Value = 0x0000_0000
- Address = Base Address + 0x00CC, Reset Value = 0x0000_0000
- Address = Base Address + 0x00D0, Reset Value = 0x0000_0000
- Address = Base Address + 0x00D4, Reset Value = 0x0000_0000
- Address = Base Address + 0x00D8, Reset Value = 0x0000_0000
- Address = Base Address + 0x00DC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	0x0
ErrByteLoc2	[25:16]	R	Error byte location of 2nd bit error	0x000
RSVD	[15:10]	–	Reserved	0x0
ErrByteLoc1	[9:0]	R	Error byte location of 1st bit error	0x000
RSVD	[31:26]	–	Reserved	0x0
ErrByteLoc4	[25:16]	R	Error byte location of 4th bit error	0x000
RSVD	[15:10]	–	Reserved	0x0
ErrByteLoc3	[9:0]	R	Error byte location of 3rd bit error	0x000
RSVD	[31:26]	–	Reserved	0x0
ErrByteLoc6	[25:16]	R	Error byte location of 6th bit error	0x000
RSVD	[15:10]	–	Reserved	0x0
ErrByteLoc5	[9:0]	R	Error byte location of 5th bit error	0x000
RSVD	[31:26]	–	Reserved	0x0
ErrByteLoc8	[25:16]	R	Error byte location of 8th bit error	0x000
RSVD	[15:10]	–	Reserved	0x0
ErrByteLoc7	[9:0]	R	Error byte location of 7th bit error	0x000
RSVD	[31:26]	–	Reserved	0x0
ErrByteLoc10	[25:16]	R	Error byte location of 10th bit error	0x000
RSVD	[15:10]	–	Reserved	0x0
ErrByteLoc9	[9:0]	R	Error byte location of 9th bit error	0x000
RSVD	[31:26]	–	Reserved	0x0
ErrByteLoc12	[25:16]	R	Error byte location of 12th bit error	0x000
RSVD	[15:10]	–	Reserved	0x0
ErrByteLoc11	[9:0]	R	Error byte location of 11th bit error	0x000
RSVD	[31:26]	–	Reserved	0x0
ErrByteLoc14	[25:16]	R	Error byte location of 14th bit error	0x000

Name	Bit	Type	Description	Reset Value
RSVD	[15:10]	–	Reserved	0x0
ErrByteLoc13	[9:0]	R	Error byte location of 13th bit error	0x000
RSVD	[31:26]	–	Reserved	0x0
ErrByteLoc16	[25:16]	R	Error byte location of 16th bit error	0x000
RSVD	[15:10]	–	Reserved	0x0
ErrByteLoc15	[9:0]	R	Error byte location of 15th bit error	0x000

NOTE: It updates these values when DecodeDone (NFECCSTAT[24]) is set ("1").

20.6.3.7 NFECCERPn (n = 0 to 3)

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x00F0, Reset Value = 0x0000_0000
- Address = Base Address + 0x00F4, Reset Value = 0x0000_0000
- Address = Base Address + 0x00F8, Reset Value = 0x0000_0000
- Address = Base Address + 0x00FC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
4th ErrBitPattern	[31:24]	R	4th Error Bit Pattern	0x00
3rd ErrBitPattern	[23:16]	R	3rd Error bit pattern	0x00
2nd ErrBitPattern	[15:8]	R	2nd Error bit pattern	0x00
1st ErrBitPattern	[7:0]	R	1st Error bit pattern	0x00
8th ErrBitPattern	[31:24]	R	8th Error bit pattern	0x00
7th ErrBitPattern	[23:16]	R	7th Error bit pattern	0x00
6th ErrBitPattern	[15:8]	R	6th Error bit pattern	0x00
5th ErrBitPattern	[7:0]	R	5th Error bit pattern	0x00
12th ErrBitPattern	[31:24]	R	12th Error bit pattern	0x00
11th ErrBitPattern	[23:16]	R	11th Error bit pattern	0x00
10th ErrBitPattern	[15:8]	R	10th Error bit pattern	0x00
9th ErrBitPattern	[7:0]	R	9th Error bit pattern	0x00
16th ErrBitPattern	[31:24]	R	16th Error bit pattern	0x00
15th Error bit pattern	[23:16]	R	15th Error bit pattern	0x00
14th ErrBitPattern	[15:8]	R	14th Error bit pattern	0x00
13th ErrBitPattern	[7:0]	R	13th Error bit pattern	0x00

NOTE: It updates these values when DecodeDone (NFECCSTAT[25]) is set ("1").

20.6.3.8 NFECCCONECCn (n = 0 to 6)

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x0110, Reset Value = 0x0000_0000
- Address = Base Address + 0x0114, Reset Value = 0x0000_0000
- Address = Base Address + 0x0118, Reset Value = 0x0000_0000
- Address = Base Address + 0x011C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0120, Reset Value = 0x0000_0000
- Address = Base Address + 0x0124, Reset Value = 0x0000_0000
- Address = Base Address + 0x0128, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
4th Conversion Code	[31:24]	RW	4th ECC Parity Conversion Code	0x00
3rd Conversion Code	[23:16]	RW	3rd ECC Parity conversion code	0x00
2nd Conversion Code	[15:8]	RW	2nd ECC Parity conversion code	0x00
1st Conversion Code	[7:0]	RW	1st ECC Parity conversion code	0x00
8th Conversion Code	[31:24]	RW	8th ECC Parity conversion code	0x00
7th Conversion Code	[23:16]	RW	7th ECC Parity conversion code	0x00
6th Conversion Code	[15:8]	RW	6th ECC Parity conversion code	0x00
5th Conversion Code	[7:0]	RW	5th ECC Parity conversion code	0x00
12th Conversion Code	[31:24]	RW	12th ECC Parity conversion code	0x00
11th Conversion Code	[23:16]	RW	11th ECC Parity conversion code	0x00
10th Conversion Code	[15:8]	RW	10th ECC Parity conversion code	0x00
9th Conversion Code	[7:0]	RW	9th ECC Parity conversion code	0x00
16th Conversion Code	[31:24]	RW	16th ECC Parity conversion code	0x00
15th Conversion Code	[23:16]	RW	15th ECC Parity conversion code	0x00
14th Conversion Code	[15:8]	RW	14th ECC Parity conversion code	0x00
13th Conversion Code	[7:0]	RW	13th ECC Parity conversion code	0x00
20th Conversion Code	[31:24]	RW	20th ECC Parity conversion code	0x00
19th Conversion Code	[23:16]	RW	19th ECC Parity conversion code	0x00
18th Conversion Code	[15:8]	RW	18th ECC Parity conversion code	0x00
17th Conversion Code	[7:0]	RW	17th ECC Parity conversion code	0x00
24th Conversion Code	[31:24]	RW	24th ECC Parity conversion code	0x00
23th Conversion Code	[23:16]	RW	23th ECC Parity conversion code	0x00
22th Conversion Code	[15:8]	RW	22th ECC Parity conversion code	0x00
21th Conversion Code	[7:0]	RW	21th ECC Parity conversion code	0x00
RSVD	[31:16]	–	Reserved	0x0000
26th Conversion Code	[15:8]	RW	26th ECC Parity conversion code	0x00
25th Conversion Code	[7:0]	RW	25th ECC Parity conversion code	0x00

NOTE: For more information about ECC parity conversion codes, refer to *20.3.11 ECC Parity Conversion Code Guide for 8/12/16-bit ECC*.

21 External Bus Interface (EBI)

21.1 Overview of External Bus Interface

Exynos 4412 SCP uses External Bus Interface (EBI) as a peripheral: It relies on memory controller to release external requests for external bus when the memory controller is idle. This happens as it has no knowledge of when memory access will begin or complete. It enables one SROM Controller (SROMC), and one NAND Flash Controller (NFCON), to share an external memory bus, Memory Port 0.

21.2 Features of Exynos 4412 SCP EBI

The features of Exynos 4412 SCP EBI are:

Memory Port 0 is shared using EBI.

NOTE: Reference: ARM PrimeCell External Bus Interface (PL220) and ARM DDI 0249B.

- Two memory controllers (SROMC, NFCON) share the pad interface.
- Priority determines the pad interface ownership (can be changed).
- The handshaking between EBI and memory controller consists of a three-wire interface. They are: EBIREQ, EBIGNT, and EBIBACKOFF (all Active High).
- The memory controllers assert EBIREQ signals when they require external bus access.
- The arbitrated EBIGNT is issued to the memory controller with highest priority.
- The memory controller must complete current transfer and release the bus. To signal these actions, EBIBACKOFF is made as output of EBI.
- Can we break sentence into: The EBI arbitration scheme tracks memory controller that it currently grants. It waits for transaction from memory controller to complete (EBIREQ is set to Low by the memory controller) before it grants next memory controller. If a higher priority memory controller requests bus, then EBIBACKOFF signal informs memory controller that it currently grants to terminate current transfer as soon as possible.
- The System Controller provides booting method and Chip Select (CS) selection information to Memory Subsystem.
- nCS0 and nCS1 in memory port 0 are only reserved for SROMC.
- If it selects NAND Flash for boot device, then it uses nCS2 to access boot media.
- EBIGNT is required to be deasserted one cycle after EBIREQ is deasserted in sync. Mode.
- EBIBACKOFF is required to be deasserted one cycle after EBIREQ is deasserted in sync. Mode.
- In case if it deasserts EBIREQ because of higher priority EBIBACKOFF, EBIREQ signal must be set to low for at least one clock cycle in sync Mode.
- It does not require EBI_REQ duration for at least four cycles from CSYSREQ to CACTIVE

21.3 Block Diagram of Memory Interface through EBI

Figure 21-1 illustrates the memory interface through EBI.

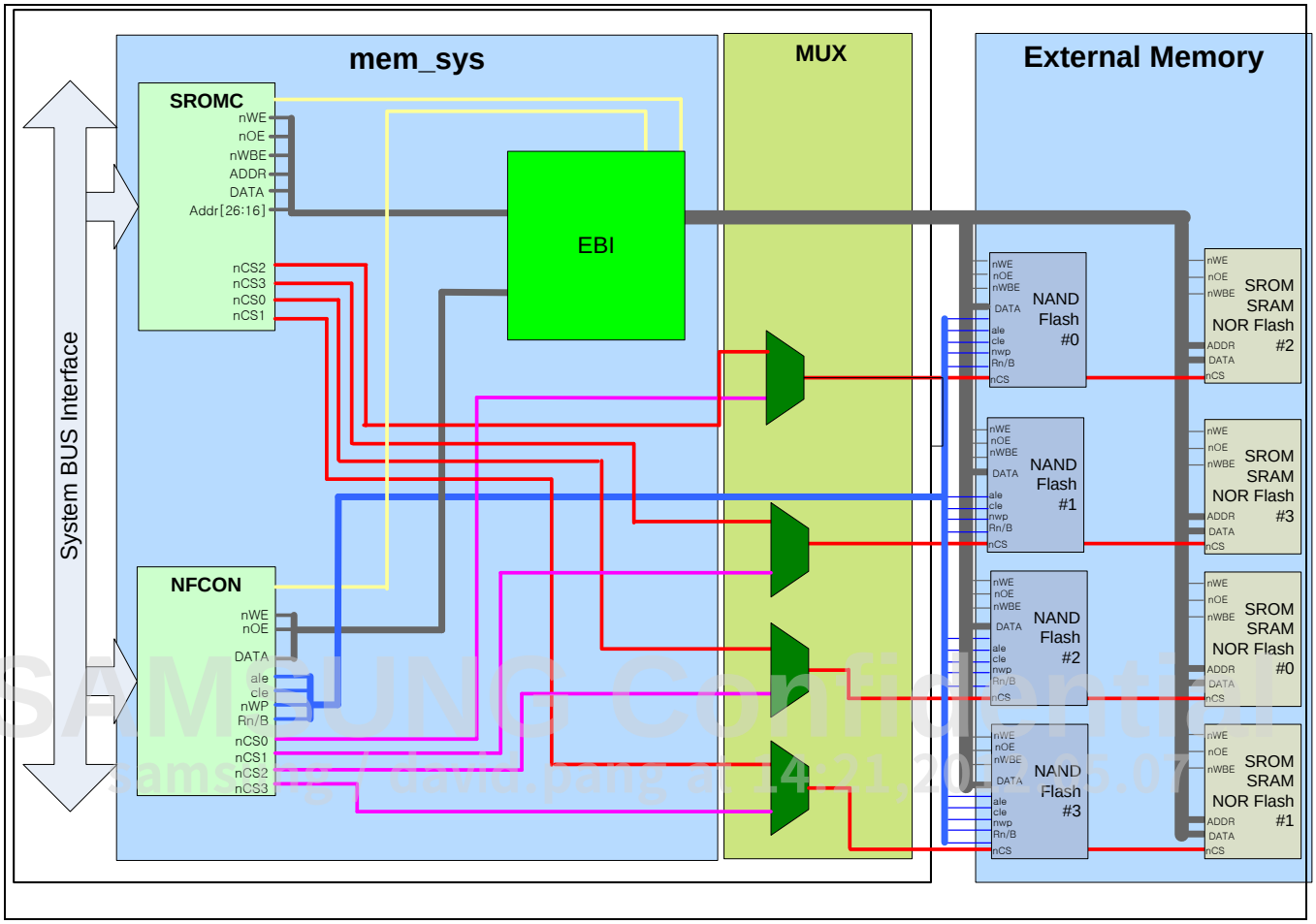


Figure 21-1 Memory Interface Through EBI

21.4 Clock Scheme of Memory Controllers and EBI

Figure 21-2 illustrates the clock scheme of memory controllers through EBI.

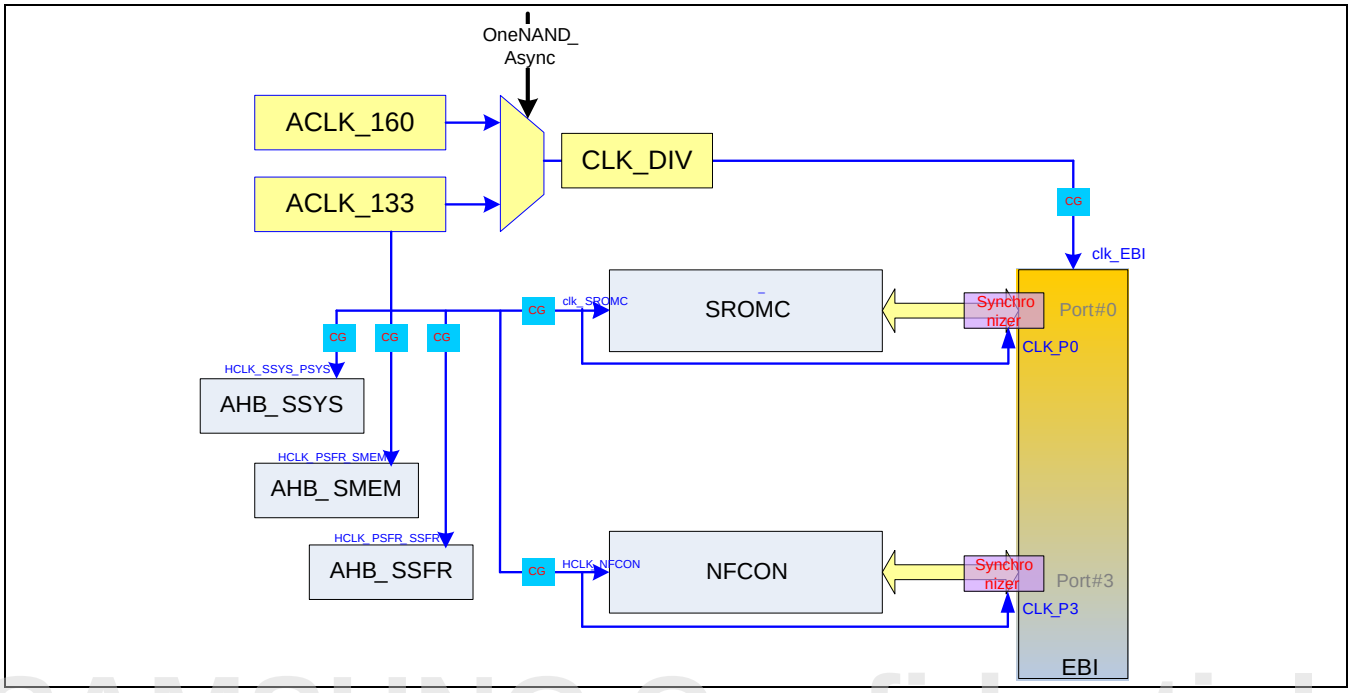


Figure 21-2 Clock Scheme of Memory Controllers and EBI

The constraint EBI has is:

EBI uses both the clocks, that is, ACLK_160 and ACLK_133. ACLK_160 and ACLK_133 are for external EBI interface.ACLK_133 is also for internal bus interfaces for SROMC, and NFCON memory controller.

When ACLK_160 is used for and EBI interface, the clock frequencies between ACLK_160 and ACLK_133 for internal bus interfaces are asynchronous. Therefore it consumes few clock cycles for EBI handshaking between EBI, SROMC and NFCON. The clock for EBI should be higher than SROMC and NFCON's because you can operate SROMC and NFCON under the condition.

The clock frequency setting for ACLK_160 and ACLK_133 which are used for EBI and memory interface, should be set carefully so that each memory controller can access designated external memory.

22 Secure Digital/MultiMediaCard MMC Controller

22.1 Overview

This chapter describes the Secure Digital (SD/SDIO), MultiMediaCard (MMC), Consumer Electronic-Advanced Technology Attachment (CE-ATA) host controller, and related registers that Exynos 4412 SCP RISC microprocessor supports.

The SD/MMC host controller is a combo host for SD/MMC. This host controller is based on SD Association's (SDA) Host Standard Specification.

The SD/MMC host controller is an interface between system and SD/MMC. The performance of this host is powerful, because clock rate is 50 MHz and can access 8-bit data pin simultaneously.

22.2 Features

Features of High-Speed MMC controller supports:

- SD Standard Host Specification Version 2.0 standard
- SD Memory Card Specification Version 2.0/High Speed MMC Specification Version 4.3 standard
- SDIO Card Specification Version 2.0 standard
- 512 bytes FIFO for data Tx/Rx
- CPU Interface and DMA data transfer mode
- 1-bit/4-bit/8-bit mode switch
- 8-bit 2 channel or 4-bit 4 channel
- Auto CMD12
- Suspend/Resume
- Read Wait operation
- Card Interrupt
- CE-ATA mode

22.3 Block Diagram

Figure 22-1 illustrates the SDMMC clock domain.

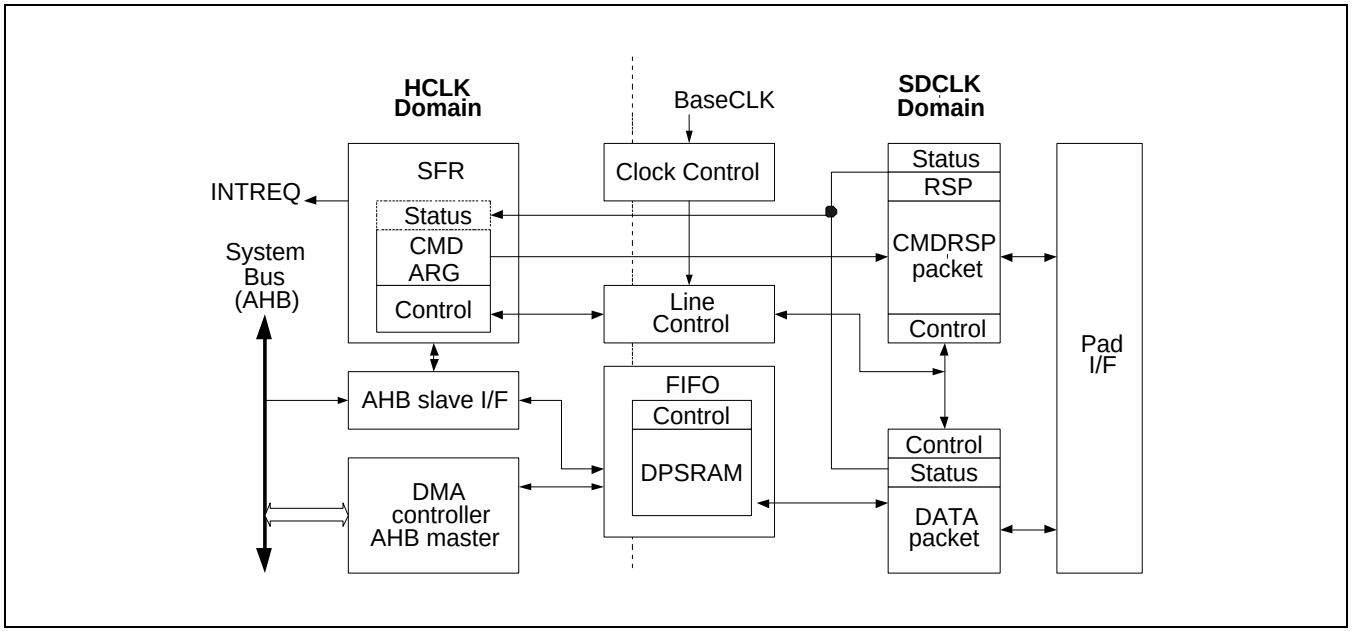


Figure 22-1 SDMMC Clock Domain

22.4 Operation Sequence

This section defines basic operation flow chart divided into several sub sequences. It uses wait for interrupts in the flow chart. This means, the Host Driver waits until it asserts specified interrupts. If already asserted, then it follows the next step in the flow chart. Timeout checking is only available when it does not generate interrupt. This is not described in the flow chart.

22.4.1 SD Card Detection Sequence

Figure 22-2 illustrates the flow chart to detect a SD card.

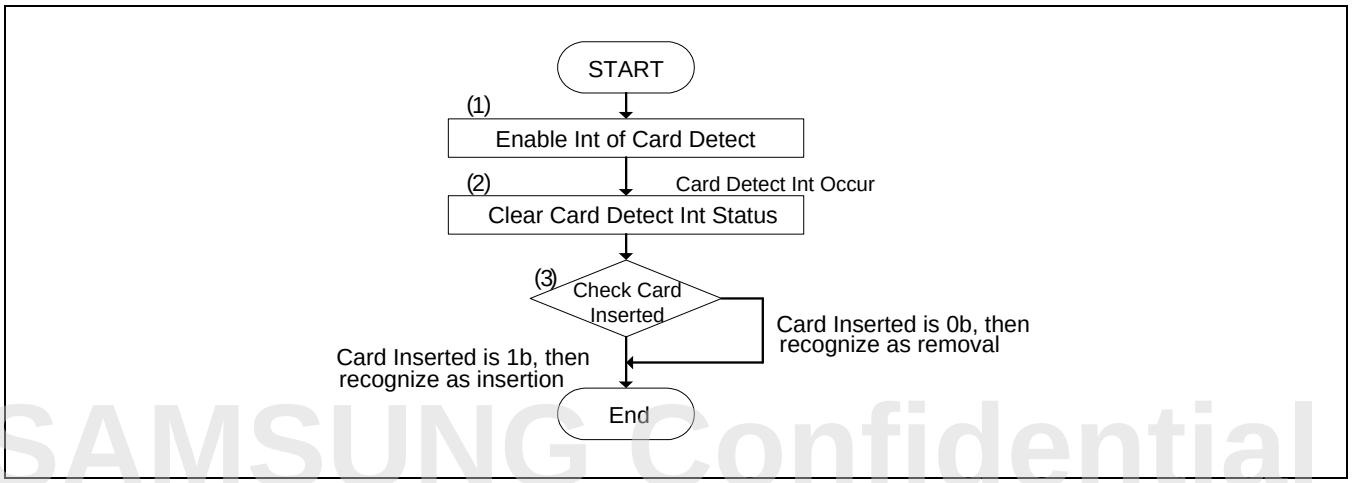


Figure 22-2 SD Card Detect Sequence

The steps to detect SD card are:

1. To enable interrupt for card detection, write 1 to these bits:
   Card Insertion Status Enable (ENSTACARDNS) in the Normal Interrupt Status Enable register.
   Card Insertion Signal Enable (ENSIGCARDNS) in the Normal Interrupt Signal Enable register.
   Card Removal Status Enable (ENSTACARDREM) in the Normal Interrupt Status Enable register.
   Card Removal Signal Enable (ENSIGCARDREM) in the Normal Interrupt Signal Enable register.

2. If Host Driver detects the card insertion or removal, it clears the interrupt statuses.
   If it generates Card Insertion interrupt (STACARDINS), write 1 to Card Insertion in the Normal Interrupt Status register.
   If it generates Card Removal interrupt (STACARDREM), write 1 to Card Removal in the Normal Interrupt Status register.

3. Check Card Inserted in the Present State register.
   If Card Inserted (INSCARD) is 1, the Host Driver supplies power and clock to the SD card.
   If Card Inserted is 0, it stops the executing process of the Host Driver.

22.4.2 SD Clock Supply Sequence

Figure 22-3 illustrates the sequence to set SD Clock to a SD card.

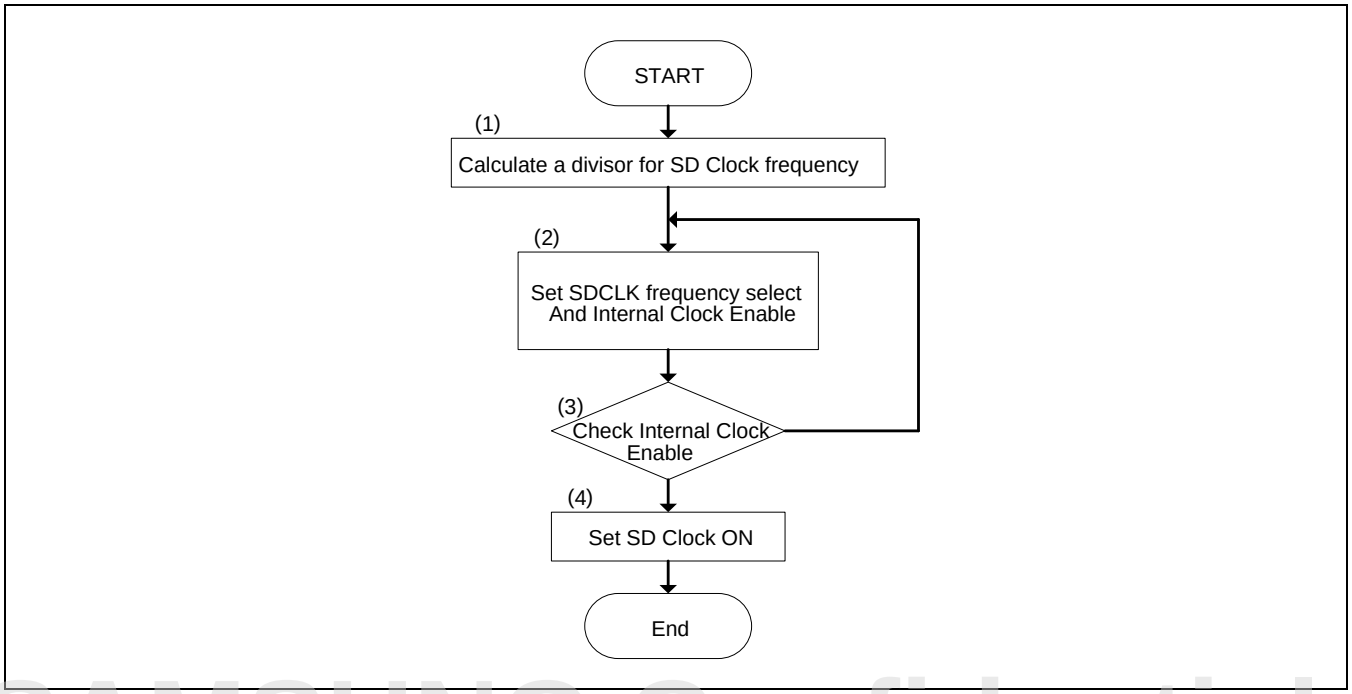


Figure 22-3 SD Clock Supply Sequence

It enables the clock before one of these actions:

- Issuing a SD command.
- Detect an interrupt from a SD card in 4-bit mode.

The steps to set SD Clock to a SD card are:

1. Calculate a divisor to determine SD Clock frequency for SD Clock (SDCLK) by reading Base Clock Frequency.
   Refer to Clock Control register (*22.9.1.17 CLKCONn (n = 0 to 3)*) for more information.

2. Set Internal Clock Enable (ENINTCLK) and SDCLK Frequency Select in the Clock Control register in accordance with the calculated result of step (1).

3. Check Internal Clock Stable (STBLINTCLK) in the Clock Control register. Repeat this step until Clock Stable is 1.

4. Set SD Clock Enable (ENSDCLK) in the Clock Control register to 1. After ENSDCLK is set, the Host Controller starts SD Clock.

22.4.3 SD Clock Stop Sequence

Figure 22-4 illustrates the flow chart to stop the SD Clock.

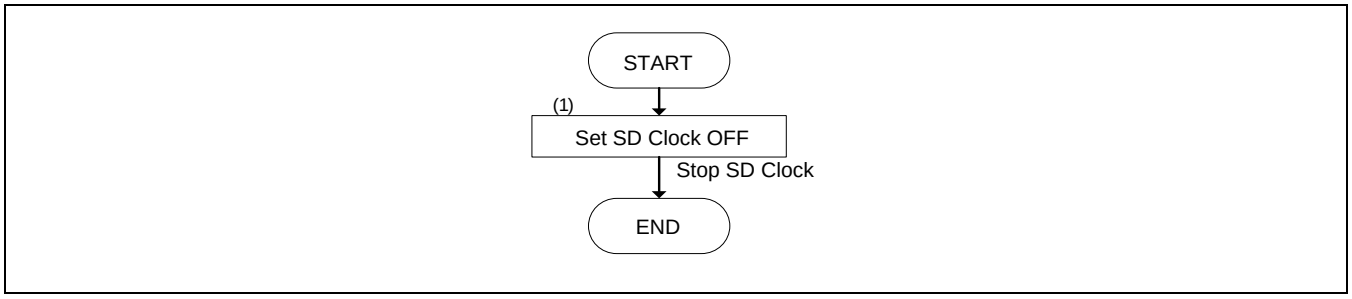


Figure 22-4 SD Clock Stop Sequence

The Host Driver does not stop the SD Clock if an SD transaction takes place on the SD Bus.: For example, it does not stop if the SD transaction takes place in either Command Inhibit (DAT) or Command Inhibit (CMD) in the Present State register is set to 1.

1. Set SD Clock Enable (ENSDCLK) in the Clock Control register to 0. After ENSDCLK is set, the Host Controller stops SD Clock.

22.4.4 SD Clock Frequency Change Sequence

Figure 22-5 illustrates the sequence to change SD Clock frequency.

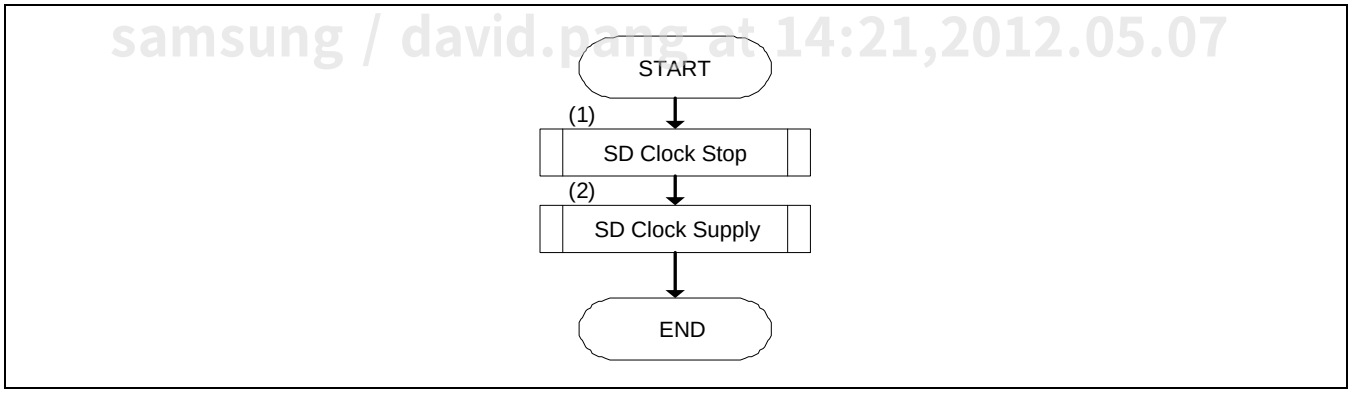


Figure 22-5 SD Clock Frequency Change Sequence

The steps to change SD Clock frequency are:

1. Perform SD Clock Stop Sequence. Refer to *22.4.2 SD Clock Supply Sequence* for more information.
2. Perform SD Clock Supply Sequence. Refer to *22.4.3 SD Clock Stop Sequence* for more information.

NOTE: If SD Clock is still off, skip step (1).

22.4.5 SD Bus Power Control Sequence

Figure 22-6 illustrates the sequence to control SD Bus Power.

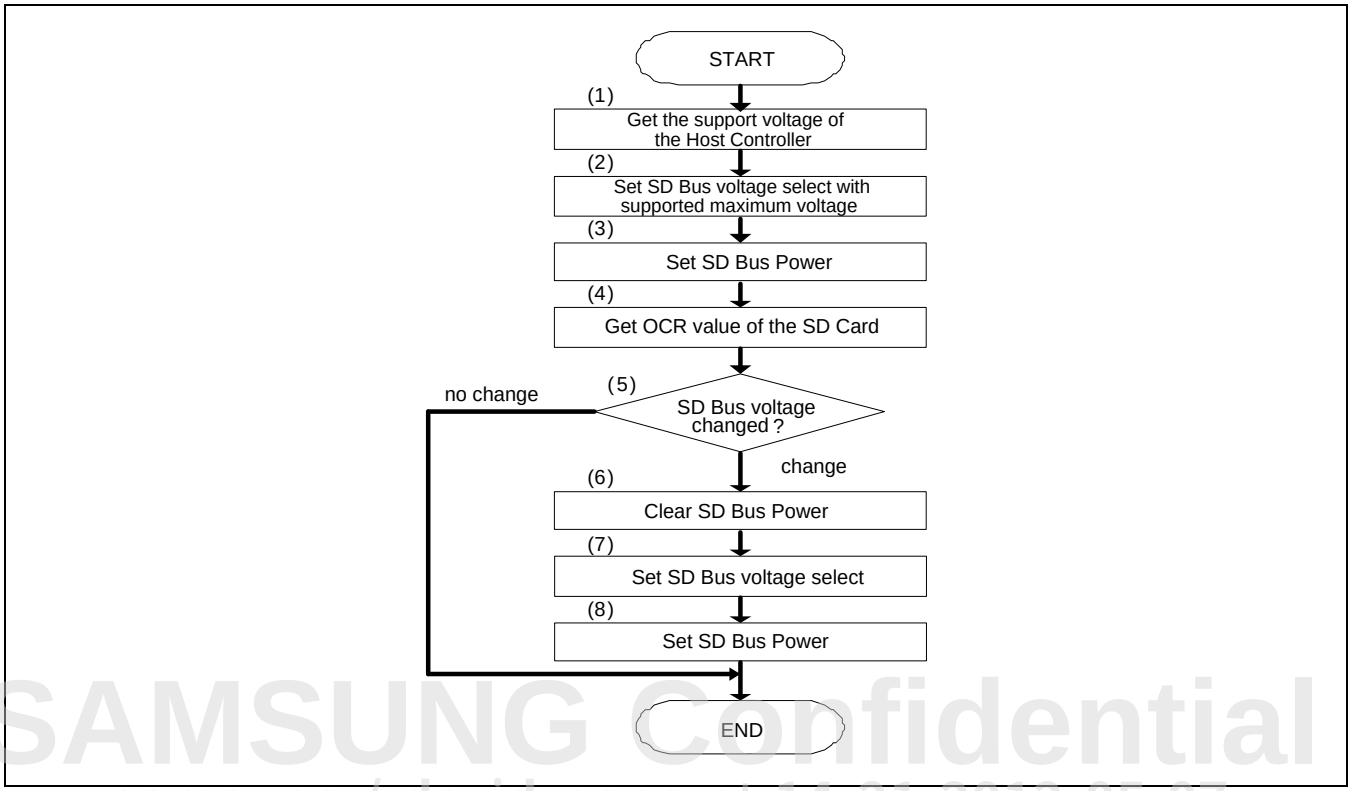


Figure 22-6 SD Bus Power Control Sequence

Steps to control SD Bus Power are:

1. Read the Capabilities register, to get the support voltage of the Host Controller.
2. Set SD Bus Voltage Select in external power regulator (optional) with maximum voltage that the Host Controller supports.
3. Set SD Bus Power (PWRON) in the Power Control register to 1.
4. Get the OCR value of all function internal of SD card.
5. Assess whether you should change SD Bus voltage or not. If you have to change SD Bus voltage, continue with step (6). If it does not require you to change SD Bus voltage, then go to "End".
6. Set SD Bus Power in the Power Control register to 0 for clearing this bit. The card requires voltage rising from 0 volt to detect it correctly. The Host Driver sets SD Bus Voltage Select to clear SD Bus Power before changing voltage.
7. Set SD Bus Voltage Select (SELPWRLVL) in the Power Control register.
8. Set SD Bus Power (PWRON) in the Power Control register to 1.

NOTE: You can execute step (2) and step (3) simultaneously. You can also execute step (7) and step (8) simultaneously.

22.4.6 Change Bus Width Sequence

Figure 22-7 illustrates the sequence to change bit mode on SD Bus.

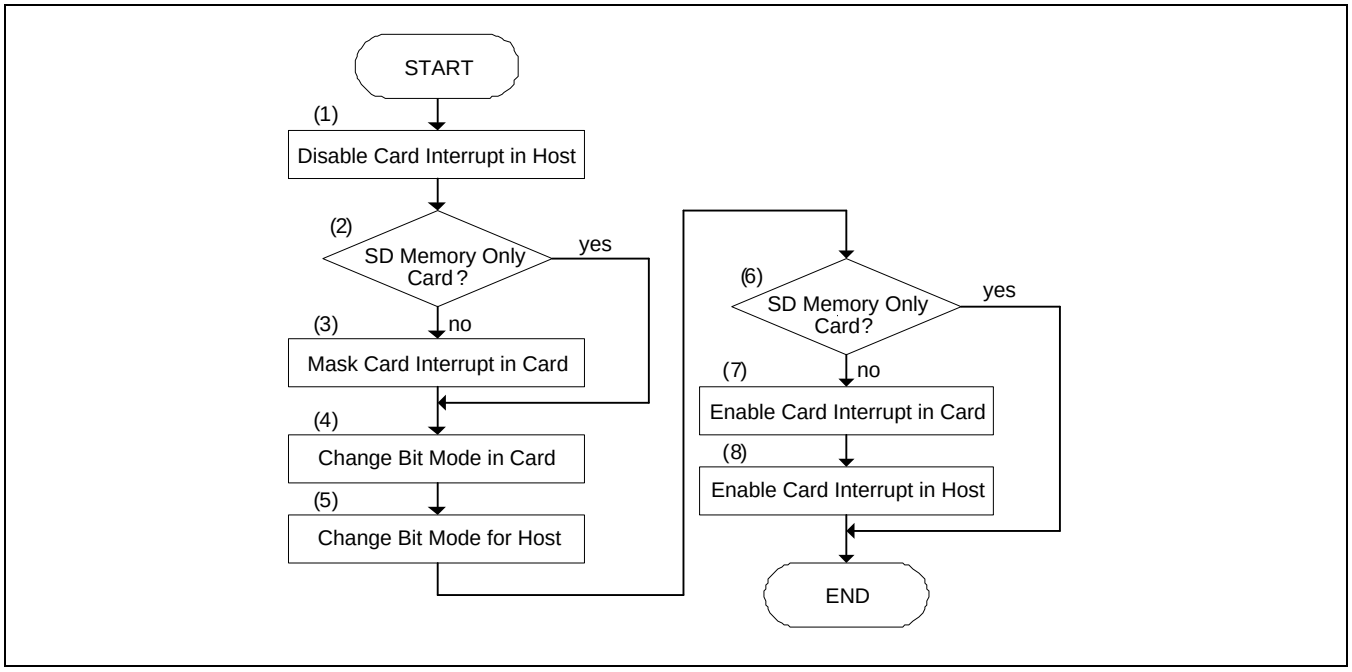


Figure 22-7 Change Bus Width Sequence

Steps to change bit mode on SD Bus are:

1. Set Card Interrupt Status Enable (STACARDINT) in the Normal Interrupt Status Enable register to 0, to mask incorrect interrupts that may occur while changing the bus width.
2. Go to step 4, if you use SD memory card. In case of other card, go to step 3.
3. Use CMD52 to set "IENM" of the CCCR in a SDIO or SD combo card to 0.
4. Change the bit mode for a SD card. To change SD memory card bus width by ACMD6 (Set bus width) and SDIO card bus width, set Bus Width of Bus Interface Control register in CCCR.
5. Set Data Transfer Width (WIDE4) to 1 in the Host Control register, if you want to change to 4-bit mode. In another case (1-bit mode), set this bit to 0.
6. Go to "End", if you use SD memory card. In case of other card, go to step 7.
7. Set "IENM" of the CCCR in a SDIO or SD combo card to 1 by CMD52.
8. Set Card Interrupt Status Enable to 1 in the Normal Interrupt Status Enable register.

22.4.7 Timeout Setting for DAT Line

Figure 22-8 illustrates the timeout setting sequence.

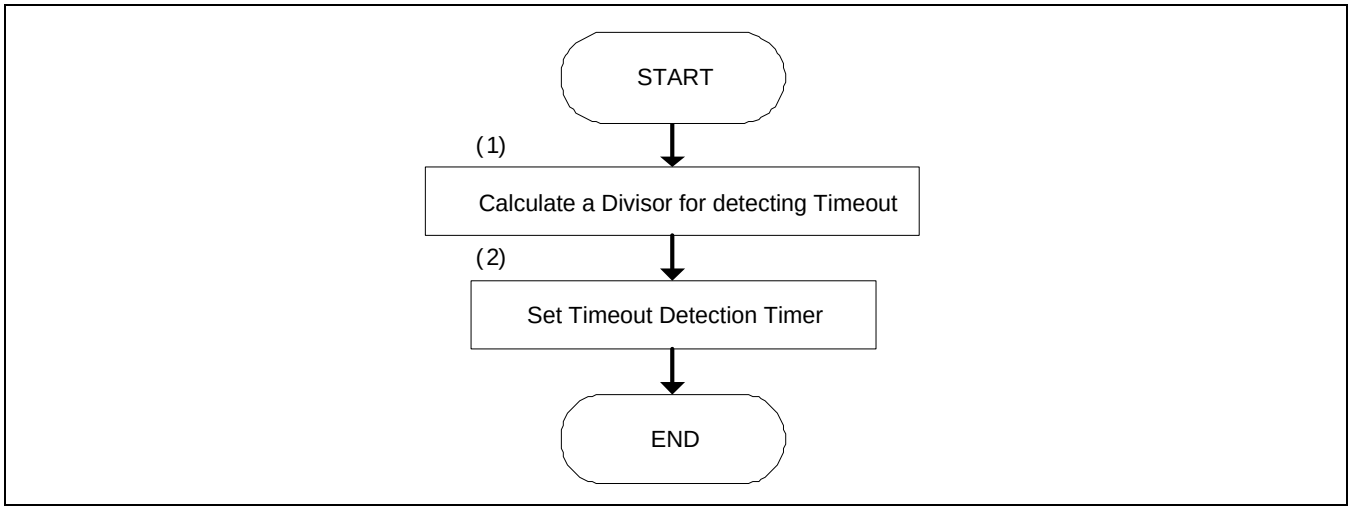


Figure 22-8 Timeout Setting Sequence

To detect timeout errors on DAT line, the Host Driver executes these two steps before any SD transaction.

1. To calculate a divisor for detecting timeout, refer to Timeout Control Register (22.9.1.18 TIMEOUTCONn (n = 0 to 3)).
2. Set Data Timeout Counter Value (TIMEOUTCON) in the Timeout Control register in accordance with the value of step (1).

22.4.8 SD Transaction Generation

This section describes the sequence to generate and control various types of SD transactions. You can classify SD transactions into three cases:

1. Transactions that do not use the DAT line.
2. Transactions that use the DAT line for the busy signal.
3. Transactions that use the DAT line for transferring data.

In this specification, it classifies the transactions of the first case and the second case as Transaction Control without Data Transfer using DAT Line. It classifies the transactions of third case as Transaction Control with Data Transfer using DAT Line.

Refer to these specifications for the detailed specifications on the SD Command itself:

- SD Memory Card Specification Part 1
  PHYSICAL LAYER SPECIFICATION Version 1.01
- SD Card Specification PART E1
  Secure Digital Input/Output (SDIO) Specification Version 1.00

22.4.9 SD Command Issue Sequence

Figure 22-9 illustrates the timeout setting sequence.

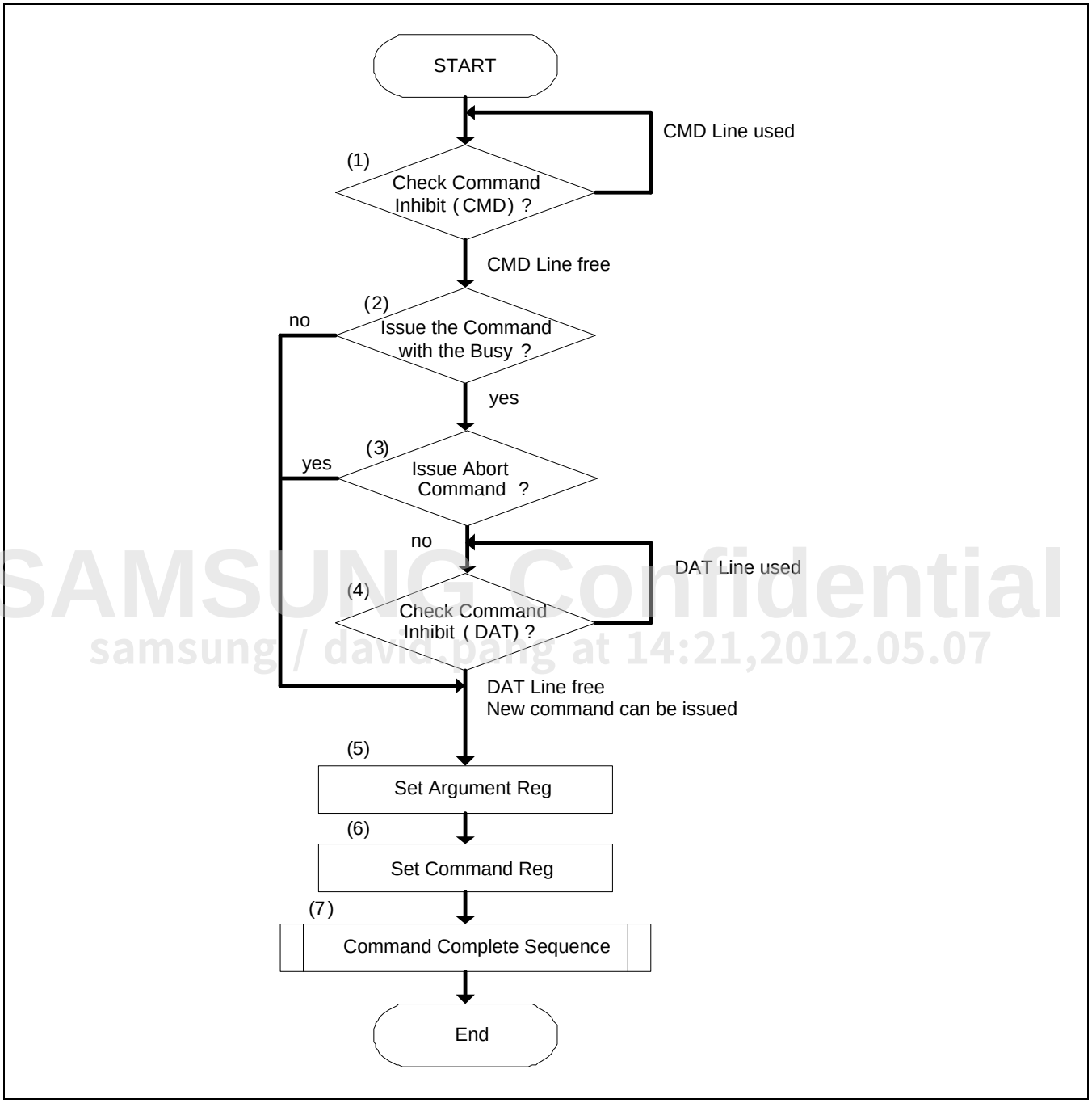


Figure 22-9 Timeout Setting Sequence

Steps to set Timeout:

1. Check Command Inhibit (CMD) in the Present State register. Repeat this step until Command Inhibit (CMD) is 0. If Command Inhibit (CMD) is 1, the Host Driver does not issue an SD Command.
2. If the Host Driver issues an SD Command with busy signal, go to step 3. If it issues without busy signal, go to step 5.
3. If the Host Driver issues an abort command, go to step 5.
   If it does not issues an abort command, go to step 4.
4. Check Command Inhibit (DAT) in the Present State register.
   Repeat this step until Command Inhibit (DAT) is 0.
5. Set the value corresponding to the issued command in the Argument register.
6. Set the value corresponding to the issued command in the Command register.

NOTE: If it writes an upper byte in the command register, it issues an SD command.

7. Perform command complete sequence.

22.4.10 Command Complete Sequence

Figure 22-7, *Figure 22-8*, *Figure 22-9*, and *Figure 22-10* illustrate the sequence to complete the SD Command. Errors can occur during this sequence: Command Index/End bit/CRC/Timeout Error.

Steps to complete the SD command:

1. Wait for the Command Complete Interrupt. If the Command Complete Interrupt occurs, go to step 2.
2. Write 1 to Command Complete (STACMDCMPLT) in the Normal Interrupt Status register to clear this bit.
3. Read the Response register and get necessary information in accordance with the issued command.
4. Assess whether the command uses the Transfer Complete Interrupt or not.
   If it uses Transfer Complete, proceed with step 5. If it does not use, go to step 7.
5. Wait for the Transfer Complete Interrupt. If the Transfer Complete Interrupt occurs, go to step 6.
6. Write 1 to Transfer Complete (STATRANCMPLT) in the Normal Interrupt Status register to clear this bit.
7. Check for errors in Response Data. If there is no error, go to step 8. If there is an error, go to step 9.
8. Return Status of "No Error".
9. Return Status of "Response Contents Error".

NOTE:

1. While waiting for the Transfer Complete interrupt, the Host Driver issues commands that do not use the busy signal.
2. The Host Driver monitors Transfer Complete to assess the Auto CMD12 (Stop Command) complete.
3. If it reads the last block of un-protected area using memory multiple blocks read command (CMD18), OUT_OF_RANGE error may occur even if the sequence is correct. The Host Driver should ignore this error.
   This error appears in the response of Auto CMD12 or in the response of the next memory command.

Figure 22-10 illustrates the command complete sequence.

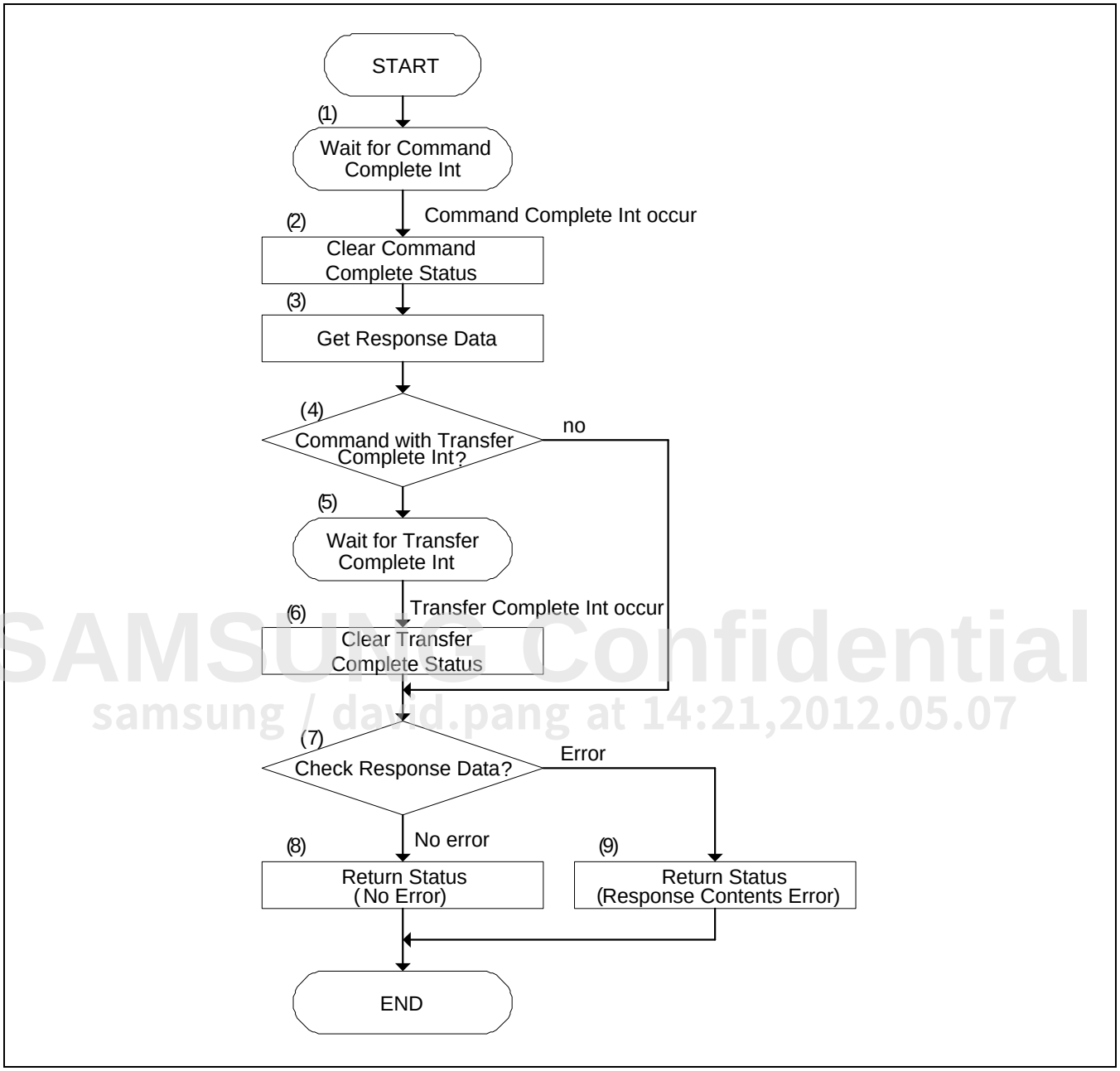


Figure 22-10 Command Complete Sequence

22.4.11 Transaction Control with Data Transfer Using DAT Line

Depending on whether it uses DMA (optional) or not, there are two execution methods. *Figure 22-11* illustrates the sequence without using DMA. *Figure 22-12* illustrates the sequence using DMA.

Additionally, it classifies sequences for SD transfers according to how it specifies the number of blocks. There are three types of classifications. They are:

1. Single Block Transfer:
   It specifies the number of blocks to the Host Controller before the transfer. It always specifies one block.

2. Multiple Block Transfer:
   It specifies the number of blocks to the Host Controller before the transfer. It specifies one or more blocks.

3. Infinite Block Transfer:
   It does not specify the number of blocks to the Host Controller before the transfer. It continues this transfer until it executes an abort transaction. In the case of a SD memory card, CMD12 (Stop Command) performs this abort transaction. In the case of a SDIO card, CMD52 (IO_RW_DIRECT) performs this abort transaction.

22.4.12 Sequence without Using DMA

Figure 22-11 illustrates the transaction control with data transfer using DAT line sequence.

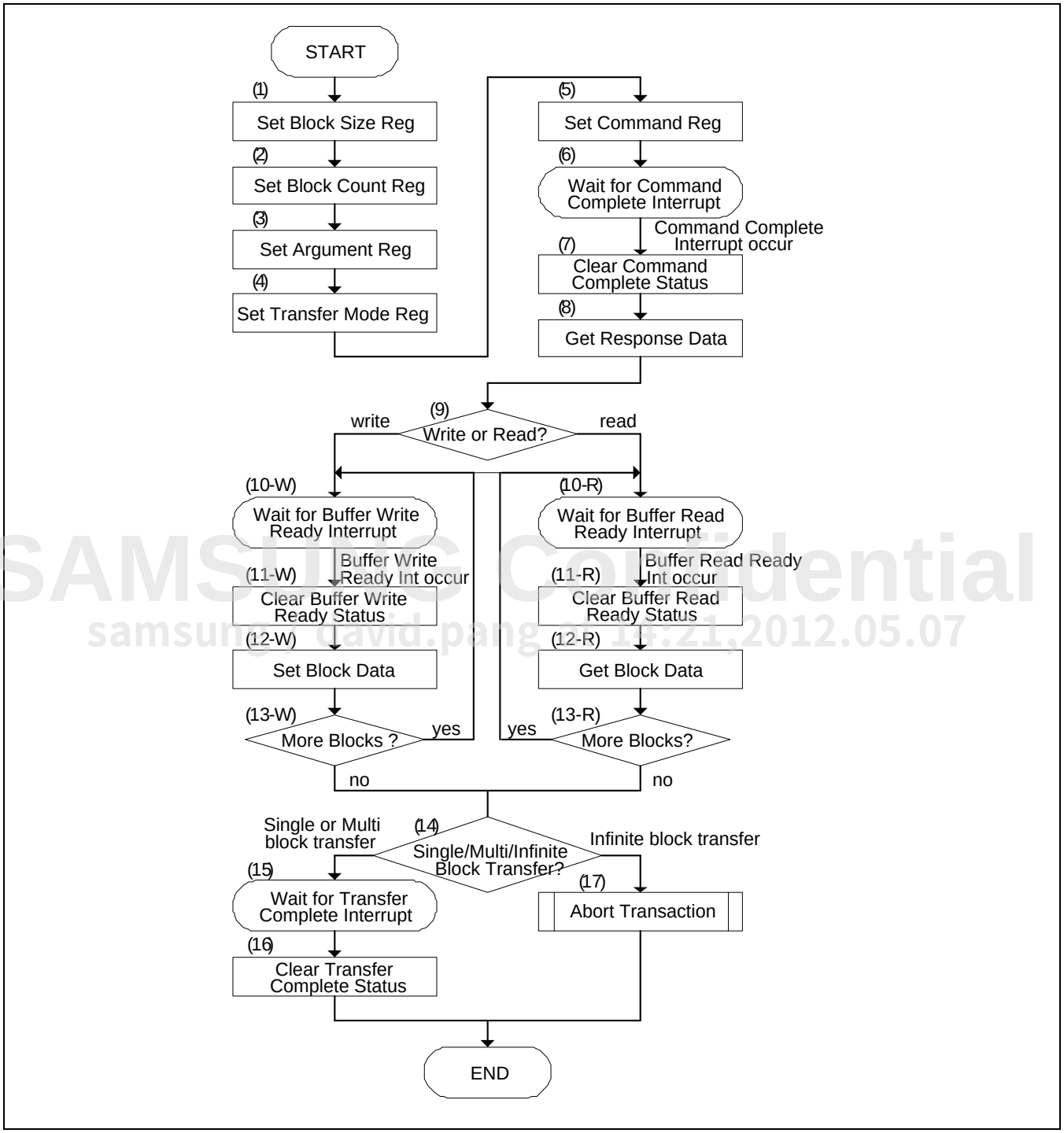


Figure 22-11 Transaction Control with Data Transfer Using DAT Line Sequence (Not Using DMA)

Steps to complete a transaction without using DMA are

1. Set the value corresponding to the executed data byte length of one block to Block Size register.
2. Set the value corresponding to the executed data block count to Block Count register.
3. Set the value corresponding to the issued command to Argument register.
4. Set the value to Multi/Single Block Select and Block Count Enable. At this time, set the value corresponding to the issued command to Data Transfer Direction, Auto CMD12 Enable, and DMA Enable.
5. Set the value corresponding to the issued command to Command register.

NOTE: If it writes the upper byte in the Command register, it issues a SD command.

6. Wait for the command complete interrupt.
7. Write 1 to the Command Complete (STACMDCMPLT) in the Normal Interrupt Status register to clear this bit.
8. Read Response register and get necessary information in accordance with the issued command.
9. If this sequence is for Write to a card, go to step (10-W). If it reads from a card, go to step (10-R).
10. (10-W) Wait for buffer write ready interrupt.
11. (11-W) Write 1 to the Buffer Write Ready (STABUFWTRDY) in the Normal Interrupt Status register to clear this bit.
12. (12-W) Write block data (according to the number of bytes specified at the step (1)) to Buffer Data Port register.
13. (13-W) Repeat until it sends all blocks and then goes to step 14.
14. (10-R) Wait for the Buffer Read Ready Interrupt.
15. (11-R) Write 1 to the Buffer Read Ready (STABUFRDRDY) in the Normal Interrupt Status register to clear this bit.
16. (12-R) Read block data (according to the number of bytes specified at the step (1)) from the Buffer Data Port register.
17. (13-R) Repeat until it receives all blocks and go to step (14).
18. (14) If this sequence is for Single or Multiple Block Transfer, go to step (15). In case of Infinite Block Transfer, go to step (17).
19. (15) Wait for Transfer Complete Interrupt.
20. (16) Write 1 to the Transfer Complete (STATRANCMPLT) in the Normal Interrupt Status register to clear this bit.
21. (17) Perform the sequence for Abort Transaction.

NOTE: You can execute step (1) and step (2) simultaneously. You can execute step (4) and step (5) simultaneously.

22.4.13 Sequence Using DMA

Figure 22-12 illustrates the transaction control with data transfer using DAT line sequence.

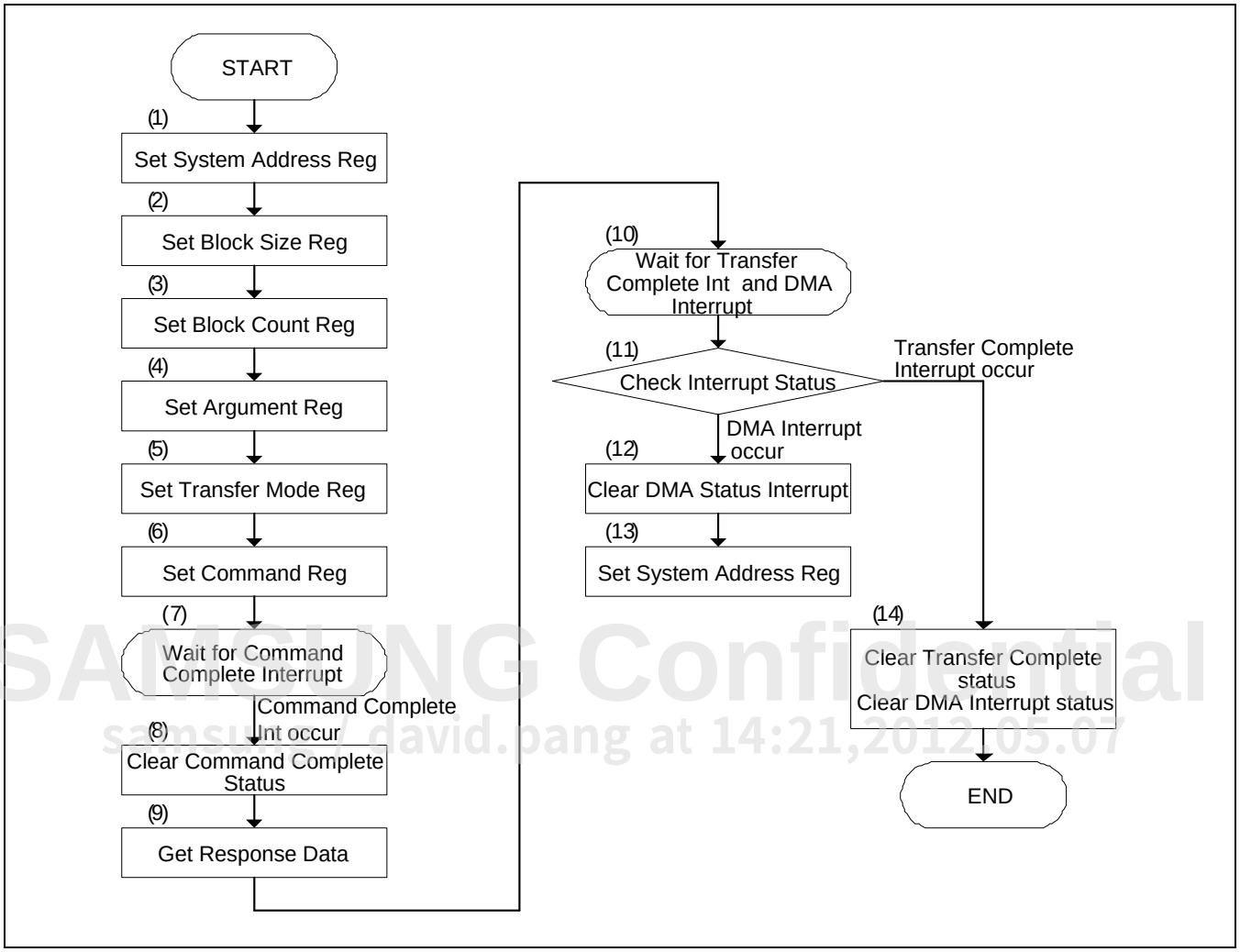


Figure 22-12 Transaction Control with Data Transfer Using DAT Line Sequence (Using DMA)

Steps to complete a transaction by using DMA are:

1. Set the system address for DMA in the System Address register.
2. Set the value corresponding to the executed data byte length of one block in the Block Size register.
3. Set the value corresponding to the executed data block count in the Block Count register (BLKCNT).
4. Set the value corresponding to the issued command in the Argument register (ARGUMENT).
5. Set the values for Multi/Single Block Select and Block Count Enable.

NOTE: At this time, set the value corresponding to the issued command for Data Transfer Direction, Auto CMD12 Enable, and DMA Enable.

6. Set the value corresponding to the issued command in the Command register (CMDREG).

NOTE: If it writes the upper byte in the Command register, it issues a SD command and it operates DMA.

7. Wait for the Command Complete Interrupt.
8. Write 1 to the Command Complete (STACMDCMPLT) in the Normal Interrupt Status register to clear this bit.
9. Read Response register and get necessary information in accordance to the issued command.
10. Wait for the Transfer Complete Interrupt and DMA Interrupt.
11. If Transfer Complete (STATRANCMPLT) is set to 1, go to step 14, If DMA Interrupt is set to 1, proceed to step 12. Transfer Complete is higher priority than DMA Interrupt.
12. Write 1 to the DMA Interrupt in the Normal Interrupt Status register to clear this bit.
13. Set the next system address of the next data position to the System Address register and go to step 10.
14. Write 1 to the Transfer Complete and DMA Interrupt in the Normal Interrupt Status register to clear this bit.

NOTE: You can execute step 2 and step 3 simultaneously. You can also execute step 5 and step 6 simultaneously.

22.5 Abort Transaction

To perform Abort transaction, issue CMD12 (Stop command) for a SD memory and issue CMD52 for a SDIO card. There are two cases where the Host Driver needs to issue an Abort Transaction:

- If the Host Driver stops Infinite Block Transfers.
- If the Host Driver stops transfers while executing a Multiple Block Transfer.

There are two ways to issue an Abort Command:

- Asynchronous abort
- Synchronous abort

In an asynchronous abort sequence, the Host Driver issues an Abort Command at anytime unless Command Inhibit (CMD) in the Present State register is set to 1.

In a synchronous abort, the Host Driver issues an Abort Command after it stops the data transfer by using Stop At Block Gap Request in the Block Gap Control register.

22.6 DMA Transaction

DMA allows a peripheral to Read and Write memory without intervention from the CPU. DMA executes one SD command transaction. Host Controllers supporting DMA transfer are capable of both single block and multiple block transfers.

The System Address register points to the first data address and then it accesses data sequentially from that address. Host Controller registers are still accessible for issuing non-DAT line commands during a DMA transfer. The result of a DMA transfer is similar, irrespective of the system bus transaction method. DMA does not support infinite transfers.

The controller stops and restarts DMA transfers using control bits in the Block Gap Control register. If the Stop At Block Gap Request is set to 1, it suspends DMA transfers. If it sets the Continue Request or issues a Resume Command, DMA continues to execute transfers. If SD Bus error occurs, it stops SD Bus transfers and DMA transfers. Setting the Software Reset for DAT Line in the Software Reset register aborts DMA transfers. Refer to Block Gap Control register for more information.

22.7 Advanced DMA

In the SD Host Controller Standard Specification Version 2.00, it defines new DMA transfer algorithm called Advanced DMA (ADMA). The DMA algorithm defined in the SD Host Controller Standard Specification Version 1.00 is called Single Operation DMA (SDMA). SDMA had disadvantage that DMA Interrupt generated at every page boundary that disturbs CPU to reprogram the new system address. This SDMA algorithm forms a performance bottleneck by interruption at every page boundary. ADMA adopts scatter gather DMA algorithm so that higher data transfer speed is available. The Host Driver can program a list of data transfers between system memory and SD card to the Descriptor Table before executing ADMA. It enables ADMA to operate without interrupting the Host Driver. Moreover, ADMA can support not only 32-bit system memory addressing but also 64-bit system memory addressing. The 32-bit system memory addressing uses lower 32-bit field of 64-bit address registers. Support of SDMA and ADMA are optional for the Host Controller. ADMA improves the restriction so that it transfers data of any location and any size in system memory. The format of Descriptor Table is different between the one of ADMA and SDMA. The Host Controller Specification Version 2.00 defines ADMA as standard ADMA.

22.7.1 Block Diagram of ADMA

Figure 22-13 illustrates the block diagram of ADMA.

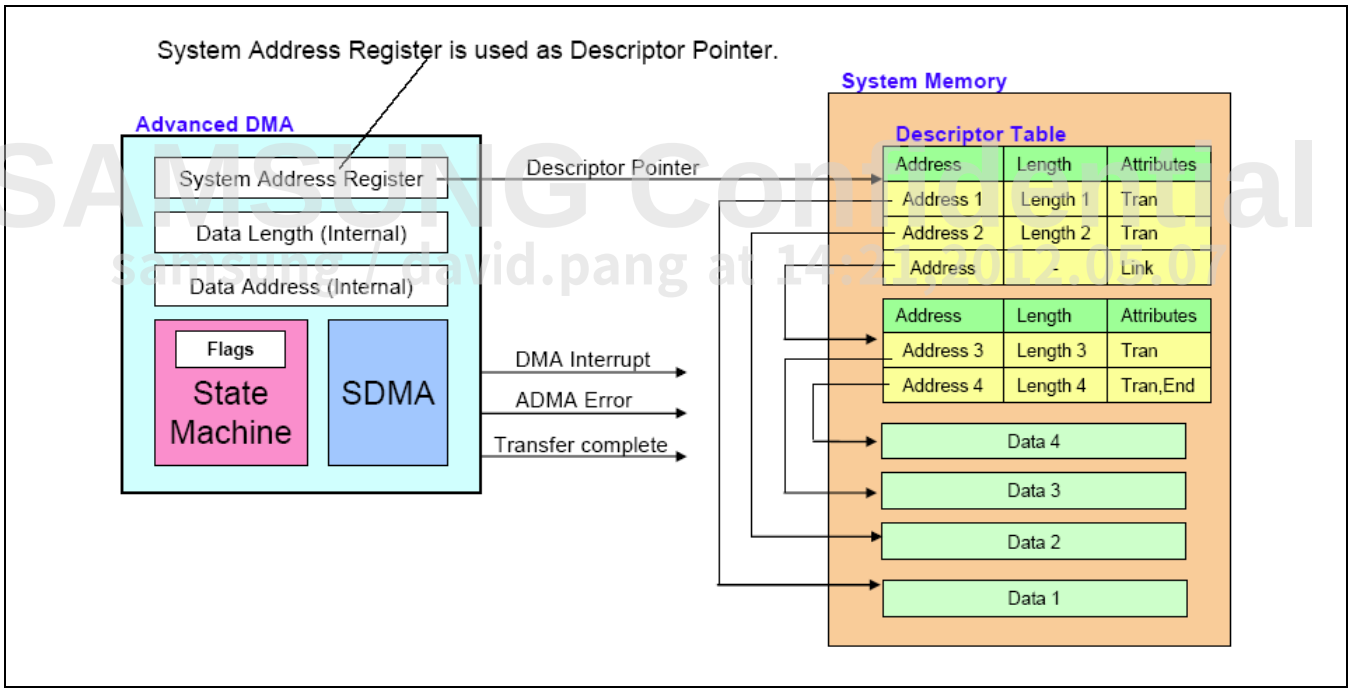


Figure 22-13 Block Diagram of ADMA

The Host Driver creates the Descriptor Table in system memory. It uses 32-bit Address Descriptor Table for the system with 32-bit addressing. It also uses 64-bit Address Descriptor Table for the system with 64-bit addressing. Each descriptor line (one executable unit) consists of address, length, and attribute field. The attribute specifies operation of the descriptor line. ADMA includes SDMA, State Machine, and Registers circuits. ADMA does not use 32-bit SDMA System Address Register (offset 0) but uses the 64-bit ADMA System Address register (offset 058h) for descriptor pointer. Writing Command register triggers off ADMA transfer. ADMA fetches one descriptor line and executes. It repeats this procedure until it finds end of descriptor (End = 1 in attribute).

22.7.2 Example of ADMA Programming

Figure 22-14 illustrates the example of ADMA data transfer.

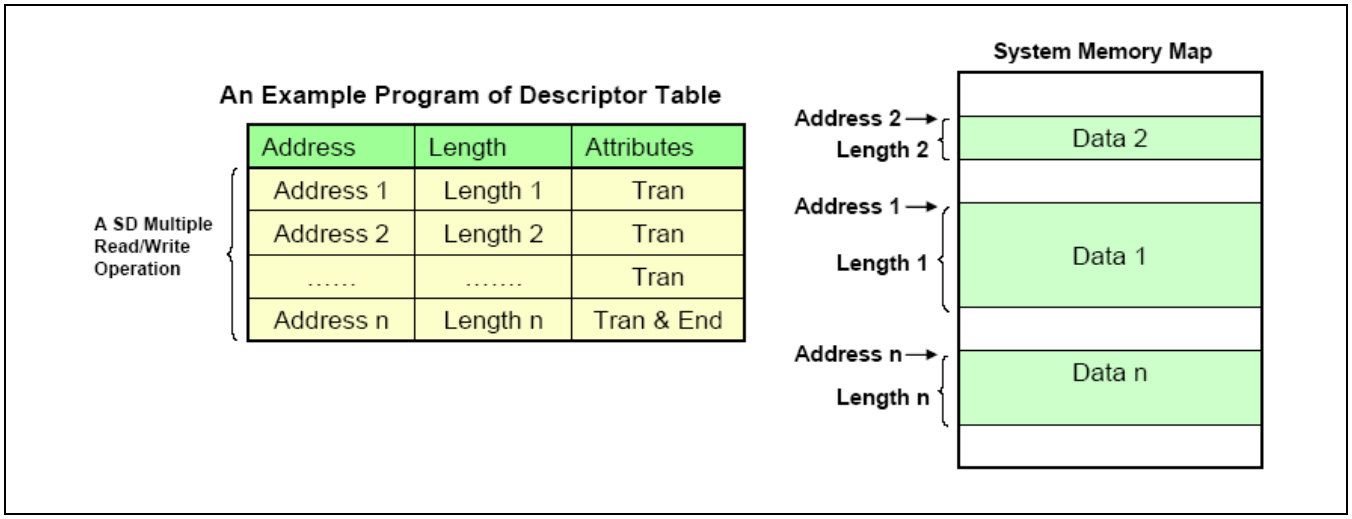


Figure 22-14 Example of ADMA Data Transfer

It slices the data area in various lengths and it places each slice somewhere in system memory. The Host Driver describes the Descriptor Table with set of addresses, length, and attributes. It transfers each sliced data in turns as programmed in descriptor.

22.7.3 Data Address and Data Length Requirements

There are three requirements to program descriptor:

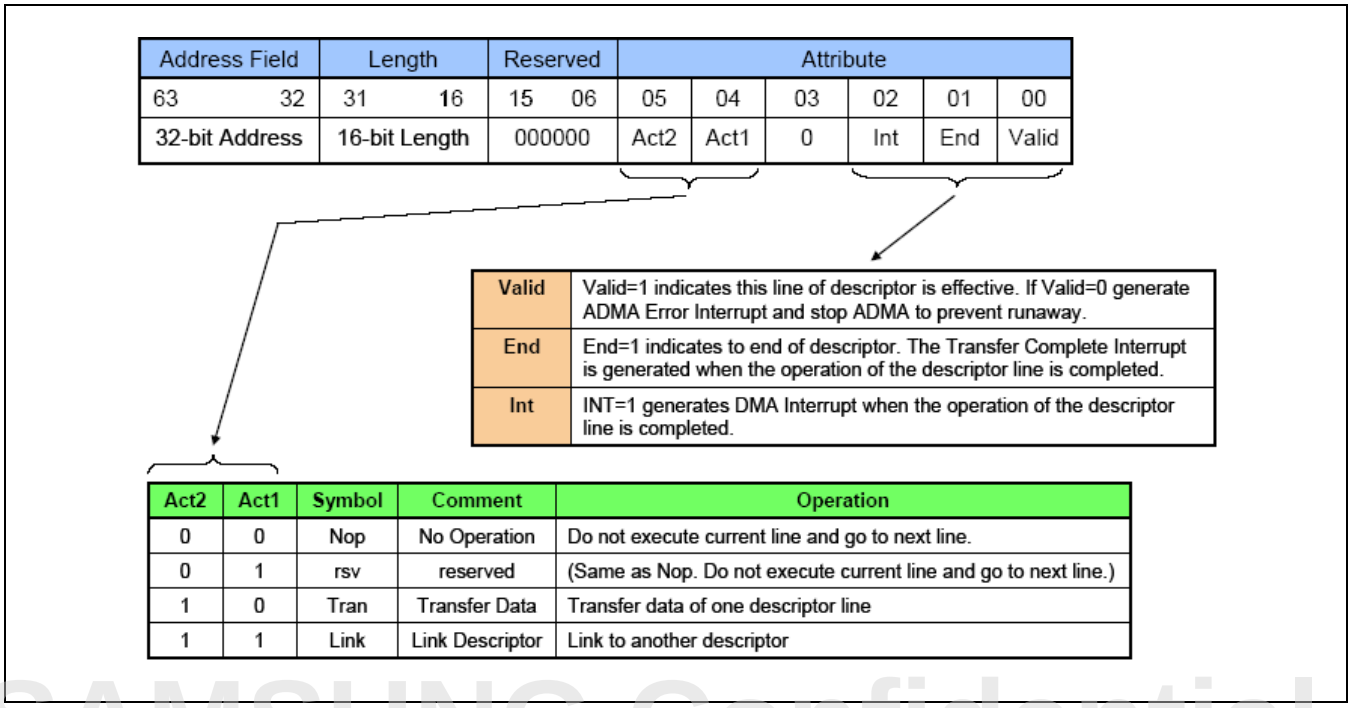
- The minimum unit of address is 4 byte.
- The maximum data length of each descriptor line is less than 64 KB.

Total Length = Length 1 + Length 2 + Length 3 + ... + Length n = multiple of Block Size

If total length of a descriptor were not multiple of block size, it may not terminate ADMA transfer. In this case, data timeout should abort the transfer. It defines Block Count register as 16-bit register and it limits the maximum of 65535 blocks transfer. If ADMA operation is less than or equal to 65535 blocks transfer, it uses Block Count register. In this case, total length of Descriptor Table is equivalent to multiply block size and block count. If ADMA operation is more than 65535 blocks transfer, it disables Block Count register by setting 0 to Block Count Enable in the Transfer Mode Register. In this case, block count does not designate length of data transfer. Descriptor Table designates length of data transfer. Therefore, the timing of detecting the last block on SD Bus may be different and it affects the control of Read Transfer Active, Write Transfer Active, and DAT line Active in the Present State register. For Read operation, it may read several blocks more than it may require. The Host Driver ignores out of range error if the Read operation is for the last block of memory area.

22.7.4 Descriptor Table

Figure 22-15 illustrates the definition of 32-bit Address Descriptor Table.



Address Field		Length		Reserved		Attribute					
63	32	31	16	15	06	05	04	03	02	01	00
32-bit Address		16-bit Length		000000		Act2	Act1	0	Int	End	Valid

Valid	Valid=1 indicates this line of descriptor is effective. If Valid=0 generate ADMA Error Interrupt and stop ADMA to prevent runaway.
End	End=1 indicates to end of descriptor. The Transfer Complete Interrupt is generated when the operation of the descriptor line is completed.
Int	INT=1 generates DMA Interrupt when the operation of the descriptor line is completed.

Act2	Act1	Symbol	Comment	Operation
0	0	Nop	No Operation	Do not execute current line and go to next line.
0	1	rsv	reserved	(Same as Nop. Do not execute current line and go to next line.)
1	0	Tran	Transfer Data	Transfer data of one descriptor line
1	1	Link	Link Descriptor	Link to another descriptor

Figure 22-15 32-bit Address Descriptor Table

One descriptor line consumes 64-bit (8 byte) memory space. It uses attribute to control descriptor. It specifies three action symbols. "Nop" operation skips current descriptor line and fetches next one. Address and length field designate Tran operation transfers data. It uses Link operation to connect two separated descriptors. The address field of link points to next Descriptor Table. It reserves the combination of Act2 = 0 and Act1 = 1 and defines the similar operation as Nop. A future version of controller may use this field and redefine a new operation. It stores 32-bit address in the lower 32-bit of 64-bit address registers. Address field is set on 32-bit boundary (Lower 2-bit is always set to 0) for 32-bit address descriptor table.

Table 22-1 describes the definition of length field in the Descriptor Table.

Table 22-1 ADMA Length Field

Length Field	Value of Length
0000h	65536 bytes
0001h	1 byte
0002h	2 bytes
…	…
FFFFh	65535 bytes

22.7.5 ADMA States

ADMA defines four states. They are:

- Fetch Descriptor state
- Change Address state
- Transfer Data state
- Stop ADMA state

Figure 22-16 illustrates the state diagram of ADMA.

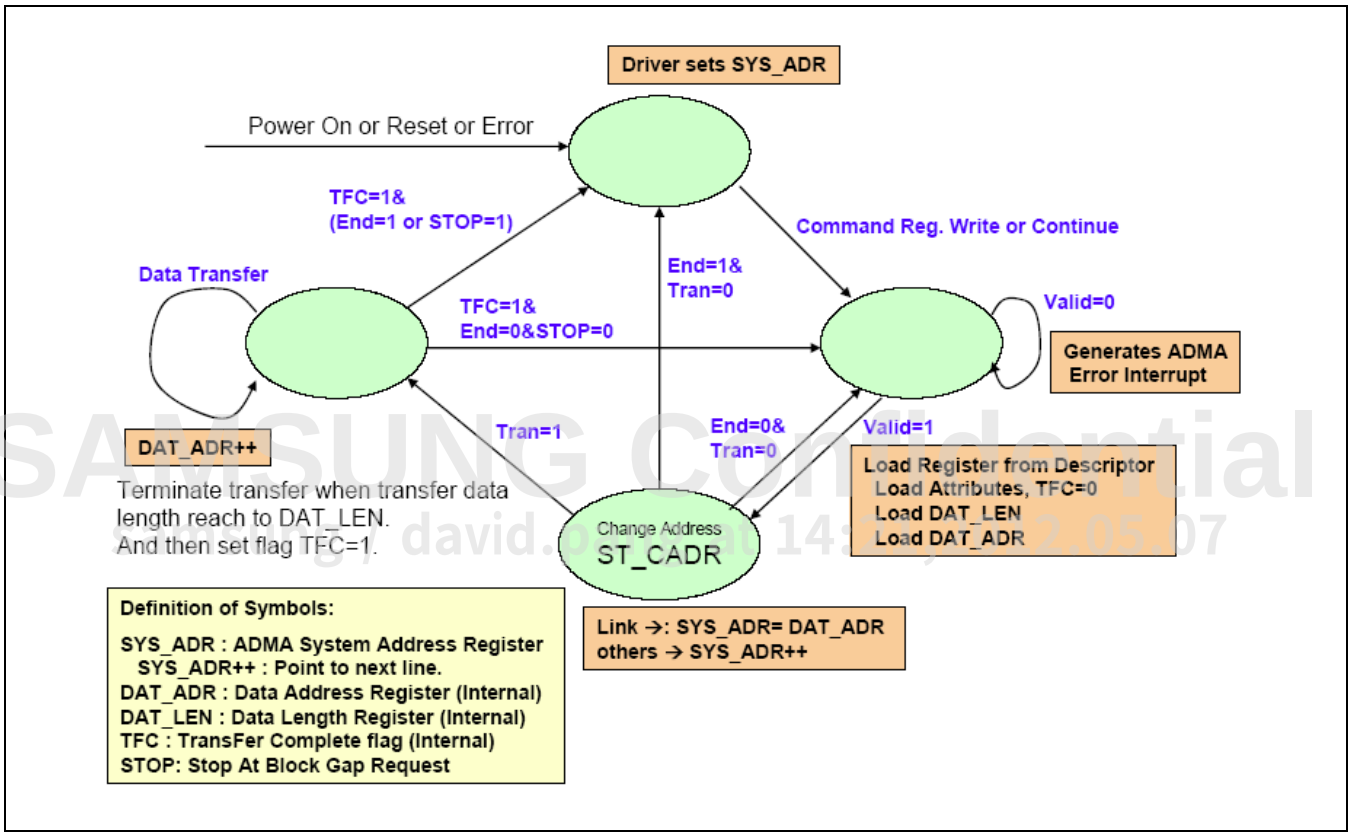


Figure 22-16 State Diagram of the ADMA

Table 22-2 describes the operation of each ADMA state.

Table 22-2 ADMA States

State Name	Operation
ST_FDS (Fetch Descriptor)	ADMA fetches a descriptor line and set parameters in internal registers. Next go to ST_CADR state.
ST_CADR (Change Address)	Link operation loads another Descriptor address to ADMA System Address register. In other operations, it increments ADMA System Address register to point next descriptor line. If End = 0, go to ST_TFR state. It will not stop ADMA at this state even if some errors occur.
ST_TFR (Transfer Data)	It executes data transfer of one descriptor line between system memory and SD card. If data transfer continues (End = 0), go to ST_FDS state. If data transfer completes, go to ST_STOP state.
ST_STOP (Stop DMA)	ADMA stays in ST_STOP state in these cases: 1. After Power on reset or software reset. 2. It completes all descriptor data transfers. If it starts a new ADMA operation by writing Command register, go to ST_FDS state.

ADMA does not support suspend/resume function but stop and continue functions are available. When the Stop At Block Gap Request in the Block Gap Control register is set to 1 during the ADMA operation, it generates the Block Gap Event Interrupt when it stops ADMA at block gap. The Host Controller stops ADMA read operation by using Read Wait or stopping SD Clock. While stopping ADMA, it cannot issue SD commands. (In case of Host Controller version 1.00, it can set the Stop at Block Gap Request only when the card supports the Read Wait.)

Error occurrence during ADMA transfer can stop ADMA operation and generates ADMA Error Interrupt. It stops the ADMA Error State field in the ADMA Error Status register holds state of ADMA. The host driver can identify error descriptor location by following the below method.

If it stops ADMA at ST_FDS state, the ADMA System Address Register points the error descriptor line. If it stops ADMA at ST_TFR or ST_STOP state, the ADMA System Address Register points the next location of error descriptor line. By this reason, ADMA2 does not stop at ST_CADR state.

22.8 I/O Description

Signal	I/O	Description	Pad	Type
SD_0_CLK	OUTPUT	Clock for SDMMC0	Xmmc0CLK	muxed
SD_0_CMD	IN/OUT	Command for SDMMC0	Xmmc0CMD	muxed
SD_0_DATA[0]	IN/OUT	Data for SDMMC0	Xmmc0DATA[0]	muxed
SD_0_DATA[1]	IN/OUT	Data for SDMMC0	Xmmc0DATA[1]	muxed
SD_0_DATA[2]	IN/OUT	Data for SDMMC0	Xmmc0DATA[2]	muxed
SD_0_DATA[3]	IN/OUT	Data for SDMMC0	Xmmc0DATA[3]	muxed
SD_0_DATA[4]	IN/OUT	Data for SDMMC0	Xmmc1DATA[0]	muxed
SD_0_DATA[5]	IN/OUT	Data for SDMMC0	Xmmc1DATA[1]	muxed
SD_0_DATA[6]	IN/OUT	Data for SDMMC0	Xmmc1DATA[2]	muxed
SD_0_DATA[7]	IN/OUT	Data for SDMMC0	Xmmc1DATA[3]	muxed
SD_0_CDn	INPUT	Card Detect for SDMMC0	Xmmc0CDn	muxed
SD_1_CLK	OUTPUT	Clock for SDMMC1	Xmmc1CLK	muxed
SD_1_CMD	IN/OUT	Command for SDMMC1	Xmmc1CMD	muxed
SD_1_DATA[0]	IN/OUT	Data for SDMMC1	Xmmc1DATA[0]	muxed
SD_1_DATA[1]	IN/OUT	Data for SDMMC1	Xmmc1DATA[1]	muxed
SD_1_DATA[2]	IN/OUT	Data for SDMMC1	Xmmc1DATA[2]	muxed
SD_1_DATA[3]	IN/OUT	Data for SDMMC1	Xmmc1DATA[3]	muxed
SD_1_CDn	INPUT	Card Detect for SDMMC1	Xmmc1CDn	muxed
SD_2_CLK	OUTPUT	Clock for SDMMC2	Xmmc2CLK	muxed
SD_2_CMD	IN/OUT	Command for SDMMC2	Xmmc2CMD	muxed
SD_2_DATA[0]	IN/OUT	Data for SDMMC2	Xmmc2DATA[0]	muxed
SD_2_DATA[1]	IN/OUT	Data for SDMMC2	Xmmc2DATA[1]	muxed
SD_2_DATA[2]	IN/OUT	Data for SDMMC2	Xmmc2DATA[2]	muxed
SD_2_DATA[3]	IN/OUT	Data for SDMMC2	Xmmc2DATA[3]	muxed
SD_2_DATA[4]	IN/OUT	Data for SDMMC2	Xmmc3DATA[0]	muxed
SD_2_DATA[5]	IN/OUT	Data for SDMMC2	Xmmc3DATA[1]	muxed
SD_2_DATA[6]	IN/OUT	Data for SDMMC2	Xmmc3DATA[2]	muxed
SD_2_DATA[7]	IN/OUT	Data for SDMMC2	Xmmc3DATA[3]	muxed
SD_2_CDn	INPUT	Card Detect for SDMMC2	Xmmc2CDn	muxed
SD_3_CLK	OUTPUT	Clock for SDMMC3	Xmmc3CLK	muxed
SD_3_CMD	IN/OUT	Command for SDMMC3	Xmmc3CMD	muxed
SD_3_DATA[0]	IN/OUT	Data for SDMMC3	Xmmc3DATA[0]	muxed
SD_3_DATA[1]	IN/OUT	Data for SDMMC3	Xmmc3DATA[1]	muxed
SD_3_DATA[2]	IN/OUT	Data for SDMMC3	Xmmc3DATA[2]	muxed
SD_3_DATA[3]	IN/OUT	Data for SDMMC3	Xmmc3DATA[3]	muxed

Signal	I/O	Description	Pad	Type
SD_3_CDn	INPUT	Card Detect for SDMMC3	Xmmc3CDn	muxed

NOTE: It shares SDMMC external pads with CAMIF or SPI. To use these pads for SDMMC, set the GPIO before it starts the SDMMC. Refer to GPIO chapter for correct GPIO settings.

22.9 Register Description

22.9.1 Register Map Summary

- Base Address: 0x1251_0000
- Base Address: 0x1252_0000
- Base Address: 0x1253_0000
- Base Address: 0x1254_0000

Configuration register fields are assigned to one of the attributes described below:

Register	Offset	Description	Reset Value
SDMASYSADn	0x0000	SDMA system address register	0x0
BLKSIZEn	0x0004	Host DMA buffer boundary and transfer block size register	0x0
BLKCNTn	0x0006	Blocks count for current transfer	0x0
ARGUMENTn	0x0008	Command argument register	0x0
TRNMODn	0x000C	Transfer mode setting register	0x0
CMDREGn	0x000E	Command register	0x0
RSPREG0_n	0x0010	Response register 0	0x0
RSPREG1_n	0x0014	Response register 1	0x0
RSPREG2_n	0x0018	Response register 2	0x0
RSPREG3_n	0x001C	Response register 3	0x0
BDATAn	0x0020	Buffer data register	0x0
PRNSTSn	0x0024	Present state register	0x000A_0000
HOSTCTLn	0x0028	Host control register	0x0
PWRCONn	0x0029	Power control register	0x0
BLKGAPn	0x002A	Block gap control register	0x0
WAKCONn	0x002B	Wakeup control register	0x0
CLKCONn	0x002C	Clock control register	0x0
TIMEOUTCONn	0x002E	Timeout control register	0x0
SWRSTn	0x002F	Software reset register	0x0
NORINTSTSn	0x0030	Normal interrupt status register	0x0
ERRINTSTSn	0x0032	Error Interrupt status register	0x0
NORINTSTSENn	0x0034	Normal interrupt status enable register	0x0
ERRINTSTSENn	0x0036	Error interrupt status enable register	0x0
NORINTSIGENn	0x0038	Normal interrupt signal enable register	0x0
ERRINTSIGENn	0x003A	Error interrupt signal enable register	0x0
ACMD12ERRSTSn	0x003C	Auto CMD12 error status register	0x0
CAPAREGn	0x0040	Capabilities register	0x05E8_0080
MAXCURRn	0x0048	Maximum current capabilities register	0x0

Register	Offset	Description	Reset Value
FEAERn	0x0050	Force event auto CMD12 error interrupt register	0x0000
FEERRn	0x0052	Force event error interrupt register error interrupt	0x0000
ADMAERRn	0x0054	ADMA error status register	0x00
ADMASYSADDRn	0x0058	ADMA system address register	0x00
CONTROL2_n	0x0080	Control register 2	0x0
CONTROL3_n	0x0084	FIFO interrupt control (Control register 3)	0x7F5F_3F1F
CONTROL4_n	0x008C	Control register 4	0x0
HCVERn	0x00FE	Host controller version register	0x2401

22.9.1.1 SDMASYSADn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0000, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
SDMASYSAD	[31:0]	RW	SDMA System Address This register contains the system memory address for a DMA transfer. If the Host Controller stops a DMA transfer, this register points to the system address of the next contiguous data position. It is accessed if no transaction is in-progress (that is, after it stops a transaction). Read operations during transfers can return an invalid value. The Host Driver initializes this register before starting a DMA transaction. After it stops DMA, it reads the next system address of the next contiguous data position from this register. The DMA transfer waits at every boundary that Host SDMA Buffer Boundary specifies in the Block Size register. The Host Controller generates DMA Interrupt to request the Host Driver to update this register. The Host Driver set the next system address of the next data position to this register. If it writes the most upper byte of this register (003h), the Host Controller restarts the DMA transfer. If you restart DMA by setting Continue Request in the Block Gap Control register or by setting Resume command, the Host Controller starts at the next contiguous address stored here in the System Address register.	0x00

NOTE: This register contains the physical system memory address used for DMA transfers.

22.9.1.2 BLKSIZEn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0004, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[15]	–	Reserved	0
BUF BOUND	[14:12]	RW	Host DMA Buffer Boundary The large contiguous memory space may not be available in the virtual memory system. To perform long SDMA transfer, it updates System Address register at every system memory boundary during SDMA transfer. These bits specify the size of contiguous buffer in the system memory. The SDMA transfer waits at every boundary specified by these fields and the Host Controller generates the DMA Interrupt to request the Host Driver to update the SDMA System Address register. At the end of transfer, the Host Controller may issue or may not issue DMA Interrupt. Particularly, it does not issue DMA Interrupt after it issues Transfer Complete Interrupt. If this register is set to 0 (buffer size = 4K bytes), lower 12-bit of byte address points data in the contiguous buffer and the upper 20-bit points the location of the buffer in the system memory. The DMA transfer stops if the Host Controller detects carry out of the address from bit 11 to 12. It supports these bits if the SDMA support in the Capabilities register is set to 1. This function is active if DMA Enable in the Transfer Mode register is set to 1. 000b = 4K bytes (Detects A11 carry out) 001b = 8K bytes (Detects A12 carry out) 010b = 16K bytes (Detects A13 carry out 011b = 32K bytes (Detects A14 carry out) 100b = 64K bytes (Detects A15 carry out) 101b = 128K bytes (Detects A16 carry out) 110b = 256K bytes (Detects A17 carry out) 111b = 512K bytes (Detects A18 carry out)	0
BLKSIZE	[11:0]	RW	Transfer Block Size This register specifies the block size of data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53. It sets values ranging from 1 up to maximum buffer size. In case of memory, it is set up to 512 bytes. It is accessed only if no transaction is in-progress (that is, after it stops a transaction). Read operations during transfers return an invalid value. It ignores Write operations. 0200h = 512 Bytes, 01FFh = 511 Bytes : 0004h = 4 Bytes, 0003h = 3 Bytes 0002h = 2 Bytes, 0001h = 1 Byte 0000h = No data transfer	0

NOTE: It uses this register to configure the number of bytes in a data block.

22.9.1.3 BLKCNTn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0006, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
BLKCNT	[15:0]	RW	Blocks Count for Current Transfer It enables this register if Block Count Enable in the Transfer Mode register is set to 1 and this register is valid only for multiple block transfers. The Host Driver sets this register to a value between 1 and the maximum block count. The Host Controller decrements the block count after each block transfer and stops if the count reaches zero. Setting the block count to 0 results in no data blocks being transferred. You should access this register if no transaction is in-progress (that is, after it stops transactions). During data transfer, Read operations on this register return an invalid value and it ignores Write operations. If saving transfer context as a result of a Suspend command, by reading this register it determines the number of blocks are yet to transfer. If restoring transfer context prior to issuing a Resume command, the Host Driver restores the previously saved block count. FFFFh = 65535 blocks : 0002h = 2 blocks 0001h = 1 block 0000h = Stop Count	0

NOTE: It uses this register to configure the number of data blocks.

22.9.1.4 ARGUMENTn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0008, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
ARGUMENT	[31:0]	RW	Command Argument It specifies the SD Command Argument as bit [39:8] of Command-Format in the SD Memory Card Physical Layer Specification.	0

NOTE: This register contains the SD Command Argument.

22.9.1.5 TRNMODn (n = 0 to 3)

- Base Address : 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x000C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[15:14]	–	Reserved	0
BOOTACK	[13]	RW	Boot ACK Receive Enable in Boot mode.	0
BOOTCMD	[12]	RW	Boot Command mode Enable In boot mode, Do Not Enable "Auto CMD12 Enable".	0
RSVD	[11:10]	–	Reserved	0
CCSCON	[9:8]	RW	Command Completion Signal Control 00 = No Command Completion Signal (CCS) Operation (Normal operation and No CE-ATA mode) 01 = Read or Write data transfer CCS enable (Only CE-ATA mode) 10 = Without data transfer CCS enable (Only CE-ATA mode) 11 = Abort Completion Signal (ACS) generation (Only CE-ATA mode)	0
RSVD	[7:6]	–	Reserved	0
MUL1SIN0	[5]	RW	Multi/Single Block Select This bit enables multiple block DAT line data transfers. For any other commands, this bit is set to 0. If this bit is set to 0, it is not mandatory to set the Block Count register. (Refer to Determination of Transfer Type table for more information) 0 = Single Block 1 = Multiple Block	0
RD1WT0	[4]	RW	Data Transfer Direction Select This bit defines the direction of DAT line data transfers. Host Driver sets the bit to 1 to transfer data from the SD card to the SD Host Controller. This bit is set to 0 for all other commands. 0 = Write (Host to Card) 1 = Read (Card to Host)	0
RSVD	[3]	–	Reserved	0
ENACMD12	[2]	RW	Auto CMD12 Enable Multiple block transfers for memory require CMD12 to stop the transaction. If this bit is set to 1 and last block transfer is complete, the Host Controller issues CMD12 automatically. The Host Driver does not set this bit to issue commands that do not require CMD12 to stop data transfer. 0 = Disables Auto CMD 12 1 = Enables Auto CMD 12	0
ENBLKCNT	[1]	RW	Block Count Enable	0

Name	Bit	Type	Description	Reset Value
			It uses this bit to enable the Block Count register, which is only relevant for multiple block transfers. If this bit is set to 0, it disables the Block Count register, which is useful in executing an infinite transfer. (Refer to Determination of Transfer Type table for more information). 0 = Disables 1 = Enables	
ENDMA	[0]	RW	DMA Enable This bit enables DMA functionality. It enables DMA if it supports as it indicates in the DMA Support in the Capabilities register. If it does not support DMA, this bit is meaningless and it always reads as 0. If this bit is set to 1, a DMA operation begins if the Host Driver writes to the upper byte of Command register (0Eh). 0 = Disables 1 = Enables	0

Determination of Transfer Type table lists the summary of how register settings determine types of data transfer.

It uses this register to control the data transfer operations. The Host Driver sets this register before issuing a command which transfers data (refer to Data Present Select in the Command register (*22.9.1.6 CMDREGn (n = 0 to 3)* for more information) or before issuing a Resume command. The Host Driver saves the value of this register if it suspends data transfer (as a result of a Suspend command) and restores it before issuing a Resume command. To prevent data loss, the Host Controller implements Write protection for this register during data transactions. It ignores Write to this register if the Command Inhibit (DAT) in Present State register is set to 1. .

Determination of Transfer Type

Multi/Single Block Select	Block Count Enable	Block Count	Function
0	Don't care	Don't care	Single Transfer
1	0	Don't care	Infinite Transfer
1	1	Not Zero	Multiple Transfer
1	1	Zero	Stop Multiple Transfer

NOTE: For CE-ATA access, it should issue (Auto) CMD12 after Command Completion Signal Disable.

Notification for Boot Mode Operation

NOTE:

1. In the boot mode, do not use Auto Command 12 operation and set Response Type Select field to 2'b00 (no response).
2. Host Controller does not support alternative boot operation using CMD0 with the argument of 0xFFFF_FFFA.
3. After boot code transfer is done, perform byte-write of address 0xD and set it to 0 so that CMD line goes back to HIGH (clears BOOTCMD, BOOTACK filed). Then, wait for minimum of 56 SDCLK cycles as it describes in MMC 4.3 Specification.
4. When BOOTACK is set to 1, Host Controller needs to incur certain error interrupt if ACK pattern is not S-010-E. Currently, simple Data CRC Error occurs. Therefore, it is difficult to figure out whether ACK pattern is wrong.

22.9.1.6 CMDREGn (n = 0 to 3)

- Base Address : 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x000E, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[15:14]	–	Reserved	0
CMDIDX	[13:8]	RW	Command Index It sets these bits to the command number (CMD0-63 and ACMD0-63) that is specified in bits 45-40 of the Command-Format in the SD Memory Card Physical Layer Specification and SDIO Card Specification.	0
CMDTYP	[7:6]	RW	Command Type There are three types of special commands: • Suspend Command • Resume Command • Abort Command It sets these bits to 00b for all other commands. • Suspend Command If the Suspend command succeeds, the Host Controller assumes that the SD Bus has been released and it is possible to issue the next command, which uses the DAT line. The Host Controller de-asserts Read Wait for read transactions and stops verifying busy for write transactions. The interrupt cycle starts, in 4-bit mode. If the Suspend command fails, the Host Controller maintains its current state and the Host Driver restarts the transfer by setting Continue Request in the Block Gap Control register. • Resume Command The Host Driver restarts the data transfer by restoring the registers in the range of 000-00Dh (refer to Suspend and Resume mechanism for more information). The Host Controller verifies for busy before starting Write transfers. • Abort Command If this command is set to active when executing a Read transfer, the Host Controller stops reads to the buffer. If this command is set to active while executing a Write transfer, the Host Controller stops driving the DAT line. After issuing the Abort command, the Host Driver should issue software reset (refer to Abort Transaction (5) for more information). 00b = Other Normal commands 01b = Suspend CMD52 for writing Bus Suspend in CCCR 10b = Resume CMD52 for writing Function Select in CCCR 11b = Abort CMD12, CMD52 for writing I/O Abort in CCCR	0
DATAPRNT	[5]	RW	Data Present Select This bit is set to 1 to indicate that data is present and it transfers the data using the DAT line. It is set to 0 for: (1) Commands using only CMD line (for example, CMD52). (2) Commands with no data transfer but using busy signal on	0

Name	Bit	Type	Description	Reset Value
			DAT[0] line (R1b or R5b ex. CMD38) (3) Resume command 0 = No Data Present 1 = Data Present	
ENCMDIDX	[4]	RW	Command Index Check Enable If this bit is set to 1, the Host Controller checks the Index field in the response to see if it has the same value as the command index. If it does not set this bit, it reports this as a Command Index Error. If this bit is set to 0, it does not check the Index field. 0 = Disables 1 = Enables	0
ENC MDCRC	[3]	RW	Command CRC Check Enable If this bit is set to 1, the Host Controller checks the CRC field in the response. If it detects an error, it reports this as a Command CRC Error. If this bit is set to 0, it does not check the CRC field. The number of bits checked by the CRC field value changes according to the length of the response. 0 = Disables 1 = Enables	0
RSVD	[2]	–	Reserved	0
RSPTYP	[1:0]	RW	Response Type Select 00 = No Response 01 = Response Length 136 10 = Response Length 48 11 = Response Length 48 check Busy after response	0

NOTE: This register contains the SD Command Argument.

The Host Driver checks the Command Inhibit (DAT) bit and Command Inhibit (CMD) bit in the Present State register before writing to this register. Writing to the upper byte of this register triggers SD command generation. The Host Driver is responsible to write this register, because the Host Controller does not protect the writing if Command Inhibit (CMD) is set to active.

Relation between Parameters and the Name of Response Type

These bits determine Response types.

Response Type	Index Check Enable	CRC Check Enable	Name of Response Type
00	0	0	No Response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R6, R5
11	1	1	R1b, R5b

NOTE:

1. In the SDIO specification, it does not define response type notation of R5b. R5 includes R5b in the SDIO specification. However, it defines R5b in this specification to specify the Host Controller checks busy after receiving response. For example, usually it uses CMD52 as R5 but it uses I/O abort command as R5b.
2. For CMD52 to read BS after writing Bus Suspend, Command Type should be "Suspend" as well.

22.9.1.7 RSPREG0_n (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0010, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
CMDRSP	[127:0]	ROC	Command Response Response Bit Definition for Each Response Type table describes the mapping of command responses from the SD Bus to this register for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit range within the Response register. 128-bit Response bit order: {RSPREG3, RSPREG2, RSPREG1, RSPREG0}	0

NOTE: It uses this register to store responses from SD cards.

22.9.1.8 RSPREG1_n (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0014, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
CMDRSP	[127:0]	ROC	Command Response Response Bit Definition for Each Response Type table describes the mapping of command responses from the SD Bus to this register for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit range within the Response register. 128-bit Response bit order: {RSPREG3, RSPREG2, RSPREG1, RSPREG0}	0

NOTE: It uses this register to store responses from SD cards.

22.9.1.9 RSPREG2_n (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0018, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
CMDRSP	[127:0]	ROC	Command Response Response Bit Definition for Each Response Type table describes the mapping of command responses from the SD Bus to this register for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit range within the Response register. 128-bit Response bit order: {RSPREG3, RSPREG2, RSPREG1, RSPREG0}	0

NOTE: It uses this register to store responses from SD cards.

22.9.1.10 RSPREG3_n (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x001C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
CMDRSP	[127:0]	ROC	Command Response Response Bit Definition for Each Response Type table describes the mapping of command responses from the SD Bus to this register for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit range within the Response register. 128-bit Response bit order: {RSPREG3, RSPREG2, RSPREG1, RSPREG0}	0

NOTE: It uses this register to store responses from SD cards.

22.9.1.10.1 Response Bit Definition for Each Response Type

Kind of Response	Meaning of Response	Response Field	Response Register
R1, R1b (normal response)	Card Status	R[39:8]	REP[31:0]
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R[39:8]	REP[127:96]
R2 (CID, CSD register)	CID or CSD register.	R[127:8]	REP[119:0]
R3 (OCR register)	OCR register for memory	R[39:8]	REP[31:0]
R4 (OCR register)	OCR register for I/O and so on	R[39:8]	REP[31:0]
R5, R5b	SDIO response	R[39:8]	REP[31:0]
R6 (Published RCA response)	New published RCA[31:16] and so on.	R[39:8]	REP[31:0]

The Response Field indicates bit positions of Responses defined in the PHYSICAL LAYER SPECIFICATION Version 1.01. The above Table lists that most responses with a length of 48 (R[47:0]) have 32 bits of the response data (R[39:8]) stored in the Response register at REP[31:0]. Responses of type R1b (Auto CMD12 responses) have response data bits R[39:8] stored in the Response register at REP[127:96]. Responses with length 136 (R[135:0]) have 120 bits of the response data (R[127:8]) stored in the Response register at REP[119:0].

To Read the response status efficiently, the Host Controller only stores part of the response data in the Response register. This enables the Host Driver to efficiently read 32 bits of response data in one read cycle on a 32-bit bus system. The Host Controller checks parts of the response, the Index field, and the CRC (as the Command Index Check Enable and the Command CRC Check Enable bits in the Command register specify) and generates an error interrupt if it detects an error. The bit range for the CRC check depends on the response length. If the response length is 48, the Host Controller checks R[47:1], and if the response length is 136 the Host Controller checks R[119:1].

Since the Host Controller may have a multiple block data DAT line transfer executing concurrently with a CMD_wo_DAT command, the Host Controller stores the Auto CMD12 response in the upper bits (REP[127:96]) of the Response register. It stores the CMD_wo_DAT response in REP[31:0]. This allows the Host Controller to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa.

If the Host Controller modifies part of the Response register, as Response Bit Definition for Each Response Type table lists, it preserves the unmodified bits.

22.9.1.11 BDATAn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0020, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
BUFDAT	[31:0]	RW	Buffer Data It accesses the Host Controller buffer through this 32-bit single port SRAM memory. It separates Write and Read memories.	Not fixed

This is bit data port register to access internal buffer.

NOTE: You have to copy the documents in detail from SD Host Standard Specification.

22.9.1.12 PRNSTSn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0024, Reset Value = 0x000A_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0
PRNTCMD	[24]	R/ROC	CMD Line Signal Level (RO) It uses this status for checking and debugging the CMD line level to recover from errors. NOTE: It maps CMD port to SD0_CMD pin	0
PRNTDAT	[23:20]	R/ROC	DAT[3:0] Line Signal Level (RO) It uses this status for checking and debugging the DAT line level to recover from errors. This is especially useful in detecting the busy signal level from DAT[0]. D20: DAT[0] D21: DAT[1] D22: DAT[2] D23: DAT[3] NOTE: It maps DAT port to SD0_DAT pin.	Line State
RSVD	[19]	–	Reserved	1
PRNTCD	[18]	R/ROC	Card Detect Pin Level (RO) This bit reflects the inverse value of the SDCD# pin. It does not perform debouncing on this bit. This bit is valid if Card State Stable is set to 1, but it does not guarantee because of propagation delay. It limits the use of this bit to testing because software should debounce this bit. 0 = No card present (SDCD# = 1) 1 = Card present (SDCD# = 0) NOTE: It maps SDCD# port to SD0_nCD pin and fixes SD2_nCD (Channel 2) port to LOW.	Line State
STBLCARD	[17]	R/ROC	Card State Stable (RO) It uses this bit for testing. If this bit is set to 0, the Card Detect Pin Level is not stable. If this bit is set to 1, it means the Card Detect Pin Level is stable. If this bit is set to 1 and Card Inserted is set to 0 it does not detect No Card state. The Software Reset For All in the Software Reset register does not affect this bit. 0 = Reset or Debouncing. 1 = No Card or Inserted.	1 (After Reset)
INSCARD	[16]	R/ROC	Card Inserted (RO) This bit indicates whether a card has been inserted. The Host Controller debounce this signal so that the Host Driver does not require to wait for it to stabilize. Changing from 0 to 1 generates a Card Insertion interrupt in the Normal Interrupt Status register and changing from 1 to 0 generates a Card Removal interrupt in the Normal Interrupt Status register. The Software Reset For All in the	0

Name	Bit	Type	Description	Reset Value
			Software Reset register does not affect this bit. If it removes a card while its power is on and its clock is oscillating, the Host Controller clears SD Bus Power in the Power Control register and SD Clock Enable in the Clock Control register. If it changes this bit from 1 to 0, the Host Controller immediately stops driving CMD and DAT[3:0] (tri-state). Additionally, the Host Driver should clear the Host Controller by the Software Reset For All in Software Reset register. The card detect is active regardless of the SD Bus Power. 0 = Reset or Debouncing or No Card 1 = Card Inserted.	
RSVD	[15:12]	–	Reserved	–
BUFRDRDY	[11]	R/ROC	Buffer Read Enable (ROC) It uses this status for non-DMA read transfers. The Host Controller implements multiple buffers to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer status. If this bit is set to 1, readable data exists in the buffer. A change of this bit from 1 to 0 occurs if it Reads all the block data from the buffer. A change of this bit from 0 to 1 occurs if block data is ready in the buffer and generates the Buffer Read Ready interrupt. 1 = Enables Read 0 = Disables Read	0
BUFWTRDY	[10]	R/ROC	Buffer Write Enable (ROC) It uses this status for non-DMA write transfers. The Host Controller implements multiple buffers to transfer data efficiently. This read only flag indicates if space is available for write data. If this bit is set to 1, it writes data to the buffer. A change of this bit from 1 to 0 occurs if it Writes all the block data to the buffer. A change of this bit from 0 to 1 occurs if it Writes top of block data to the buffer and generates the Buffer Write Ready interrupt. 0 = Disables Write 1 = Enables Write	0
RDTRANACT	[9]	R/ROC	Read Transfer Active (ROC) It uses this status to detect completion of a read transfer. This bit is set to 1 for either of these conditions: (1) After the end bit of the read command (2) If writing a 1 to Continue Request in the Block Gap Control register to restart a read transfer It clears this bit to 0 for either of these conditions: (1) If the last data block as block length specify transferred to the System. (2) If it transfers all valid data blocks to the System and it does not send current block transfers as a result of the	0

Name	Bit	Type	Description	Reset Value
			Stop At Block Gap Request is set to 1. It generates a Transfer Complete interrupt if this bit changes to 0. 0 = No valid data 1 = Transferring data	
WTTRANACT	[8]	R/ROC	Write Transfer Active (ROC) This status indicates that a write transfer is active. If this bit is set to 0, it means no valid write data exists in the Host Controller. This bit is set in either these cases: (1) After the end bit of the Write command (2) If it writes 1 to Continue Request in the Block Gap Control register to restart a write transfer. It clears this bit in either of these cases: (1) After getting the CRC status of the last data block as the transfer count (Single and Multiple) specifies (2) After getting the CRC status of any block where data transmission is about to stop by a Stop At Block Gap Request. During a write transaction, if it changes this bit to 0, it generates a Block Gap Event interrupt, as a result of the Stop At Block Gap Request being set. This status is useful for the Host Driver to determine the right time to issue commands during write busy. 0 = No valid data 1 = Transfers data	0
RSVD	[7:3]	–	Reserved	0
DATLINEACT	[2]	R/ROC	DAT Line Active (ROC) This bit indicates whether one of the DAT line on SD Bus is in use. • (a) In the case of read transactions This status indicates if a read transfer is in-progress on the SD Bus. Change in this value from 1 to 0 between data blocks generates a Block Gap Event interrupt in the Normal Interrupt Status register. This bit is set in either of these cases: (1) After the end bit of the read command. (2) If it Writes 1 to Continue Request in the Block Gap Control register to restart a Read transfer. It clears this bit in either of these cases: (1) If it sends the end bit of the last data block from the SD Bus to the Host Controller. (2) When beginning a wait, a Stop At Block Gap Request initiates read transfer at a stop at the block gap. The Host Controller waits at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), the Host Controller waits for current block gap by continuing to drive the Read Wait signal. It is necessary to	0

Name	Bit	Type	Description	Reset Value
			support Read Wait to use Suspend/ Resume function. • (b) In the case of write transactions This status indicates that a write transfer is executing on the SD Bus. Change in this value from 1 to 0 generates a Transfer Complete interrupt in the Normal Interrupt Status register. This bit is set in either of these cases: (1) After the end bit of the write command. (2) If it writes 1 to Continue Request in the Block Gap Control register to continue a write transfer. It clears this bit in either of these cases: (1) If the SD card releases write busy of the last data block the Host Controller detects if output is not busy. If SD card does not drive busy signal for 8 SD Clocks, the Host Controller considers the card drive "Not Busy". (2) If the SD card releases write busy prior to waiting for write transfer as a result of a Stop At Block Gap Request. 0 = DAT Line Inactive 1 = DAT Line Active	
CMDINHDAT	[1]	R/ROC	Command Inhibit (DAT) (ROC) It generates this status bit if either the DAT Line Active or the Read Transfer Active is set to 1. If this bit is set to 0, it indicates the Host Controller can issue the next SD Command. Commands with busy signal belong to Command Inhibit (DAT) (for example, R1b and R5b type). NOTE: The SD Host Driver saves registers in the range of 000-00Dh for a suspend transaction after it changes this bit from 1 to 0. 0 = Issues command which uses the DAT line 1 = Cannot issue command which uses the DAT line	0
CMDINHCMD	[0]	R/ROC	Command Inhibit (CMD) (ROC). If this bit is set to 0, it indicates the CMD line is not in use and the Host Controller issues a SD Command using the CMD line. This bit is set immediately after it writes the Command register (00Fh). It clears this bit if it receives the command response. Even if the Command Inhibit (DAT) is set to 1, it issues commands using only the CMD line if this bit is set to 0. Changing from 1 to 0 generates a Command Complete interrupt in the Normal Interrupt Status register. If the Host Controller cannot issue the command because of a command conflict error (refer to Command CRC Error for more information) or because of Command Not Issued By Auto CMD12 Error. This bit remains 1 and the Command Complete is not set. It does not read Status issuing Auto CMD12 from this bit. 0 = Issues command using only CMD line 1 = Cannot issue command	0

The Host Driver can get status of the Host Controller from this 32-bit read only register.

NOTE: You should not assert Buffer Write Enable in Present register for DMA transfers, because it generates Buffer Write Ready interrupt.

Figure 22-17 illustrates the state definitions of hardware that handles "Debouncing".

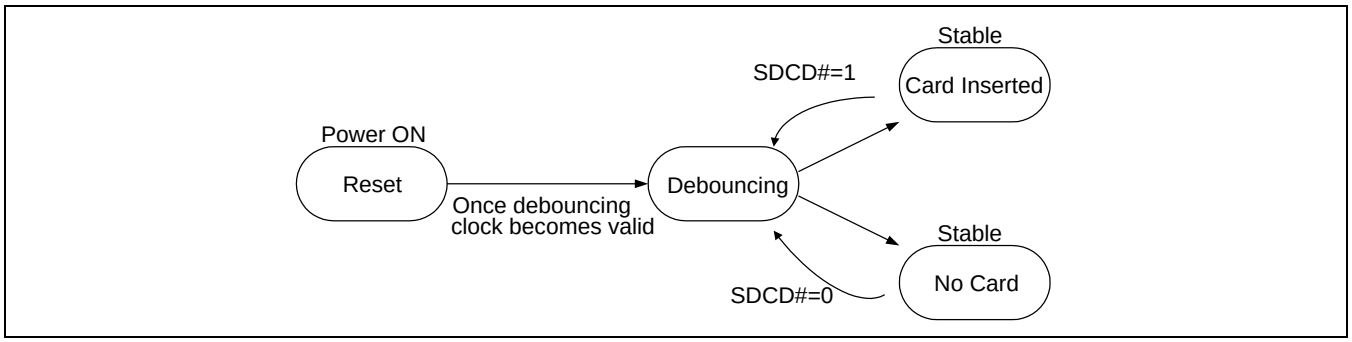


Figure 22-17 Card Detect State

Figure 22-18 illustrates the timing of Command Inhibit (DAT) and Command Inhibit (CMD) with data transfer.

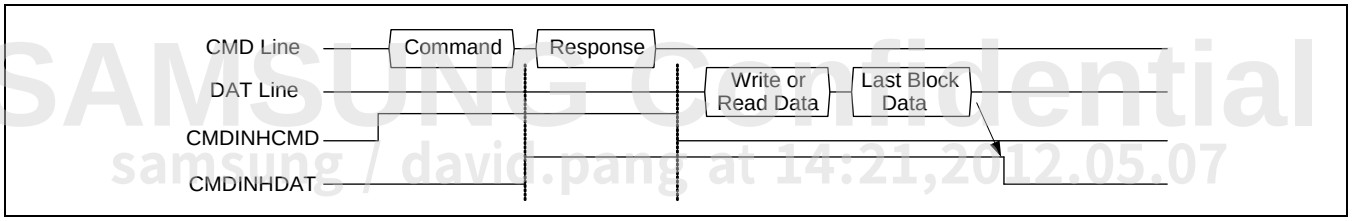


Figure 22-18 Timing of Command Inhibit (DAT) and Command Inhibit (CMD) with Data Transfer

Figure 22-19 illustrates the timing of Command Inhibit (DAT) for the case of response with busy.

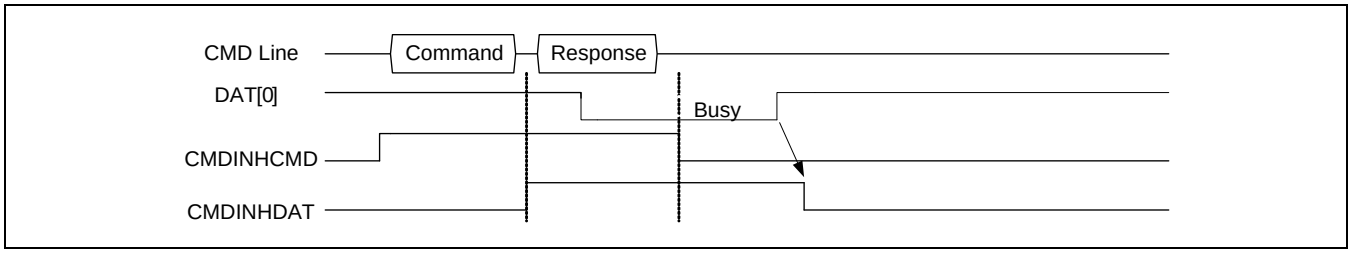


Figure 22-19 Timing of Command Inhibit (DAT) for the Case of Response with Busy

Figure 22-20 illustrates the timing of Command Inhibit (CMD) for the case of no response command.

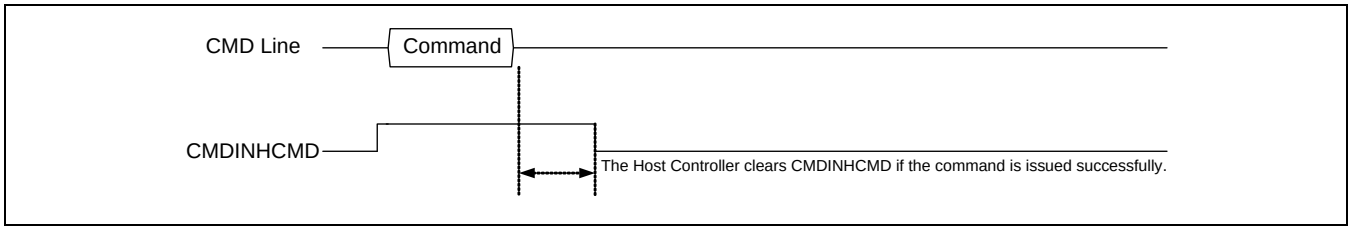


Figure 22-20 Timing of Command Inhibit (CMD) for the Case of No Response Command

22.9.1.13 HOSTCTLn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0028, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7]	–	Reserved It should fix this field to LOW.	0
RSVD	[6]	–	Reserved It should fix this field to LOW.	0
WIDE8	[5]	RW	Extended Data Transfer Width (for MMC 8-bit card) 0 = Bit 1 designates bit width (Data Transfer Width) 1 = 8-bit operation	0
DMASEL	[4:3]	RW	DMA Select It can select one of supported DMA modes. The host driver checks support of DMA modes by referring the Capabilities register. DMA Enable of the Transfer Mode register determines the use of selected DMA. 00 = Selects SDMA 01 = Reserved. 10 = Selects 32-bit Address ADMA2 11 = Selects 64-bit Address ADMA2 (does not support).	0
OUTEDGEINV	[2]	RW	Output Edge Inversion If this bit is set to 0 (default), the Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock. If this bit is set to 1, the Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock 0 = Falling edge output 1 = Rising edge output	0
WIDE4	[1]	RW	Data transfer width This bit selects the data width of the Host Controller. The Host Driver sets it to match the data width of the SD card. 1 = 4-bit mode 0 = 1-bit mode	0
RSVD	[0]	–	Reserved	0

NOTE: Card Detect Pin Level does not reflect SDCD# pin, but selects from SDCD, DAT[3], or CDTestlvl depending on CDSSigSel and SDCDSel values.

22.9.1.14 PWRCONn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0029, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	–	Reserved	–
SELPWRLVL	[3:1]	RW	SD Bus Voltage Select If you set these bits, the Host Driver selects the voltage level for the SD card. Before setting this register, the Host Driver checks the Voltage Support bits in the Capabilities register. If it selects an unsupported voltage, the Host System does not supply SD Bus voltage. 100b to 000b = Reserved 101b = 1.8 V (Typical) 110b = 3.0 V (Typical) 111b = 3.3 V (Typical)	0
PWRON	[0]	RW	SD Bus Power Before setting this bit, the SD Host Driver sets SD Bus Voltage Select. If the Host Controller detects the No Card state, it clears this bit. If it clears this bit, the Host Controller immediately stops driving CMD and DAT[3:0] (tri-state) and drive SDCLK to low level. 0 = Power off 1 = Power on	0

NOTE: This register contains the SD Command Argument.

22.9.1.15 BLKGAPn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x002A, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	–	Reserved	0
ENINTBGAP	[3]	RW	Interrupt at Block Gap This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. If it sets this bit to 1, it enables interrupt detection at the block gap for a multiple block transfer. If it sets this bit to 0, it disables interrupt detection during a multiple block transfer. If the SD card cannot signal an interrupt during a multiple block transfer, it should set this bit to 0. If the Host Driver detects an SD card insertion, it sets this bit according to the CCCR of the SDIO card. (RW) 0 = Disables interrupt at block gap 1 = Enables interrupt at block gap NOTE: This bit should be fixed to 0.	0
ENRWAIT	[2]	RW	Read Wait Control The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the DAT[2] line. Otherwise, the Host Controller has to stop the SD Clock to hold read data, which restricts commands generation. If the Host Driver detects an SD card insertion, it sets this bit according to the CCCR of the SDIO card. If the card does not support read wait, it never sets this bit to 1. Otherwise, DAT line conflict might occur. If this bit is set to 0, it does not support Suspend/ Resume. (RW) 0 = Disables Read Wait Control 1 = Enables Read Wait Control	0
CONTREQ	[1]	RW	Continue Request It uses this bit to restart a transaction which was stopped using the Stop At Block Gap Request. To cancel stop at the block gap, set Stop At Block Gap Request to 0 and set this bit 1 to restart the transfer. The Host Controller automatically clears this bit in either of these cases: (1) If a read transaction, the DAT Line Active changes from 0 to 1 as a read transaction restarts. (2) If a write transaction, the Write Transfer Active changes from 0 to 1 as the write transaction restarts. Therefore, it is not necessary for Host Driver to set this bit to 0. If Stop At Block Gap Request is set to 1, it ignores any write to this bit. (RWAC) 0 = Does not affect continue request 1 = Restarts continue request	0

Name	Bit	Type	Description	Reset Value
STOPBGAP	[0]	RW	Stop at Block Gap Request It uses this bit to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the Transfer Complete is set to 1, indicating a transfer completion the Host Driver leaves this bit as 1. Clearing both the Stop At Block Gap Request and Continue Request does not restart the transaction. Read Wait stops the read transaction at the block gap. The Host Controller honors Stop At Block Gap Request for write transfers, but for read transfers it requires that the SD card support Read Wait. Therefore, the Host Driver does not set this bit during read transfers unless the SD card supports Read Wait and has set Read Wait Control to 1. In the case of write transfers in which the Host Driver writes data to the Buffer Data Port register, the Host Driver sets this bit after all it writes block data. If this bit is set to 1, the Host Driver does not write data to Buffer Data Port register. This bit affects Read Transfer Active, Write Transfer Active, DAT Line Active, and Command Inhibit (DAT) in the Present State register. Regarding detailed control of bits D01 and D00. (RW) 0 = Transfers 1 = Stops	0

There are three cases to restart the transfer after stop at the block gap. Appropriate case depends on whether the Host Controller issues a Suspend command or the SD card accepts the Suspend command.

Three cases to restart the transfer after stop at the block gap are:

1. If the Host Driver does not issue a Suspend command, the Continue Request restarts the transfer.
2. If the Host Driver issues a Suspend command and the SD card accepts it, a Resume command restarts the transfer.
3. If the Host Driver issues a Suspend command and the SD card does not accept it, the Continue Request restarts the transfer.

Any time Stop At Block Gap Request stops the data transfer, the Host Driver waits for Transfer Complete (in the Normal Interrupt Status register) before attempting to restart the transfer. If Continue Request restarts the data transfer, the Host Driver clears Stop At Block Gap Request before or simultaneously.

NOTE: After setting Stop At Block Gap Request field, it should not be cleared unless Block Gap Event or Transfer Complete interrupt occurs. Otherwise, the module hangs.

22.9.1.16 WAKCONn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x002B, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	–	Reserved	0
StaWakeup	[3]	RW	Wakeup Event Status It sets this status if the Card Inserted/Removed or Card Interrupt Stop mode Wakeup Event Occurred. (ROC/RW1C) 0 = Wakeup Interrupt Not occurred or Cleared. 1 = Wakeup Interrupt Occurred.	0
ENWKUPREM	[2]	RW	Wakeup Event Enable on SD Card Removal This bit enables wakeup event through Card Removal assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit. (RW) 0 = Disables 1 = Enables	0
ENWKUPINS	[1]	RW	Wakeup Event Enable on SD Card Insertion This bit enables wakeup event through Card Insertion assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit. (RW) 0 = Disables 1 = Enables	0
ENWKUPINT	[0]	RW	Wakeup Event Enable on Card Interrupt This bit enables wakeup event through Card Interrupt assertion in the Normal Interrupt Status register. This bit is set to 1 if FN_WUS (Wakeup Support) in CIS is set to 1. (RW) 0 = Disables 1 = Enables	0

NOTE: This register is mandatory for the Host Controller, but wakeup functionality depends on the Host Controller system hardware and software. The Host Driver sets SD Bus Power to 1 in the Power Control Register to maintain voltage on the SD Bus, if it desires wakeup event on Card Interrupt.

22.9.1.17 CLKCONn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x002C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[15:4]	–	Reserved	–
STBLEXTCLK	[3]	RW	External Clock Stable This bit is set to 1 if SD Clock output is stable after writing to SD Clock Enable in this register to 1. The SD Host Driver waits to issue command to start until this bit is set to 1. (ROC) 0 = Not ready 1 = Ready	0
ENSDCLK	[2]	RW	SD Clock Enable The Host Controller stops SDCLK if it writes this bit to 0. SDCLK Frequency Select changes if this bit is set to 0. Then, the Host Controller maintains the same clock frequency until it stops SDCLK (Stop at SDCLK=0). If it clears the Card Inserted in the Present State register, this clears the bit (RW). 0 = Disables 1 = Enables	0
STBLINTCLK	[1]	RW	Internal Clock Stable It sets this bit to 1 if SD Clock is stable after writing to Internal Clock Enable in this register to 1. The SD Host Driver waits to set SD Clock Enable until this bit is set to 1. Note: This is useful if it uses PLL for a clock oscillator that requires setup time. (ROC) 0 = Not ready 1 = Ready	0
ENINTCLK	[0]	RW	Internal Clock Enable It sets this bit to 0 if the Host Driver is not using the Host Controller or the Host Controller awaits a wakeup interrupt. The Host Controller should stop its internal clock to go at very low power state. Still, registers is able to Read and Write. Clock starts to oscillate when this bit is set to 1. If clock oscillation is stable, it can set the Host Controller Internal Clock Stable (CLKCON) when this bit is set to 1. This bit does not affect card detection. (RW) 0 = Stops 1 = Oscillates	

NOTE: At the initialization of the Host Controller, the Host Driver sets the SDCLK Frequency Select according to the Capabilities register.

22.9.1.18 TIMEOUTCONn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x002E, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	–	Reserved	0
TIMEOUTCON	[3:0]	RW	Data Timeout Counter Value This value determines the interval by which it detects DAT line timeouts. Refer to the Data Timeout Error in the Error Interrupt Status register for more information on factors that dictate timeout generation. It generates timeout clock frequency by dividing the base clock TMCLK value by this value. While setting this register, prevent inadvertent timeout events by clearing the Data Timeout Error Status Enable (in the Error Interrupt status Enable register). 0000b TMCLK × 213 0001b TMCLK × 214 : 1101b TMCLK × 226 1110b TMCLK × 227 1111b Reserved	0

NOTE: At the initialization of the Host Controller, the Host Driver sets the Data Timeout Counter Value according to the Capabilities register.

22.9.1.19 SWRSTn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x002F, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	–	Reserved	0
RSTDAT	[2]	RW	Software Reset for DAT Line It resets only part of data circuit. It also resets DMA circuit. (RWAC) This bit clears these registers and bits: • Present State register • Buffer Read Enable • Buffer Write Enable • Read Transfer Active • Write Transfer Active • DAT Line Active • Command Inhibit (DAT) • Block Gap Control register • Continue Request • Stop At Block Gap Request • Normal Interrupt Status register • Buffer Read Ready • Buffer Write Ready • DMA Interrupt • Block Gap Event • Transfer Complete 0 = Does not Reset 1 = Resets	0
RSTCMD	[1]	RW	Software Reset for CMD Line It resets only part of command circuit (RWAC). This bit clears these registers and bits: • Buffer Data Port register • Clears and initializes buffer • Present State register • Command Inhibit (CMD) • Normal Interrupt Status register • Command Complete 0 = Does not reset 1 = Resets	0
RSTALL	[0]	RW	Software Reset for All This reset affects the entire Host Controller except for the card detection circuit. It clears register bits of type ROC, RW, RW1C, RWAC, and HWInit to 0. During its initialization, the Host Driver sets this bit to 1 to reset the Host Controller. The Host Controller reset this bit to 0 if capabilities registers are valid and the Host Driver reads them. If this bit is set to 1, the SD card resets itself	0

Name	Bit	Type	Description	Reset Value
			and the Host Driver should reinitialize. (RWAC) 0 = Does not reset 1 = Resets	

NOTE: It generates a reset pulse when writing 1 to each bit of this register. After completing the reset, the Host Controller clears each bit. Because it takes time to complete software reset, the SD Host Driver confirms that these bits are 0.

22.9.1.20 NORINTSTSn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0030, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
STAERR	[15]	ROC/ RW1C	Error Interrupt If it sets any of the bits in the Error Interrupt Status register, then it sets this bit. Therefore, the Host Driver checks this bit first to efficiently tests for an error. This bit is Read only. (ROC) 0 = No Error 1 = Error	0
STAFIA3	[14]	ROC/ RW1C	FIFO SD Address Pointer Interrupt 3 Status (RW1C) 0 = Occurred 1 = Not Occurred If the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 3 values, it asserts this status bit.	0
STAFIA2	[13]	ROC/ RW1C	FIFO SD Address Pointer Interrupt 2 Status (RW1C) 0 = Occurred 1 = Not Occurred If the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 2 values, it asserts this status bit.	0
STAFIA1	[12]	ROC/ RW1C	FIFO SD Address Pointer Interrupt 1 Status (RW1C) 0 = Occurred 1 = Not Occurred If the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 1 value, it asserts this status bit.	0
STAFIA0	[11]	ROC/ RW1C	FIFO SD Address Pointer Interrupt 0 Status (RW1C) 0 = Occurred 1 = Not Occurred If the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 0 value, it asserts this status bit.	0
STARWAIT	[10]	ROC/ RW1C	Read Wait Interrupt Status (RW1C) 0 = Read wait interrupt not occurred 1 = Read wait interrupt occurred NOTE: 1. After checking response for the suspend command, release Read Wait interrupt status manually if BS = 0 (BS means Bus Status field Bus Suspend register in the SDIO card specification). 2. It starts Read Wait operation procedure after 4-SDCLK from the end of the block data read transfer.	0

Name	Bit	Type	Description	Reset Value
STACCS	[9]	ROC/ RW1C	CCS Interrupt Status (RW1C) Command Complete Signal Interrupt Status bit is for CE-ATA interface mode. 0 = CCS Interrupt Occurred 1 = CCS Interrupt Not Occurred	0
STACARDINT	[8]	ROC/ RW1C	Card Interrupt Writing this bit to 1 does not clear this bit. It clears this bit by resetting the SD card interrupt factor. In 1-bit mode, the Host Controller detects the Card Interrupt without SD Clock to support wakeup. In 4-bit mode, it samples the card interrupt signal during the interrupt cycle. Therefore, there are some sample delays between the interrupt signal from the SD card and the interrupt to the Host System. It is necessary to define how to handle this delay. If this status is set and the Host Driver needs to start this interrupt service, Card Interrupt Signal Enable in the Normal Interrupt Signal Enable register must be set to 0 in order to clear the card interrupt status latched in the Host Controller and to stop driving the interrupt signal to the Host System. After completion of the card interrupt service (It should reset interrupt factors in the SD card and it may not assert the interrupt signal), write 1 to clear this register field (RW1C) and set Card Interrupt Signal Enable to 1 to restart sampling the interrupt signal. The Card Interrupt Status Enable should remain set to high. (RW1C) (2) (3) 0 = Does not generate Card Interrupt 1 = Generates Card Interrupt	0
STACARDREM	[7]	ROC/ RW1C	Card removal It sets this status if the Card Inserted in the Present State register changes from 1 to 0. If the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register must be confirmed. Because the card detect state may possibly be changed if the Host Driver clear this bit and interrupt event may not be generated. (RW1C) 0 = Card state stable or Debouncing 1 = Card removed.	0
STACARDINS	[6]	ROC/ RW1C	Card insertion It sets this status if the Card Inserted in the Present State register changes from 0 to 1. If the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register must be confirmed. Because the card detect state may possibly be changed if the Host Driver clear this bit and interrupt event may not be generated. (RW1C). 0 = Card state stable or Debouncing	0

Name	Bit	Type	Description	Reset Value
			1 = Card inserted.	
STABUFRDRDY	[5]	ROC/ RW1C	Buffer Read Ready It sets this status if the Buffer Read Enable changes from 0 to 1. Refer to Buffer Read Enable in the Present State register (9.10) for more information. (RW1C) 0 = Not ready to read buffer 1 = Ready to read buffer	0
STABUFWTRDY	[4]	ROC/ RW1C	Buffer Write Ready It sets this status if the Buffer Write Enable changes from 0 to 1. Refer to Buffer Write Enable in the Present State register (9.10) for more information. (RW1C) 0 = Not ready to write buffer 1 = Ready to write buffer	0
STADMAINT	[3]	ROC/ RW1C	DMA Interrupt It sets this status if the Host Controller detects the Host SDMA Buffer boundary during transfer. Refer to Host SDMA Buffer Boundary in the Block Size register (9.3) for more information. It may add other DMA interrupt factors in the future. In case of ADMA, by setting interrupt field in the descriptor table, Host Controller generates this interrupt. If it uses this for debugging, it does not generate this interrupt after the Transfer Complete. (RW1C) 0 = Does not generate DMA interrupt 1 = Generates DMA interrupt	0
STABLKGAP	[2]	ROC/ RW1C	Block Gap Event If it sets the Stop At Block Gap Request in the Block Gap Control register, it sets this bit if it stops h Read/Write transaction at a block gap. If it does not set Stop At Block Gap Request to 1, it does not set this bit to 1. (RW1C) (1) In the case of a Read Transaction It sets this bit at the falling edge of the DAT Line Active Status (when it stops the transaction at SD Bus timing. It should support the Read Wait to use this function). (2) Case of Write Transaction It sets this bit at the falling edge of Write Transfer Active Status (after getting CRC status at SD Bus timing). 0 = No Block Gap Event 1 = Transaction stopped at block gap	0
STATRANCMPLT	[1]	ROC/ RW1C	Transfer complete It sets this bit if a Read/Write transfer is complete. (1) In the case of a Read Transaction It sets this bit at the falling edge of Read Transfer Active Status. There are two cases in which it generates this interrupt. • If a data transfer is complete as data length specifies	0

<table>
<tr><th>Name</th><th>Bit</th><th>Type</th><th>Description</th><th>Reset Value</th></tr>
<tr><td></td><td></td><td></td><td>(after it reads the last data to the Host System).
• If it stops data at the block gap and complete the data transfer by setting the Stop at Block Gap Request in the Block Gap Control register (after it reads valid data to the Host System).
(2) In the case of a Write Transaction
It sets this bit at the falling edge of the DAT Line Active Status. There are two cases in which it generates this interrupt.
• If it writes the last data to the SD card as data length specifies and releases the busy signal.
• If it stops data transfers at the block gap by setting Stop at Block Gap Request in the Block Gap Control register and data transfers complete. (After it writes valid data to the SD card and releases the busy signal). (RW1C)
The table below shows that Transfer Complete has higher priority than Data Timeout Error. If it sets both bits to 1, it considers the data transfer is complete.
Relation between Transfer Complete and Data is:
<table>
<tr><th>Transfer Complete</th><th>Data Timeout Error</th><th>Meaning of the Status</th></tr>
<tr><td>0</td><td>0</td><td>Interrupted by another factor</td></tr>
<tr><td>0</td><td>1</td><td>Timeout occur during transfer</td></tr>
<tr><td>1</td><td>Don’t care</td><td>Data transfer complete</td></tr>
</table>
0 = No Transfer Complete.
1 = Data Transfer Complete</td><td></td></tr>
<tr><td>STACMDCMPLT</td><td>[0]</td><td>ROC/ RW1C</td><td>Command Complete
It sets this bit when it received the end bit of the command response (Except Auto CMD12). Refer to Command Inhibit (CMD) in the Present State register for more information.
The table shows that Command Timeout Error has higher priority than Command Complete. If it sets both bits to 1, it is considered that the response was not received properly. (RW1C)
<table>
<tr><th>Command Complete</th><th>Command Timeout Error</th><th>Meaning of the Status</th></tr>
<tr><td>0</td><td>0</td><td>Interrupted by another factor</td></tr>
<tr><td>Don't care</td><td>1</td><td>Response not received within 64 SDCLK cycles.</td></tr>
<tr><td>1</td><td>0</td><td>Response received</td></tr>
</table>
0 = No command complete</td><td>0</td></tr>
</table>

Name	Bit	Type	Description	Reset Value
			1 = Command Complete	

The Normal Interrupt Status Enable affects reads of this register, but Normal Interrupt Signal Enable does not affect these Reads. It generates an interrupt if it enables the Normal Interrupt Signal Enable and it sets at least one of the status bits to 1. For all bits except Card Interrupt and Error Interrupt, writing 1 to a bit clear it; writing 0 to a bit, keeps the bit unchanged. It clears more than one status with a single register write. It clears the Card Interrupt if the card stops asserting the interrupt; that is, if the Card Driver services the interrupt condition.

NOTE:

1. Host Driver checks if it actually clears the interrupt by polling or monitoring the INTREQ port. If HCLK is much faster than SDCLK, it takes long time to clear for the bits.
2. Card Interrupt status bit keeps previous value until next card interrupt period (level interrupt) and it clears if write to 1 (RW1C).
3. SD/MMC Controller of the Exynos 4412 SCP does not support card interrupt at block gap used if the multiple block 4-bit operation.

22.9.1.21 ERRINTSTSn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0032, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[15:11]	–	Reserved	0
STABOOTACKERR	[10]	ROC/ RW1C	Boot Ack Error It sets this bit when the Host Controller detects Boot Ack receive error during Boot mode.	0
STAADMAERR	[9]	ROC/ RW1C	ADMA Error It sets this bit if the Host Controller detects errors during ADMA based data transfer. It saves the state of the ADMA at an error occurrence in the ADMA Error Status Register. Additionally, the Host Controller generates this Interrupt if it detects invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the ADMA Error Status indicates that an error occurs in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor. 0 = No Error 1 = Error	0
STAACMDERR	[8]	ROC/ RW1C	Auto CMD12 Error Occurs if it detects that one of the bits in Auto CMD12 Error Status register has changed from 0 to 1. It sets this bit to 1, if the errors in Auto CMD12 occur and if it does not execute Auto CMD12 due to the previous command error. 0 = No Error 1 = Error	0
STACURERR	[7]	ROC/ RW1C	Current Limit Error It is Not implemented in this version. Always 0.	0
STADENDERR	[6]	ROC/ RW1C	Data End Bit Error Occurs if it detects 0 at the end bit position of read data which uses the DAT line or at the end bit position of the CRC Status. 0 = No Error 1 = Error	0
STADATCRCERR	[5]	ROC/ RW1C	Data CRC Error Occurs if it detects CRC error when transferring read data which uses the DAT line or if it detects the Write CRC status having a value of other than "010". 0 = No Error 1 = Error	0
STADATTOUTERR	[4]	ROC/ RW1C	Data Timeout Error Occurs if it detects one of these timeout conditions:	0

Name	Bit	Type	Description	Reset Value
			(1) Busy timeout for R1b, R5b type (2) Busy timeout after Write CRC status (3) Write CRC Status timeout (4) Read Data timeout 0 = No Error 1 = Timeout	
STACMDIDXERR	[3]	ROC/ RW1C	Command Index Error Occurs if a Command Index error occurs in the command response. 0 = No Error 1 = Error	0
STACMDEBITERR	[2]	ROC/ RW1C	Command End Bit Error Occurs if it detects that the end bit of a command response is 0. 0 = No Error 1 = End bit Error generated	
STACMDCRCERR	[1]	ROC/ RW1C	Command CRC Error It generates Command CRC Error in two cases: (1) If it returns a response and sets the Command Timeout Error to 0 (indicating no timeout), this bit is set to 1 if it detects a CRC error in the command response. (2) The Host Controller detects a CMD line conflict by monitoring the CMD line if it issues a command. If the Host Controller drives the CMD line to 1 level, but detects 0 levels on the CMD line at the next SDCLK edge, then the Host Controller aborts the command (stop driving CMD line) and set this bit to 1. The Command Timeout Error also set to 1 to distinguish CMD line conflict. 0 = No Error 1 = Generates CRC error	0
STACMDTOUTERR	[0]	ROC/ RW1C	Command Timeout Error Occurs if it does not return response within 64 SDCLK cycles from the end bit of the command. If the Host Controller detects a CMD line conflict, in which case Command CRC Error also set as described in Table. This bit sets without waiting for 64 SDCLK cycles because the Host Controller aborts command. 0 = No error 1 = Timeout	0

Signals defined in this register are enabled by the Error Interrupt Status Enable register, but not by the Error Interrupt Signal Enable register. It generates interrupt if it enables the Error Interrupt Signal Enable and at least one of the statuses is set to 1. Writing to 1 clears the bit and writing to 0 keeps the bit unchanged. It clears more than one status on one register write.

The relation between Command CRC Error and Command Timeout Error is described in this Table.

The Relation between Command CRC Error and Command Timeout Error

Command CRC Error	Command Timeout Error	Kinds of Error
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

22.9.1.22 NORINTSTSENn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0034, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[15]	RW	Fixed to 0 The Host Driver controls error interrupts using the Error Interrupt Status Enable register. (R)	0
ENSTAFIA3	[14]	RW	FIFO SD Address Pointer Interrupt 3 Status Enable 0 = Masks 1 = Enables	0
ENSTAFIA2	[13]	RW	FIFO SD Address Pointer Interrupt 2 Status Enable 0 = Masks 1 = Enables	0
ENSTAFIA1	[12]	RW	FIFO SD Address Pointer Interrupt 1 Status Enable 0 = Masks 1 = Enables	0
ENSTAFIA0	[11]	RW	FIFO SD Address Pointer Interrupt 0 Status Enable 0 = Masks 1 = Enables	0
ENSTARWAIT	[10]	RW	Read Wait interrupt status enable 0 = Masks 1 = Enables	0
ENSTACCS	[9]	RW	CCS Interrupt Status Enable 0 = Masks 1 = Enables	0
ENSTACARDINT	[8]	RW	Card Interrupt Status Enable If it sets this bit to 0, the Host Controller clears interrupt request to the System. It stops the Card Interrupt detection if it clears this bit and restarts if this bit is set to 1. The Host Driver should clear the Card Interrupt Status Enable before servicing the Card Interrupt and it should set this bit again after it clears all interrupt requests from the card to prevent inadvertent interrupts. 0 = Masks 1 = Enables	0
ENSTACARDREM	[7]	RW	Card Removal Status Enable 0 = Masks 1 = Enables	0
ENSTACARDNS	[6]	RW	Card Insertion Status Enable 0 = Masks 1 = Enables	0
ENSTABUFRDRDY	[5]	RW	Buffer Read Ready Status Enable 0 = Masks	0

Name	Bit	Type	Description	Reset Value
			1 = Enables	
ENSTABUFWTRDY	[4]	RW	Buffer Write Ready Status Enable 0 = Masks 1 = Enables	0
ENSTADMA	[3]	RW	DMA Interrupt Status Enable 0 = Masks 1 = Enables	0
ENSTABLKGAP	[2]	RW	Block Gap Event Status Enable 0 = Masks 1 = Enables	0
ENSTASTANSCMPLT	[1]	RW	Transfer Complete Status Enable 0 = Masks 1 = Enables	0
ENSTACMDCMPLT	[0]	RW	Command Complete Status Enable 0 = Masks 1 = Enables	0

NOTE: Setting to 1 enables Interrupt Status.

22.9.1.23 ERRINTSTSENn (n = 0 to 3)

- Base Address : 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0036, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[15:11]	–	Reserved	0
ENSTABOOTACKERR	[10]	RW	Boot ACK Error Status Enable 0 = Masks 1 = Enables	0
ENSTAADMAERR	[9]	RW	ADMA Error Status Enable 0 = Masks 1 = Enables	0
ENSTAACMDERR	[8]	RW	Auto CMD12 Error Status Enable 0 = Masks 1 = Enables	0
ENSTACURERR	[7]	RW	Current Limit Error Status Enable It does not implement this function in this version. 0 = Masks 1 = Enables	0
ENSTADENDERR	[6]	RW	Data End Bit Error Status Enable 0 = Masked 1 = Enabled	0
ENSTADATCRCERR	[5]	RW	Data CRC Error Status Enable 0 = Masks 1 = Enables	0
ENSTADATTOUTERR	[4]	RW	Data Timeout Error Status Enable 0 = Masks 1 = Enables	0
ENSTACMDIDXERR	[3]	RW	Command Index Error Status Enable 0 = Masked 1 = Enabled	0
ENSTACMDEBITERR	[2]	RW	Command End Bit Error Status Enable 0 = Masks 1 = Enables	0
ENSTACMDCRCERR	[1]	RW	Command CRC Error Status Enable 0 = Masks 1 = Enables	0
ENSTACMDTOUTERR	[0]	RW	Command Timeout Error Status Enable 0 = Masks 1 = Enables	0

NOTE: Setting to 1 enables Error Interrupt Status.

22.9.1.24 NORINTSIGENn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0038, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[15]	RW	Fixed to 0 The Host Driver controls error interrupts using the Error Interrupt Signal Enable register.	0
ENSIGFIA3	[14]	RW	FIFO SD Address Pointer Interrupt 3 Signal Enable 0 = Masks 1 = Enables	0
ENSIGFIA2	[13]	RW	FIFO SD Address Pointer Interrupt 2 Signal Enable 0 = Masks 1 = Enables	0
ENSIGFIA1	[12]	RW	FIFO SD Address Pointer Interrupt 1 Signal Enable. 0 = Masks 1 = Enables	0
ENSIGFIA0	[11]	RW	FIFO SD Address Pointer Interrupt 0 Signal Enable 0 = Masks 1 = Enables	0
ENSIGRWAIT	[10]	RW	Read Wait Interrupt Signal Enable 0 = Masks 1 = Enables	0
ENSIGCCS	[9]	RW	CCS Interrupt Signal Enable Command Complete Signal Interrupt Status bit is for CE-ATA interface mode. 0 = Masks 1 = Enables	0
ENSIGCARDINT	[8]	RW	Card Interrupt Signal Enable 0 = Masks 1 = Enables	0
ENSIGCARDREM	[7]	RW	Card Removal Signal Enable 0 = Masks 1 = Enables	0
ENSIGCARDNS	[6]	RW	Card Insertion Signal Enable 0 = Masks 1 = Enables	0
ENSIGBUFRDRDY	[5]	RW	Buffer Read Ready Signal Enable 0 = Masks 1 = Enables	0
ENSIGBUFWTRDY	[4]	RW	Buffer Write Ready Signal Enable 0 = Masks 1 = Enables	0
ENSIGDMA	[3]	RW	DMA Interrupt Signal Enable	0

Name	Bit	Type	Description	Reset Value
			0 = Masks 1 = Enables	
ENSIGBLKGAP	[2]	RW	Block Gap Event Signal Enable 0 = Masks 1 = Enables	0
ENSIGSTANSCMPLT	[1]	RW	Transfer Complete Signal Enable 0 = Masks 1 = Enables	0
ENSIGCMDCMPLT	[0]	RW	Command Complete Signal Enable 0 = Masks 1 = Enables	0

NOTE: It uses this register to select which interrupt status it indicates to the Host System as the interrupt. These status bits share the same1 bit interrupt line. To enable interrupt, set any of these bits to 1.

22.9.1.25 ERRINTSIGENn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x003A, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[15:11]	–	Reserved	0
ENSIGBOOTACKERR	[10]	RW	Boot ACK Error Signal Enable 0 = Masks 1 = Enables	0
ENSIGADMAERR	[9]	RW	ADMA Error Signal Enable 0 = Masks 1 = Enables	0
ENSIGACMDERR	[8]	RW	Auto CMD12 Error Signal Enable 0 = Masks 1 = Enables	0
ENSIGCURERR	[7]	RW	Current Limit Error Signal Enable It does not implement this function in this version. 0 = Masks 1 = Enables	0
ENSIGDENDERR	[6]	RW	Data End Bit Error Signal Enable 0 = Masks 1 = Enables	0
ENSIGDATCRCERR	[5]	RW	Data CRC Error Signal Enable 0 = Masks 1 = Enables	0
ENSIGDATTOUTERR	[4]	RW	Data Timeout Error Signal Enable 0 = Masks 1 = Enables	0
ENSIGCMDIDXERR	[3]	RW	Command Index Error Signal Enable 0 = Masks 1 = Enables	0
ENSIGCMDEBITERR	[2]	RW	Command End Bit Error Signal Enable 0 = Masks 1 = Enables	0
ENSIGCMDCRCERR	[1]	RW	Command CRC Error Signal Enable 0 = Masks 1 = Enables	0
ENSIGCMDTOUTERR	[0]	RW	Command Timeout Error Signal Enable 0 = Masks 1 = Enables	0

NOTE:

1. It uses this register to select which interrupt status it notifies to the Host System as the interrupt. These status bits share the same 1 bit interrupt line. To enable interrupt generation, set any of these bits to 1.
2. You should copy detailed documents from SD Host Standard Specification.

22.9.1.26 ACMD12ERRSTSn (n = 0 to 3)

- Base Address : 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x003C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[15:8]	–	Reserved	0
STANCMDAER	[7]	ROC	Command not issued by Auto CMD12 Error If the bit is set to 1, it means CMD_wo_DAT does not execute due to an Auto CMD12 Error (D04-D01) in this register. 0 = No error 1 = Not issued	0
RSVD	[6:5]	–	Reserved	0
STACMDIDXERR	[4]	ROC	Auto CMD12 Index Error Occurs if the Command Index error occurs in response to a command. 0 = No error 1 = Error	0
STACMDEBITAER	[3]	ROC	Auto CMD12 End Bit Error. Occurs if it detects that the end bit of command response is 0. 0 = No error 1 = End Bit error generated.	0
STACMDCRCAER	[2]	ROC	Auto CMD12 CRC Error Occurs if it detects a CRC error in the command response. 0 = No error 1 = CRC error generated.	0
STACMDTOUTAER	[1]	ROC	Auto CMD12 Timeout Error Occurs if the card does not return response within 64 SDCLK cycles from the end bit of command. If it sets this bit to1, the other error status bits (D04-D02) are meaningless. 0 = No error 1 = Time out	0
STANACMDAER	[0]	ROC	Auto CMD12 Not Executed When multiple-block data-transfer does not start due to command error, this bit is not set to 1, because it does not need to issue Auto CMD12. If this bit is set to 1, it means the Host Controller cannot issue Auto CMD12 to stop memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (D04-D01) are invalid. 0 = Executed 1 = Not executed	0

If Auto CMD12 Error Status is set to 1, the Host Driver checks this register to identify what kind of error Auto CMD12 indicated. This register is valid if the Auto CMD12 Error is set to 1.

This table describes the relation between Auto CMD12 CRC Error and Auto CMD12 Timeout Error ;

The Relation between Command CRC Error and Command Timeout Error

Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Kinds of Error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

It classifies the timing of changing Auto CMD12 Error Status in three scenarios:

1. If the Host Controller is going to issue Auto CMD12
   - Set D00 to 1 if it cannot issue Auto CMD12 due to an error in the previous command.
   - Set D00 to 0 if it issues Auto CMD12.

2. At the end bit of an Auto CMD12 response.
   - Check received responses by checking the error bits D01, D02, D03, and D04.
   - Set to 1 if it detects error.
   - Set to 0 if it does not detect and error.

3. Before reading the Auto CMD12 Error Status bit D07.
   - Set D07 to 1 if there is a command cannot be issued
   - Set D07 to 0 if there is no command to issue

Timing to generate the Auto CMD12 Error and writing to the Command register are asynchronous. Then it samples D07 if driver never writes to the Command register. Therefore, before reading the Auto CMD12 Error Status register, set the D07 status bit to 1. It generates Auto CMD12 Error Interrupt if it sets one of the error bits D00 to D04 to 1. A command postponed by an occurrence of Auto CMD 12 Error does not generate an interrupt.

22.9.1.27 CAPAREGn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0040, Reset Value = 0x05E8_0080

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	–
CAPAV18	[26]	HWInit	Voltage support 1.8 V (HWInit) 0 = Does not support 1.8 V 1 = Supports 1.8 V	1
CAPAV30	[25]	HWInit	Voltage Support 3.0 V (HWInit) 0 = Does not support 3.0 V 1 = Supports 3.0 V	0
CAPAV33	[24]	HWInit	Voltage Support 3.3 V (HWInit) 0 = Does not support 3.3 V 1 = Supports 3.3 V	1
CAPASUSRES	[23]	HWInit	Suspend/Resume Support (HWInit) This bit indicates if the Host Controller supports Suspend/Resume functionality. If this bit is set to 0, it does not support the Suspend and Resume function and the Host Driver does not issue either Suspend or Resume commands. 0 = Does not support 1 = Supports	1
CAPADMA	[22]	HWInit	DMA Support (HWInit) This bit indicates if the Host Controller is capable of using DMA to transfer data between system memory and the Host Controller directly. 0 = Does not support DMA 1 = Supports DMA	1
CAPAHSPD	[21]	HWInit	High Speed Support (HWInit) This bit indicates if the Host Controller and the Host System support High Speed mode and they can supply SD Clock frequency from 25 to 50 MHz. 0 = Does not support High speed. 1 = Supports High speed.	1
RSVD	[20]	–	Reserved	0
CAPAADMA2	[19]	HWInit	ADMA2 Support This bit indicates if the Host Controller is capable of using ADMA2. 0 = Does not support ADMA2. 1 = Supports ADMA2.	1
RSVD	[28]	–	Reserved	0
CAPAMAXBLKLEN	[17:16]	HWInit	Max Block Length (HWInit) This value indicates the maximum block size that the	0

Name	Bit	Type	Description	Reset Value
			Host Driver can Read and Write to the buffer in the Host Controller. The buffer transfers this block size without wait cycles. It defines three sizes as : 00 = 512 byte 01 = 1024 byte 10 = 2048 byte 11 = Reserved	
RSVD	[15:14]	–	Reserved	0
CAPABASECLK	[13:8]	HWInit	Base Clock Frequency for SD Clock (HWInit) This value indicates the base (maximum) clock frequency for the SD Clock. Unit values are 1 MHz. If the real frequency is 16.5 MHz, the lager value is set to 01 0001b (17 MHz) because the Host Driver use this value to calculate the clock divider value (refer to SDCLK Frequency Select in the Clock Control register for more information) and it does not exceed upper limit of the SD Clock frequency. The supported clock range is 10 to 63 MHz. If these bits are all 0, the Host System has to get information through another method. Not 0 = 1 to 63 MHz 000000b = Get information through another method.	0
CAPATOUTUNIT	[7]	HWInit	Timeout Clock Unit (HWInit) This bit shows the unit of base clock frequency used to detect Data Timeout Error. 0 = kHz 1 = MHz	1
RSVD	[6]	–	Reserved	0
CAPATOUTCLK	[5:0]	HWInit	Timeout Clock Frequency (HWInit) This bit shows the base clock frequency used to detect Data Timeout Error. The Timeout Clock Unit defines the unit of this field value. Timeout Clock Unit = 0[kHz] unit: 1 to 63 kHz Timeout Clock Unit = 1[MHz] unit: 1 to 63 MHz Not 0 = 1 to 63 kHz or 1 MH to 63 MHz 00 0000b = Get information through another method	0

NOTE: When HWINITFIN bit (CONTROL2 register) is set as 0, you can update this register.

This register provides the Host Driver with information specific to the Host Controller implementation. The Host Controller implements these values as fixed or loaded from flash memory during power on initialization. Refer to Software Reset for the Software Reset register for loading from flash memory and completion timing control.

22.9.1.28 MAXCURRn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0048, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	–
MAXCURR18	[23:16]	HWInit	Maximum Current for 1.8 V (HWInit).	0
MAXCURR30	[15:8]	HWInit	Maximum Current for 3.0 V (HWInit).	0
MAXCURR33	[7:0]	HWInit	Maximum Current for 3.3 V (HWInit).	0

When HWINITFIN bit (CONTROL2 register) is set as 0, you can update this register.

These registers indicate maximum current capability for each voltage. The value is meaningful if Voltage Support is set in the Capabilities register. If Host System supplies this information through another method, all Maximum Current Capabilities register is set to 0.

This register measures current in 4 mA steps. This table describes current support of each voltage level.

Maximum Current Value Definition

Register Value	Current Value
0	Get information through another method
1	4 mA
2	8 mA
3	12 mA
…	…
255	1020 mA

22.9.1.29 FEAERn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
RSVD	[15:8]	–	Reserved	0x0
FENCMDAER	[7]	W	Force Event for Command not Issued by Auto CMD12 Error 0 = Does not generate interrupt 1 = Generates interrupt	0
RSVD	[6:5]	–	Reserved	0
FECMDIDXERR	[4]	W	Force Event for Auto CMD12 Index Error 0 = Does not generate interrupt 1 = Generates interrupt.	0
FECMDEBITAER	[3]	W	Force Event for Auto CMD12 End Bit Error 0 = Does not generate interrupt 1 = Generates interrupt	0
FECMDCRCAER	[2]	W	Force Event for Auto CMD12 CRC Error 0 = Does not generate interrupt 1 = Generates interrupt	0
FECMDTOUTAER	[1]	W	Force Event for Auto CMD12 Timeout Error 0 = Does not generate interrupt 1 = Generates interrupt	0
FENACMDAER	[0]	W	Force Event for Auto CMD12 Not Executed 0 = Does not generate interrupt 1 = Generates interrupt	0

NOTE: The Force Event Register is not a physically implemented register. Rather, it is an address at which it can write the Auto CMD12 Error Status Register.

Warning: 0 = No effect
1 = Set each bit of the Auto CMD12 Error Status Register

22.9.1.30 FEERRn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0052, Reset Value = 0x0000

Name	Bit	Type	Description	Reset Value
RSVD	[15:11]	–	Reserved	0x0
FEBOOTACKERR	[10]	W	Force Event for Boot ACK Error 0 = Does not generate interrupt 1 = Generates interrupt	0
FEADMAERR	[9]	W	Force Event for ADMA Error 0 = Does not generate interrupt 1 = Generates iInterrupt	0
FEACMDERR	[8]	W	Force Event for Auto CMD12 Error 0 = Does not generate interrupt 1 = Generates interrupt	0
RSVD	[7]	–	Reserved	0
FEDENDERR	[6]	–	Reserved	0
FEDATCRCERR	[5]	W	Force Event for Data CRC Error 0 = Does not generate interrupt 1 = Generates interrupt	0
FEDATTOUTERR	[4]	W	Force Event for Data Timeout Error 0 = Does not generate interrupt 1 = Generates interrupt	0
FECMDIDXERR	[3]	W	Force Event for Command Index Error 0 = Does not generate interrupt 1 = Generates interrupt	0
FECMDEBITERR	[2]	W	Force Event for Command End Bit Error 0 = Does not generate interrupt 1 = Generates interrupt	0
FECMDCRCERR	[1]	W	Force Event for Command CRC Error 0 = Does not generate interrupt 1 = Generates interrupt	0
FECMDTOUTERR	[0]	W	Force Event for Command Timeout Error 0 = Does not generate interrupt 1 = Generates interrupt	0

The Force Event Register is not a physically implemented register. Rather, it is an address at which the Error Interrupt Status register is written. : If it sets the corresponding bit of the Error Interrupt Status Enable Register, it reflects the effect of a write to this address in the Error Interrupt Status Register.

Warning: 0 = Does not have effect
1 = Sets each bit of the Error Interrupt Status Register

NOTE: By setting this register, you can set the Error Interrupt in the Error Interrupt Status register. To generate interrupt signal, you should set Error Interrupt Status Enable and Error Interrupt Signal Enable.

22.9.1.31 ADMAERRn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0054, Reset Value = 0x00

<table>
<tr><th>Name</th><th>Bit</th><th>Type</th><th>Description</th><th>Reset Value</th></tr>
<tr><td>RSVD</td><td>[31:11]</td><td>–</td><td>Reserved</td><td>0x00</td></tr>
<tr><td>STAADMAFINBLK</td><td>[10]</td><td>RW</td><td>ADMA Final Block Transferred (ROC)
In ADMA operation mode, it sets this field to High if the Transfer Complete condition and the block are final (no block transfer remains).
If this bit is Low when the Transfer Complete condition and Transfer Complete is done due to the Stop at Block Gap, so data to be transferred still remains.</td><td>0</td></tr>
<tr><td>ADMACONTREQ</td><td>[9]</td><td>RW</td><td>ADMA Continue Request (WO)
If it is the stop state by ADMA Interrupt, ADMA operation set this bit to HIGHT to continue.</td><td>0</td></tr>
<tr><td>ADMASTAINT</td><td>[8]</td><td>RW</td><td>ADMA Interrupt Status (RW1C)
It sets this bit to HIGH if it asserts INT attribute in the ADMA Descriptor Table. ADMA error interrupt does not affect this bit.</td><td>0</td></tr>
<tr><td>RSVD</td><td>[7:3]</td><td>–</td><td>Reserved</td><td>0</td></tr>
<tr><td>ADMALENMISERR</td><td>[2]</td><td>RW</td><td>ADMA Length Mismatch Error
This error occurs in these two cases.
(1) While it sets Block Count Enable, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length.
(2) Block length cannot divide total data length.
0 = No Error
1 = Error</td><td>00</td></tr>
<tr><td>ADMAERRST</td><td>[1:0]</td><td>RW</td><td>ADMA Error State
This field indicates the state of ADMA when error is occurred during ADMA data transfer. This field never indicates "10" because ADMA never stops in this state.
<table>
<tr><th>D01-D00</th><th>ADMA Error State when Error is Occurred</th><th>Contents of SYS_SDR Register</th></tr>
<tr><td>00</td><td>ST_STOP (Stops DMA)</td><td>Points next of the error descriptor</td></tr>
<tr><td>01</td><td>ST_FDS (Fetch Descriptor)</td><td>Points the error descriptor</td></tr>
<tr><td>10</td><td>Never set this state</td><td>(not used)</td></tr>
<tr><td>11</td><td>ST_TFR (Transfer Data)</td><td>Points the next of the error descriptor</td></tr>
</table></td><td>0</td></tr>
</table>

If ADMA Error Interrupt occurs, the ADMA Error States field in this register holds the ADMA state. The ADMA System Address Register holds the address around the error descriptor. For recovering the error, the Host Driver requires the ADMA state to identify the error descriptor address as:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address
- ST_CADR: It never sets this state because ADMA error did not generate error in this state.
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address

In case of write operation, the Host Driver should use ACMD22 to get the number of written block rather than using this information, since unwritten data may exist in the Host Controller.

The Host Controller generates the ADMA Error Interrupt if it detects invalid descriptor data (Valid = 0) at the ST_FDS state. In this case, ADMA Error State indicates that an error occurs at ST_FDS state. The Host Driver finds that it does not set the valid bit in the error descriptor.

22.9.1.32 ADMASYSADDRn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0058, Reset Value = 0x00

<table>
<tr><th>Name</th><th>Bit</th><th>Type</th><th>Description</th><th>Reset Value</th></tr>
<tr><td>ADMASYSAD</td><td>[31:0]</td><td>RW</td><td>ADMA System Address
This register holds byte address of executing command of the Descriptor table.
32-bit Address Descriptor uses lower 32-bit of this register. At the start of ADMA, the Host Driver sets start address of the Descriptor table. The ADMA increments this register address, which points to next line, if having fetched a Descriptor line. If it generates the ADMA Error Interrupt, this register holds valid Descriptor address depending on the ADMA state. The Host Driver programs Descriptor Table on 32-bit boundary and sets 32-bit boundary address to this register. ADMA2 ignores lower 2-bit of this register and assumes it to be 00b.
32-bit Address ADMA
<table>
<tr><th>Register Value</th><th>32-bit System Address</th></tr>
<tr><td>xxxxxxxx 00000000h</td><td>00000000h</td></tr>
<tr><td>xxxxxxxx 00000004h</td><td>00000004h</td></tr>
<tr><td>xxxxxxxx 00000008h</td><td>00000008h</td></tr>
<tr><td>xxxxxxxx 0000000Ch</td><td>0000000Ch</td></tr>
<tr><td>.....</td><td>.....</td></tr>
<tr><td>xxxxxxxx FFFFFFFCh</td><td>FFFFFFFCh</td></tr>
</table>
NOTE: The data length of the ADMA Descriptor Table should be the word unit (multiple of the 4-byte).</td><td>00</td></tr>
</table>

NOTE: This register contains the physical descriptor address used for ADMA data transfer.

22.9.1.33 CONTROL2_n (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0080, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
ENSTAASYNCCLR	[31]	RW	Write Status Clear Async Mode Enable This bit makes async-clear enable about Normal and Error interrupt status bit. During the initialization procedure command operation, you should enable this bit. 0 = Disables 1 = Enables	0
ENCMDCNFMSK	[30]	RW	Command Conflict Mask Enable This bit can mask enable the Command Conflict Status (bit [1:0] of the ERROR INTERRUPT STATUS REGISTER). 0 = Disables Mask 1 = Enables Mask NOTE: If it sets the OUTEDGEINV field in the Host Control Register (High Speed data transfer), you should enable this field to prevent from command conflict status alarm.	0
RSVD	[29]	–	Reserved (must be 1'b0)	0
SELCARDOUT	[28]	RW	Card Removed Condition Selection 0 = Card Removed condition is "Not Card Insert" State (when the transition from "Card Inserted" state to "Debouncing" state in *Figure 22-17*). 1 = Card Removed state is "Card Out" State (if the transition from "Debouncing" state to "No Card" state in *Figure 22-17*).	0
FLTCLKSEL	[27:24]	RW	Filter Clock (iFLTCLK) Selection Filter Clock period = $2^{(FltClkSel + 5)}$ × iSDCLK period 0000 = 25 × iSDCLK 0001 = 26 × iSDCLK : 1111 = 220 × iSDCLK	0
LVLDAT	[23:16]	RW	DAT Line LEVEL Bit[23] = DAT[7] Bit[22] = DAT[6] Bit[21] = DAT[5] Bit[20] = DAT[4] Bit[19] = DAT[3] Bit[18] = DAT[2] Bit[17] = DAT[1] Bit[16] = DAT[0] (Read Only)	Line state
ENFBCLKTX	[15]	RW	Feedback Clock Enable for Tx Data/Command Clock 0 = Disables	0

Name	Bit	Type	Description	Reset Value
			1 = Enables	
ENFBCLKRX	[14]	RW	Feedback Clock Enable for Rx Data/Command Clock 0 = Disables 1 = Enables	0
RSVD	[13]	–	Reserved (must be 1'b0)	0
SDOPSIGPC	[12]	RW	SD Output signal Power Control Support If this field is set to 1, it enables output CMD and DAT referencing to SD Bus Power bit in the "PWRCON register". 0 = SD Bus Power bit does not control CMD and DAT outputs 1 = SD Bus Power bit controls CMD and DAT outputs (masked). NOTE: It does not implement this function in this version.	0
ENBUSYCHKTXSTART	[11]	RW	CE-ATA I/F mode Busy state check before Tx Data start state. 0 = Disables 1 = Enables	0
DFCNT	[10:9]	RW	Debounce Filter Count. Debounce Filter Count setting register for Card Detect signal input (SDCD#). 00 = Does not use debounce filter 01 = 4 × iFLTCLK 10 = 16 × iFLTCLK 11 = 64 × iFLTCLK	0
ENCLKOUTHOLD	[8]	RW	SDCLK Hold Enable Host Controller performs enter and exit of the SDCLK Hold state. 0 = Disables 1 = Enables NOTE: This field should be 1.	0
RWAITMODE	[7]	RW	Read Wait Release Control 0 = The Host Controller (Auto) releases Read Wait state. 1 = Host Driver (Manual) releases Read Wait state.	0
DISBUFRD	[6]	RW	Buffer Read Disable 0 = Normal mode, user can read buffer (FIFO) data using 0x20 register. 1 = User cannot read buffer (FIFO) data using 0x20 register. In this case, it reads the buffer memory through memory area (debug purpose).	0
RSVD	[5:4]	–	Reserved	–
SDINPSIGPC	[3]	RW	SD Input Signal Power Control Support If this field is set to 1, it enables input CMD and DAT referencing to SD Bus Power bit in the "PWRCON" register.	0

Name	Bit	Type	Description	Reset Value
			0 = No Sync, no switch input enable signal (Command, Data). 1 = Sync, control input enable signal (Command, Data). NOTE: It does implement this function in this version.	
RSVD	[2]	–	Reserved	0
ENCLKOUTMSKCON	[1]	RW	SDCLK Output Clock Masking When Card Insert Cleared If this field is High, then it uses this field to not to stop SDCLK if No Card state. 0 = Disables 1 = Enables	0
HWINITFIN	[0]	RW	SD Host Controller Hardware Initialization Finish 0 = Does not finish 1 = Finishes	0

NOTE:

1. Ensure to set SDCLK Hold Enable (EnSCHold) if the card does not support Read Wait to guarantee for Receive data as it does not overwrite to the internal FIFO memory.
2. It prohibits CMD_wo_DAT issue during READ transfer if it sets SDCLK Hold Enable.

22.9.1.34 CONTROL3_n (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x0084, Reset Value = 0x7F5F_3F1F

Name	Bit	Type	Description	Reset Value
FCSEL3	[31]	RW	Feedback Clock Select[3] Reference (1)	0x0
FIA3	[30:24]	RW	FIFO Interrupt Address Register 3 FIFO (512 byte buffer memory, word address unit) Initial value (0x7F) generates at 512 byte (128-word) position.	0x7F
FCSEL2	[23]	RW	Feedback Clock Select[2] Reference (1)	0x0
FIA2	[22:16]	RW	FIFO Interrupt Address Register 2 FIFO (512 byte buffer memory, word address unit) Initial value (0x5F) generates at 384 byte (96-word) position.	0x5F
FCSEL1	[15]	RW	Feedback Clock Select[1] Reference (2)	0x0
FIA1	[14:8]	RW	FIFO Interrupt Address Register 1 FIFO (512 byte buffer memory, word address unit) Initial value (0x3F) generates at 256 byte (64-word) position.	0x3F
FCSEL0	[7]	RW	Feedback Clock Select[0] Reference (2)	0x0
FIA0	[6:0]	RW	FIFO Interrupt Address Register 0 FIFO (512 byte buffer memory, word address unit) Initial value (0x1F) generates at 128 byte (32-word) position.	0x1F

NOTE:

1. FCSel[3:2]: Tx Feedback Clock Delay Control: Inverter delay means 10ns delay if SDCLK 50 MHz setting "00", "01" = Inverter delay, "10", "11" = basic delay (around 2 ns)
2. FCSel[1:0]: Rx Feedback Clock Delay Control: Inverter delay means10ns delay if SDCLK 50 MHz setting "00", "01" = Inverter delay, "10", "11" = basic delay (around 2 ns)
3. Tx Feedback inversion setting (FCSel[3:2] = "00" or "01"), Tx Feedback clock enable (ENFBCLKTX = 0) and Normal Speed mode (OUTEDGEINV = 0) setting make Tx data transfer mismatch (Do not set).

22.9.1.35 CONTROL4_n (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x008C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0
STABLKGAP BUSY	[1]	RW	Status Block Gap Access Busy This bit is "High" when the clock domain crossing (HCLK to SDCLK) operation is processing when the write operation to the BLKGAP register. This bit is status bit and Read-Only (RO)	0
STABUSY	[0]	RW	Status Busy This bit is "High" if the clock domain crossing (HCLK to SDCLK) operation is under process. This bit is status bit and Read Only (RO)	0

22.9.1.36 HCVERn (n = 0 to 3)

- Base Address: 0x1251_0000, 0x1252_0000, 0x1253_0000, 0x1254_0000
- Address = Base Address + 0x00FE, Reset Value = 0x2401

Name	Bit	Type	Description	Reset Value
VENVER	[15:8]	HWInit	Vendor Version Number It receives this status for the vendor version number. The Host Driver should not use this status. 0x24: SDMMC4.2 Host Controller.	0x24
SPECVER	[7:0]	HWInit	Specification Version Number This status indicates the Host Controller Specification Version. The upper and lower 4 bits indicate the version. 00 = SD Host specification version 1.0. 01 = SD Host specification version 2.0 including the feature of the ADMA and test register. Others = Reserved	0x01

23 Mobile Storage Host

23.1 Overview

This chapter describes Secure Digital (SD/SDIO), MultiMediaCard (MMC), CE-ATA host controller and related registers that Exynos 4412 SCP RISC microprocessor supports.

The Mobile Storage Host is an interface between system and SD/MMC card. The performance of this host is very powerful, as clock rate is 50 MHz and access 8-bit data pin simultaneously. This host supports 8-bit DDR (Double Data Rate) transfer.

The specifications that Mobile Storage Host supports are:

- Secure Digital Memory Card (SD Memory Card, Version 2.0)
- Secure Digital I/O (SDIO – Version 2.0)
- Consumer Electronics Advanced Transport Architecture (CE-ATA – Version 1.1)
- Multimedia Cards (MMC – Version 4.41)

23.2 Features of Mobile Storage Host

The features of the Mobile Storage Host are:

- Supports Secure Digital memory protocol commands
- Supports Secure Digital I/O protocol commands
- Supports Multimedia Card protocol commands
- Supports CE-ATA digital protocol commands
- Supports Command Completion signal and interrupt to host processor
- Command Completion Signal disable feature

The features of MMC4.41 that Mobile Storage Host supports are:

- DDR in 4-bit and 8-bit mode
- GO_PRE_IDLE_STATE command (CMD with argument 0xF0F0F0F0)
- New EXTCSD registers
- Hardware Reset that eMMC4.41 supports:

The feature of MMC4.41 that Mobile Storage Host does not support is:

- Boot in DDR mode

23.3 Block Diagram of Mobile Storage Host

The DWC_mobile_storage consists of main functional blocks that are illustrated in *Figure 23-1 the blocks are:*

- Bus Interface Unit (BIU): Provides AMBA AHB/APB and DMA interfaces for register and data read/writes.
- Card Interface Unit (CIU): Takes care of SD_MMC_CEATA protocols and provides clock management.

The BIU provides the host interface to the registers. It also provides the data FIFO through the Host Interface Unit (HIU). Additionally, it provides independent data FIFO access through a DMA interface. You can configure host interface as either an AMBA APB or an AMBA AHB slave interface.

The IDMAC is responsible for exchanging data between FIFO and the host memory. A set of IDMAC registers is accessible by host for controlling the IDMAC operation through the AMBA AHB/AMBA APB slave interface.

The DWC_mobile_storage CIU controls the card-specific protocols. Within CIU, the command path control unit and data path control unit interface with the controller to the command and data ports of the SD_MMC_CEATA cards. The CIU also provides clock control.

Figure 23-1 illustrates the block diagram of Mobile Storage Host.

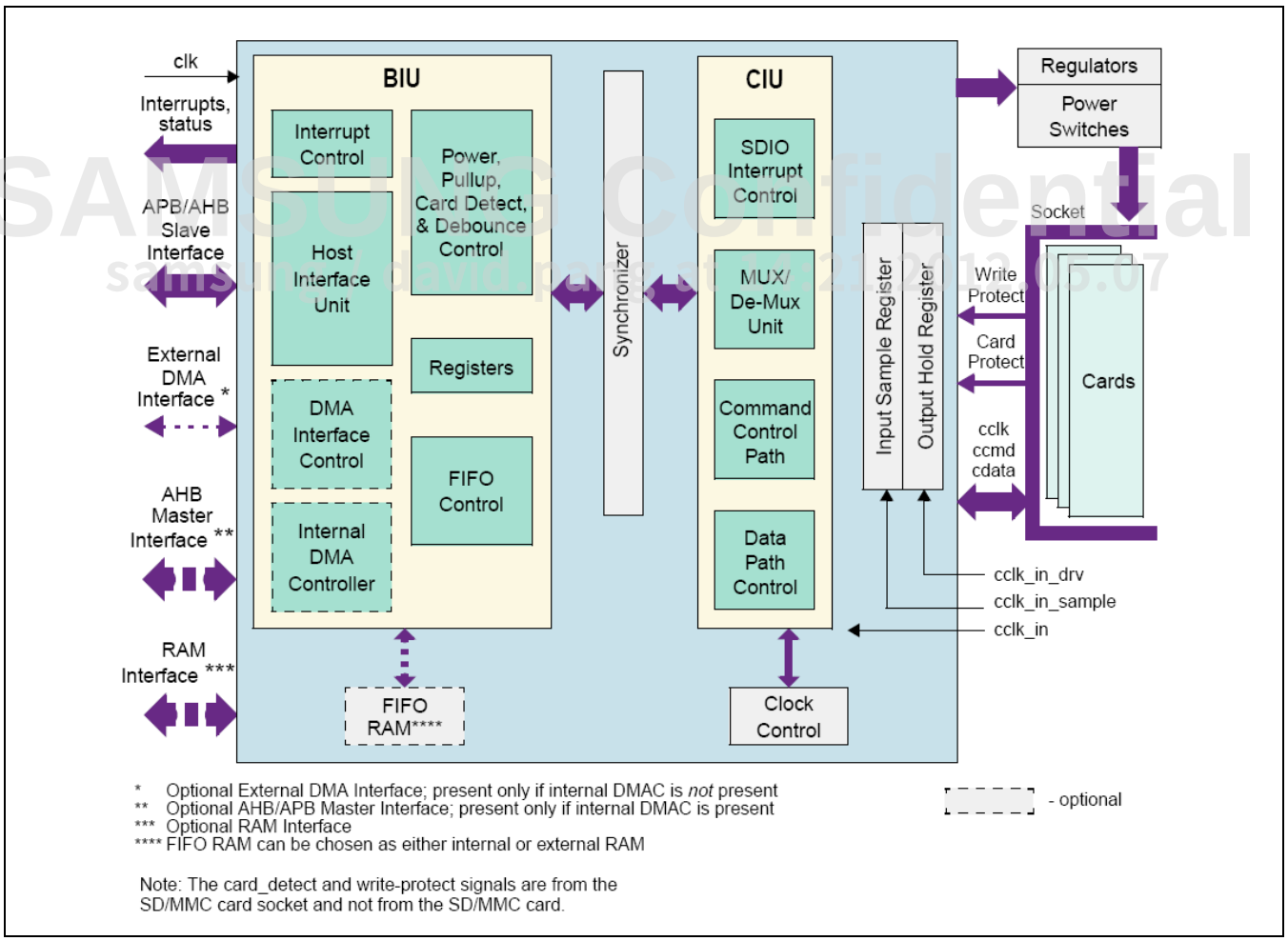


Figure 23-1 Block Diagram of Mobile Storage Host

23.4 Clock Phase Shifter

SD/MMC card receives DATA/CMD with card clock from the host controller. To synchronize the clock and DATA/CMD, it is required that the clock delay is inserted to the Tx/Rx clock path. For this to happen, the logic is added to the design as illustrated in the *Figure 23-2*. And clock selection can be done with CLKSEL register at the end of this chapter.

Figure 23-2 illustrates the clock phase shifter.

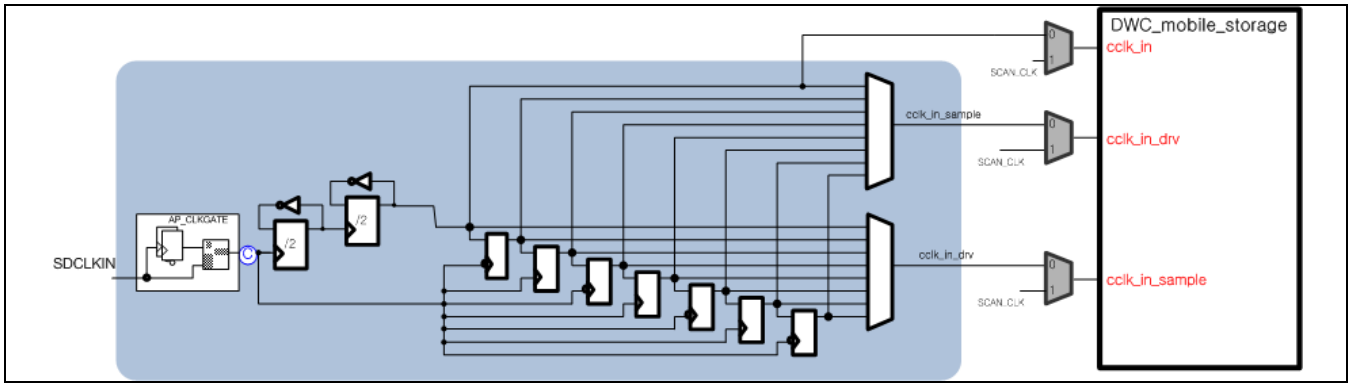


Figure 23-2 Clock Phase Shifter

This clock phase shifter makes 0, 45, 90, 135, 180, 225, 270, 315 phase shifted clocks for Tx/Rx respectively. To make 50 MHz phase shifted clock, SDCLKIN should be 200 MHz. For 8-bit DDR mode, SDCLKIN should be 400 MHz because internal logic should be operated with 100 MHz clock only in 8-bit DDR mode.

23.5 I/O Description

Signal	I/O	Description	Pad	Type
SD_4_CLK	OUTPUT	Clock for mobile storage host	Xmmc0CLK	muxed
SD_4_CMD	IN/OUT	Command for mobile storage host	Xmmc0CMD	muxed
SD_4_DATA[0]	IN/OUT	Data for mobile storage host	Xmmc0DATA[0]	muxed
SD_4_DATA[1]	IN/OUT	Data for mobile storage host	Xmmc0DATA[1]	muxed
SD_4_DATA[2]	IN/OUT	Data for mobile storage host	Xmmc0DATA[2]	muxed
SD_4_DATA[3]	IN/OUT	Data for mobile storage host	Xmmc0DATA[3]	muxed
SD_4_DATA[4]	IN/OUT	Data for mobile storage host	Xmmc1DATA[0]	muxed
SD_4_DATA[5]	IN/OUT	Data for mobile storage host	Xmmc1DATA[1]	muxed
SD_4_DATA[6]	IN/OUT	Data for mobile storage host	Xmmc1DATA[2]	muxed
SD_4_DATA[7]	IN/OUT	Data for mobile storage host	Xmmc1DATA[3]	Muxed
SD_4_CDn	INPUT	Card detect for mobile storage host	Xmmc0CDn	muxed

23.6 Register Description

23.6.1 Register Map Summary

- Base Address: 0x1255_0000
  Configuration register fields are assigned to one of the attributes described below:

Register	Offset	Description	Reset Value
CTRL	0x0000	Control register	0x0
PWREN	0x0004	Power-enable register	0x0
CLKDIV	0x0008	Clock-divider register	0x0
CLKSRC	0x000C	Clock-source register	0x0
CLKENA	0x0010	Clock-enable register	0x0
TMOUT	0x0014	Time-out register (Number of card clock output clocks-cclk_out)	0xFFFF_FF40
CTYPE	0x0018	Card-type register	0x0
BLKSIZ	0x001C	Block-size register	0x200
BYTCNT	0x0020	Byte-count register	0x200
INTMASK	0x0024	Interrupt-mask register	0x0
CMDARG	0x0028	Command-argument register	0x0
CMD	0x002C	Command register	0x2000_0000
RESP0	0x0030	Response-0 register	0x0
RESP1	0x0034	Response-1 register	0x0
RESP2	0x0038	Response-2 register	0x0
RESP3	0x003C	Response-3 register	0x0
MINTSTS	0x0040	Masks interrupt-status register	0x0
RINTSTS	0x0044	Raw interrupt-status register	0x0
STATUS	0x0048	Status register; mainly for debug purposes	{20'h00000, 4'b00xx,8'h06}
FIFOTH	0x004C	FIFO threshold register	{4'h0, 12'h7f,16'h0}
CDETECT	0x0050	Card-detect register	Value in card detect signal
WRTPRT	0x0054	Write-protect register	Value in card_write_prt signal
GPIO	0x0058	GPIO register	{24'h0, gp_in signal}
TCBCNT	0x005C	Transferred CIU card byte count	0x0
TBBCNT	0x0060	Transferred host/DMA to/from BIU-FIFO byte count	0x0
DEBNCE	0x0064	Card detect debounce register (Number of host clocks-clk)	0x00FF_FFFF
USRID	0x0068	User ID register	UID_REG configuration parameter

Register	Offset	Description	Reset Value
VERID	0x006C	Synopsys version ID register	0x5342_240A
HCON	0x0070	Hardware configuration register	Dependent on user-defined values in configuration parameters
UHS_REG	0x0074	UHS-1 register	0x0
BMOD	0x0080	Bus mode register; controls host Interface mode.	0x0
PLDMND	0x0084	Poll demand register. Used by the host to instruct IDMAC to poll descriptor list while in suspend.	0x0
DBADDR	0x0088	Descriptor list base address register. Points IDMAC to the start of descriptor list.	0x0
IDSTS	0x008C	Internal DMAC status register. Software driver application reads this register during interrupt service routine or polling to determine status of IDMAC.	0x0
IDINTEN	0x0090	Internal DMAC interrupt enable register. Enables the interrupts reported by the IDMAC status register (IDSTS).	0x0
DSCADDR	0x0094	Current host descriptor address register. Points to the start of current descriptor read by the IDMAC.	0x0
BUFADDR	0x0098	Current host buffer address register. Points to the current buffer address accessed by the IDMAC.	0x0
CLKSEL	0x009C	DRV/sample clock selection register.	0x0
RSVD	0x9C-FF	Reserved; although AHB/APB access to this region completes, write data is ignored and read data is 32’hx.	–
DATA	>=0x100	Data FIFO read/write; if address is equal or greater than 0x100, then FIFO is selected as long as device is selected (hsel/psel active)	32'hx

23.6.1.1 CTRL

- Base Address: 0x1255_0000
- Address = Base Address + 0x0000, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	–
use_internal_dmac	[25]	RW	Present only for the Internal DMAC configuration; else, it is reserved. 0 = Host performs data transfers through the slave interface 1 = Internal DMAC used for data transfer	0
enable_OD_pullup	[24]	RW	External open-drain pull up: 0 = Disables 1 = Enables Inverted value of this bit is output to ccmd_od_pullup_en_n port. When bit is set, command output always driven in open-drive mode. It means DWC_mobile_storage drives either 0 or high impedance, and does not drive hard 1.	1
Card_voltage_b	[23:20]	RW	Card regulator-B voltage setting; output to card_volt_b port. Optional feature; ports can be used as general-purpose outputs.	0
Card_voltage_a	[19:16]	RW	Card regulator-A voltage setting; output to card_volt_a port. Optional feature; ports can be used as general-purpose outputs.	0
RSVD	[15:12]	–	Reserved	0
ceata_device_interrupt_status	[11]	RW	Interrupt status of CE-ATA device 0 = Interrupts not enabled in CE-ATA device (nIEN = 1 In ATA control register) 1 = Interrupts are enabled in CE-ATA device (nIEN = 0 In ATA control register) Software should appropriately write to this bit after power-on reset or any other reset to CE-ATA device. After reset, usually CE-ATA device interrupt is disabled (nIEN = 1). If the host enables CE-ATA device interrupt, then software should set this bit.	0
send_auto_stop_ccsd	[10]	RW	Send auto stop CCSD 0 = Clears bit if DWC_mobile_storage does not reset the bit. 1 = Sends internally generated STOP after sending CCSD to CE-ATA device. NOTE: Always set send_auto_stop_ccsd and send_ccsd bits together. send_auto_stop_ccsd should not be set independent of send_ccsd. When set, DWC_Mobile_Storage automatically sends internally-generated STOP command (CMD12) to CE-ATA device. After sending internally-generated STOP command, Auto Command Done (ACD) bit in RINTSTS is set and generates interrupt to host if Auto Command Done interrupt is not masked. After sending CCSD, DWC_mobile_storage	0

Name	Bit	Type	Description	Reset Value
			automatically clears send_auto_stop_ccsd bit.	
send_ccsd	[9]	RW	Send CCSD 0 = Clears bit if DWC_mobile_storage does not reset the bit. 1 = Send Command Completion Signal Disable (CCSD) to CE-ATA device When set, DWC_mobile_storage sends CCSD to CE-ATA device. Software sets this bit only if current command is expecting CCS (that is, RW_BLK) and interrupts are enabled in CE-ATA device. Once CCSD pattern is sent to device, DWC_mobile_storage automatically clears send_ccsd bit. It also sets Command Done (CD) bit in RINTSTS register and generates interrupt to host if Command Done interrupt is not masked. NOTE: Once send_ccsd bit is set, it takes two card clock cycles to drive CCSD on CMD line. Due to this, during boundary conditions, it may send CCSD to CE-ATA device, even if device signals CCS.	0
abort_read_data	[8]	RW	Abort read data 0 = No change 1 = After suspend command is issued during read-transfer, software polls card to find when suspend happens. Once suspend occurs, software sets bit to reset data state-machine, which is waiting for next block of data. Bit automatically clears once data state-machine and resets to idle. Use SDIO in card suspend sequence.	0
send_irq_response	[7]	RW	Send IRQ response 0 = No change 1 = Send auto IRQ response Bit automatically clears once response is sent. To wait for MMC card interrupts, host issues CMD40. DWC_mobile_storage waits for interrupt response from MMC card(s). Meanwhile, if host wants DWC_mobile_storage to exit waiting for interrupt state, it sets this bit. at which time DWC_mobile_storage command state-machine sends CMD40 response on bus and returns to idle state.	0
read_wait	[6]	RW	Read wait 0 = Clears read wait 1 = Asserts read wait For sending read-wait to SDIO cards.	0
dma_enable	[5]	RW	DMA enable 0 = Disables DMA transfer mode 1 = Enables DMA transfer mode Valid only when DWC_mobile_storage configured for External DMA interface. Even when DMA mode is enabled, host can still push/pop data into or from FIFO. This should not happen during normal operation. If there is simultaneous FIFO access from	0

Name	Bit	Type	Description	Reset Value
			host/DMA, the data coherency is lost. Also, there is no arbitration inside DWC_mobile_storage to prioritize simultaneous host/DMA access.	
int_enable	[4]	RW	Global interrupt enable/disable bit: 0 = Disables interrupts 1 = Enables interrupts The int port is 1 only when this bit is 1 and one or more unmasked interrupts are set.	0
RSVD	[3]	–	Reserved	0
dma_reset	[2]	RW	DMA reset 0 = No change 1 = Resets internal DMA interface control logic To reset DMA interface, firmware sets bit to 1. This bit auto-clears after two AHB clocks.	0
fifo_reset	[1]	RW	FIFO reset 0 = No change 1 = Reset to data FIFO To reset FIFO pointers To reset FIFO, firmware sets bit to 1. This bit is auto-cleared after completion of reset operation.	0
controller_ reset	[0]	RW	Controller Reset 0 = No change 1 = Resets DWC_mobile_storage controller To reset controller, firmware sets bit to 1. This bit is auto-cleared after two AHB and two cclk_in clock cycles. This resets: • BIU/CIU interface • CIU and state machines • abort_read_data, send_irq_response, and read_wait bits of Control register • start_cmd bit of Command register Does not affect any registers or DMA interface, or FIFO or host interrupts.	0

23.6.1.2 PWREN

- Base Address: 0x1255_0000
- Address = Base Address + 0x0004, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	–
power_enable	[29:0]	RW	Power on/off switch for up to 16 cards. For example, bit[0] controls card 0. Once power is turned on, firmware waits for regulator/switch ramp-up time before trying to initialize card. 0 = Power off 1 = Power on Only NUM_CARDS number of bits are implemented. Bit values output to card_power_en port. Optional feature; ports can be used as general-purpose outputs.	0

23.6.1.3 CLKDIV

- Base Address: 0x1255_0000
- Address = Base Address + 0x0008, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
clk_divider3	[31:24]	RW	Clock divider-3 value. Clock division is 2 × n. For example, value of 0 means divide by 2 × 0 = 0 (No division, bypass) value of 1 means divide by 2 × 1 = 2 value of "ff" means divide by 2 × 255 = 510, and so on.	0
clk_divider2	[23:16]	RW	Clock divider-2 value. Clock division is 2 × n. For example, value of 0 means divide by 2 × 0 = 0 (No division, bypass) value of 1 means divide by 2 × 1 = 2 value of "ff" means divide by 2 × 255 = 510, and so on.	0
clk_divider1	[15:8]	RW	Clock divider-1 value. Clock division is 2 × n. For example, value of 0 means divide by 2 × 0 = 0 (No division, bypass) value of 1 means divide by 2 × 1 = 2 value of "ff" means divide by 2 × 255 = 510, and so on.	0
clk_divider0	[7:0]	RW	Clock divider-0 value. Clock division is 2 × n. For example, value of 0 means divide by 2 × 0 = 0 (No division, bypass) value of 1 means divide by 2 × 1 = 2 value of "ff" means divide by 2 × 255 = 510, and so on.	0

Clock divider 1, 2 and 3 are not implemented. So, user can use clock divider 0 only.

23.6.1.4 CLKSRC

- Base Address: 0x1255_0000
- Address = Base Address + 0x000C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
clk_source	[31:0]	RW	Clock divider source for up to 16 SD cards supported. Each card has two bits assigned to it. For example, bits[1:0] assigned for card-0, which maps and internally routes clock divider[3:0] output to cclk_out[15:0] pins, depending on bit value. 00 = Clock divider 0 01 = Clock divider 1 10 = Clock divider 2 11 = Clock divider 3	0

The host controller in Exynos 4412 SCP uses only one card. So, this register is fixed to 0x0.

23.6.1.5 CLKENA

- Base Address: 0x1255_0000
- Address = Base Address + 0x0010, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
cclk_low_power	[31:16]	RW	Low-power control for up to 16 SD card clocks and one MMC card clock supported. 0 = Non-low-power mode 1 = Low-power mode; stops clock when card in IDLE (Should be normally set to only MMC and SD memory cards. For SDIO cards, if interrupts must be detected, clock should not be stopped).	0
cclk_enable	[15:0]	RW	Clock-enable control for up to 16 SD card clocks and one MMC card clock supported. 0 = Disables Clock 1 = Enables Clock	0

The host controller in Exynos 4412 SCP uses only one card. So in this register, bit[16] and bit[0] can be used.

23.6.1.6 TMOUT

- Base Address: 0x1255_0000
- Address = Base Address + 0x0014, Reset Value = 0xFFFF_FF40

Name	Bit	Type	Description	Reset Value
data_timeout	[31:8]	RW	Value for card Data Read Timeout; same value also used for Data Starvation by Host timeout. Value is in number of card output clocks – cclk_out of selected card.	0xFFFFFF
response_ timeout	[7:0]	RW	Response timeout value. Value is in number of card output clocks – cclk_out.	0x40

23.6.1.7 CTYPE

- Base Address: 0x1255_0000
- Address = Base Address + 0x0018, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0
card_width	[16]	RW	Bit width of card: 0 = Non 8-bit mode 1 = 8-bit mode	0
RSVD	[15:1]	–	Reserved	0
card_width	[0]	RW	Bit width of card: 0 = 1-bit mode 1 = 4-bit mode	0

The listed examples use values for CTYPE[16]:

- If CTYPE[16] = 1, The card is in 8-bit mode. Note that CTYPE[0] value is ignored; it is recommended to keep this set to 0.
- If CTYPE[16] = 0, The card is in either 1-bit or 4-bit mode, depending upon value of CTYPE[0]; that is, if CTYPE[0] = 1 – 4-bit, CTYPE[0] = 0 – 1-bit.

23.6.1.8 BLKSIZE

- Base Address: 0x1255_0000
- Address = Base Address + 0x001C, Reset Value = 0x200

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	–
block_size	[15:0]	RW	Block size	0x200

23.6.1.9 BYTCNT

- Base Address: 0x1255_0000
- Address = Base Address + 0x0020, Reset Value = 0x200

Name	Bit	Type	Description	Reset Value
byte_count	[31:0]	RW	Number of bytes to be transferred; should be integer multiple of Block Size for block transfers. For undefined number of byte transfers, byte count should be set to 0. When byte count is set to 0, it is the responsibility of the host to explicitly send stop/abort command to terminate data transfer.	0x200

23.6.1.10 INTMASK

- Base Address: 0x1255_0000
- Address = Base Address + 0x0024, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
sdio_int_ mask	[31:16]	RW	Masks SDIO interrupts One bit for each card. Bit [31] corresponds to card[15], and bit[16] corresponds to card[0]. When masked, SDIO interrupt detection for that card is disabled. A 0 masks an interrupt, and 1 enables an interrupt.	0x0
int_mask	[15:0]	RW	Bits used to mask unwanted interrupts. A value of 0 masks interrupt; whereas a value of 1 enables interrupt bit. [15] = End-bit error (read)/Write no CRC (EBE) bit [14] = Auto command done (ACD) bit [13] = Start-bit error (SBE) bit [12] = Hardware locked write error (HLE) bit [11] = FIFO under run/overrun error (FRUN) bit [10] = Data starvation-by-host timeout (HTO)/Volt_switch_int bit [9] = Data read timeout (DRTO) bit [8] = Response timeout (RTO) bit [7] = Data CRC error (DCRC) bit [6] = Response CRC error (RCRC) bit [5] = Receive FIFO data request (RXDR) bit [4] = Transmit FIFO data request (TXDR) bit [3] = Data transfer over (DTO) bit [2] = Command done (CD) bit [1] = Response error (RE) bit	0x0

23.6.1.11 CMDARG

- Base Address: 0x1255_0000
- Address = Base Address + 0x0028, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
cmd_arg	31:0	RW	Value indicates command argument to be passed to card.	0

23.6.1.12 CMD

- Base Address: 0x1255_0000
- Address = Base Address + 0x002C, Reset Value = 0x2000_0000

Name	Bit	Type	Description	Reset Value
start_cmd	[31]	R	Start command. Once command is taken by CIU, bit is cleared. When bit is set, host should not attempt to write to any command registers. If write is attempted, hardware lock error is set in raw interrupt register. Once command is sent and response is received from SD_MMC_CEATA cards, it sets Command Done bit in raw interrupt register.	0
RSVD	[30]	–	Reserved	0
use_hold_reg	[29]	R	Use Hold Register 0 = CMD and DATA sent to card bypassing HOLD Register 1 = CMD and DATA sent to card through the HOLD Register NOTE: 1. Set to 1'b1 for SDR12 and SDR25 (with non-zero phase-shifted cclk_in_drv); zero phase shift is not allowed in these modes. 2. Set to 1'b0 for SDR50, SDR104, and DDR50 (with zero phase-shifted cclk_in_drv) 3. Set to 1'b1 for SDR50, SDR104, and DDR50 (with non-zero phase-shifted cclk_in_drv)	1
volt_switch	[28]	R	Voltage switch bit 0 = No voltage switching 1 = Enables voltage switching; should be set for CMD11 only	0
boot_mode	[27]	R	Boot Mode 0 = Mandatory Boot operation 1 = Alternate Boot operation	0
disable_boot	[26]	R	Disables Boot. When software sets this bit along with start_cmd, CIU terminates boot operation. Do NOT set disable_boot and enable_boot together.	0
expect_ boot_ack	[25]	R	Expect Boot Acknowledge. When Software sets this bit along with enable_boot, CIU expects a boot acknowledge start pattern of 0-1-0 from the selected card.	0
enable_boot	[24]	R	Enable Boot–this bit should be set only for mandatory boot	0

Name	Bit	Type	Description	Reset Value
			mode. When Software sets this bit along with start_cmd, CIU starts boot sequence for the corresponding card by asserting the CMD line low. Do NOT set disable_boot and enable_boot together.	
ccs_ expected	[23]	R	CCS expected 0 = Interrupts are not enabled in CE-ATA device (nIEN = 1 in ATA control register), or command does not expect CCS from device. 1 = Interrupts are enabled in CE-ATA device (nIEN = 0). RW_BLK command expects command completion signal from CE-ATA device. If the command expects Command Completion Signal (CCS) from the CE-ATA device, the software sets this control bit. DWC_mobile_storage sets Data Transfer Over (DTO) bit in RINTSTS register. It generates interrupt to host if Data Transfer Over interrupt is not masked.	0
read_ ceata_device	[22]	R	Read CEATA device 0 = Host is not performing read access (RW_REG or RW_BLK) towards CE-ATA device 1 = Host performs read access (RW_REG or RW_BLK) towards CE-ATA device Software should set this bit to indicate that it accesses CE-ATA device for read transfer. This bit is used to disable read data timeout indication while performing CE-ATA read transfers. Maximum value of I/O transmission delay can be no less than 10 seconds. DWC_mobile_storage should not indicate read data timeout while waiting for data from CE-ATA device.	0
update_ clock_ registers_ only	[21]	R	Update clock registers only 0 = Normal command sequence 1 = Do not send commands, just update clock register value into card clock domain Following register values transferred into card clock domain: CLKDIV, CLRSRC, CLKENA. Changes card clocks (change frequency, truncate off or on, and set low-frequency mode); provided in order to change clock frequency or stop clock without having to send command to cards. During normal command sequence, when update_clock_registers_only = 0, following control registers are transfers from BIU to CIU: CMD, CMDARG, TMOUT, CTYPE, BLKSIZ, BYTCNT. CIU uses new register values for new command sequence to card (s). When bit is set, there are no Command Done interrupts because it does not send any command to SD_MMC_CEATA cards.	0
card_number	[20:16]	R	Card number in use. Represents physical slot number of card being accessed. In MMC-Version 3.3-only mode, up to 30	0

Name	Bit	Type	Description	Reset Value
			cards are supported;. In SD-only mode, up to 16 cards are supported. Registered version of this is reflected on dw_dma_card_num and ge_dma_card_num ports, which can be used to create separate DMA requests, if required. Additionally, in SD mode this is used to mux or demux signals from selected card because each card is interfaced to DWC_mobile_storage by separate bus.	
send_ initialization	[15]	R	Send initialization 0 = Does not send initialization sequence (80 clocks of 1) before sending this command 1 = Sends initialization sequence before sending this command After power on, 80 clocks must be sent to card for initialization before sending any commands to card. Bit should be set while sending first command to card so that controller initializes clocks before sending command to card. This bit should not be set for either of the boot modes (Alternate or mandatory).	0
stop_ abort_cmd	[14]	R	Stop abort command 0 = Neither stops nor aborts command to stop current data transfer in progress. If abort is sent to function-number currently selected or not in data-transfer mode, then bit should be set to 0. 1 = Stops or aborts command that intends to stop current data transfer in progress. When open-ended or predefinedata transfer is in progress, and host issues stop or abort command to stop data transfer, bit should be set so that command/data state-machines of CIU can return correctly to idle state. This is also applicable for Boot mode transfers. To Abort boot mode, this bit should be set along with CMD[26] = disable_boot.	0
wait_prvdata _ complete	[13]	R	Wait previous data complete 0 = Sends command at once, even if previous data transfer does not complete 1 = Waits for previous data transfer completion before sending command The wait_prvdata_complete = 0. option typically used to query status of card during data transfer or to stop current data transfer. Card_number should be same as in previous command.	0
send_ auto_stop	[12]	R	Send Auto stop. 0 = No stop command sent at end of data transfer 1 = Sends stop command at end of data transfer When set, DWC_mobile_storage sends stop command to SD_MMC_CEATA cards at end of data transfer. When send_auto_stop bit should be set, since some data transfers do not need explicit stop commands Open-ended transfers that software should explicitly send to	0

Name	Bit	Type	Description	Reset Value
			stop command Additionally, when "resume" is sent to resume – suspended memory access of SD-Combo card – bit should be set correctly if suspended data transfer requires to send_auto_stop. Don't care if no data expected from card.	
transfer_ mode	[11]	R	Transfer mode 0 = Block data transfer command 1 = Stream data transfer command Don’t care if no data expected.	0
read/write	[10]	R	Read write 0 = Reads from card 1 = Writes to card Don’t care if no data expected from card.	0
data_ expected	[9]	R	Data expected 0 = No data transfer expected (Read/Write) 1 = Expects data transfer (Read/Write)	0
check_ response_ crc	[8]	R	Check response CRC 0 = Does not check response CRC 1 = Checks response CRC Some of command responses do not returns valid CRC bits. Software should disable CRC checks for those commands i to disable CRC checking by controller.	0
response_ length	[7]	R	Response length 0 = Expects short response from card 1 = Expects long response from card	0
response_ expect	[6]	R	Response expect 0 = No response expected from card 1 = Expects response from card	0
cmd_index	[5:0]	R	Command index	0

23.6.1.13 RESP0

- Base Address: 0x1255_0000
- Address = Base Address + 0x0030, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
response0	[31:0]	R	Bit[31:0] of response	0

23.6.1.14 RESP1

- Base Address: 0x1255_0000
- Address = Base Address + 0x0034, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
response1	[31:0]	R	Register represents bit [63:32] of long response. When CIU sends auto-stop command, then response is saved in register. It still preserves response for previous command sent by host in Response 0 register. Additional auto-stop issued only for data transfer commands, and response type is always "short" for them.	0

23.6.1.15 RESP2

- Base Address: 0x1255_0000
- Address = Base Address + 0x0038, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
response2	[31:0]	R	Bit[95:64] of long response	0

23.6.1.16 RESP3

- Base Address: 0x1255_0000
- Address = Base Address + 0x003C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
response3	[31:0]	R	Bit[127:96] of long response	0

23.6.1.17 MINTSTS

- Base Address: 0x1255_0000
- Address = Base Address + 0x0040, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
sdio_interrupt	[31:16]	R	Interrupt from SDIO card; one bit for each card. Bit[31] corresponds to Card[15], and bit[16] is for Card[0]. Enables SDIO interrupt for card only if corresponding sdio_int_mask bit is set in Interrupt mask register (Mask bit 1 enables interrupt; 0 masks interrupt). 0 = No SDIO interrupt from card 1 = SDIO interrupt from card	0
int_status	[15:0]	R	Enables interrupt only if corresponding bit in interrupt mask register is set. Bit[15]: End-bit error (Read/Write) no CRC (EBE) Bit[14]: Auto Command Done (ACD) Bit[13]: Start-bit Error (SBE) Bit[12]: Hardware Locked Write Error (HLE) Bit[11]: FIFO underrun/overrun error (FRUN) Bit[10]: Data Starvation byHhost Timeout (HTO)/Volt_switch_int Bit[9]: Data Read Timeout (DRTO) Bit[8]: Response Timeout (RTO) Bit[7]: Data CRC Error (DCRC) Bit[6]: Response CRC Error (RCRC) Bit[5]: Receive FIFO Data Request (RXDR) Bit[4]: Transmit FIFO Data Request (TXDR) Bit[3]: Data Transfer Over (DTO) Bit[2]: Command Done (CD) Bit[1]: Response Error (RE) Bit[0]: Card Detect (CD)	0

23.6.1.18 RINTSTS

- Base Address: 0x1255_0000
- Address = Base Address + 0x0044, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
sdio_interrupt	[31:16]	RW	Interrupt from SDIO card; one bit for each card. Bit[31] corresponds to Card[15], and bit[16] is for Card[0]. Write to these bits clears them. Value of 1 clears bit and 0 leaves bit intact. 0 = No SDIO interrupt from card 1 = SDIO interrupt from card	0
int_status	[15:0]	RW	A Write to these bits clears status bit. Value of 1 clears status bit, and value of 0 leaves bit intact. Bits are logged regardless of interrupt mask status. Bit[15]: End-bit Error (read)/write no CRC (EBE) Bit[14]: Auto Command Done (ACD) Bit[13]: Start-bit Error (SBE) Bit[12]: Hardware Locked Write Error (HLE) Bit[11]: FIFO underrun/overrun error (FRUN) Bit[10]: Data Starvation-by-host Timeout (HTO)/Volt_switch_int Bit[9]: Data Read Timeout (DRTO)/Boot Data Start (BDS) Bit[8]: Response Timeout (RTO)/Boot Ack Received (BAR) Bit[7]: Data CRC error (DCRC) Bit[6]: Response CRC error (RCRC) Bit[5]: Receive FIFO Data Request (RXDR) Bit[4]: Transmit FIFO Data Request (TXDR) Bit[3]: Data Transfer Over (DTO) Bit[2]: Command Done (CD) Bit[1]: Response Error (RE) Bit[0]: Card Detect (CD) 0000b TMCLK × 213	0

23.6.1.19 STATUS

- Base Address: 0x1255_0000
- Address = Base Address + 0x0048, Reset Value = {20'h00000, 4'b00xx, 8'h06}

Name	Bit	Type	Description	Reset Value
dma_req	[31]	R	DMA request signal state Either dw_dma_req or ge_dma_req, depending on DW-DMA or Generic-DMA selection.	0
dma_ack	[30]	R	DMA acknowledge signal state Either dw_dma_ack or ge_dma_ack, depending on DW-DMA or Generic-DMA selection.	0
fifo_count	[29:17]	R	FIFO count Number of filled locations in FIFO	0
response_ index	[16:11]	R	Index of previous response, including any auto-stop sent by core	0
data_state_ mc_bus y	[10]	R	Data transmit or receive state-machine is busy	0
data_busy	[9]	R	Inverted version of raw selected card_data[0] 0 = Card data is not busy 1 = Card data is busy	1 or 0; depends on dcata_in
data_3_status	[8]	R	Raw selected card_data[3]; checks whether card is present 0 = Card is not present 1 = Card is present	1 or 0; depends on dcata_in
command fsm states	[7:4]	R	Command FSM states: [15] = Wait; CMD-to-response turnaround [14] = Cmd path wait NCC [13] = Rx resp end bit [12] = Rx resp crc7 [11] = Rx resp data [10] = Rx resp cmd idx [9] = Rx resp tx bit [8] = Rx resp IRQ response [7] = Rx resp start bit [6] = Tx cmd end bit [5] = Tx cmd crc7 [4] = Tx cmd index + arg [3] = Tx cmd tx bit [2] = Tx cmd start bit [1] = Send init sequence [0] = Idle NOTE: The command FSM state is represented using 19 bits. The STATUS Register[7:4] has four bits to represent command FSM states. Using these four bits, only 16 states can be represented. Therefore three states cannot be represented in STATUS[7:4] register. The three states that	0

Name	Bit	Type	Description	Reset Value
			are not represented in the STATUS Register[7:4] are: • Bit[18]: Boot Mode • Bit[17]: Send CCSD • Bit[16]: Wait for CCS Due to this, while command FSM is in "Wait for CCS state" or "Send CCSD" or "Boot Mode", Status register indicates status as 0 for the bit field[7:4].	
fifo_full	[3]	R	FIFO is full status	0
fifo_empty	[2]	R	FIFO is empty status	1
fifo_tx_ watermark	[1]	R	FIFO reaches Transmit watermark level; Does not qualify with data transfer.	1
fifo_rx_ watermark	[0]	R	FIFO reaches Receive watermark level; Does not qualify with data transfer.	0

23.6.1.20 FIFOTH

- Base Address: 0x1255_0000
- Address = Base Address + 0x004C, Reset Value = {4'h0, 12'h7F, 16'h0}

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
DW_DMA_ Mutiple_ Transaction_ Size	[30:28]	RW	Burst size of multiple transactions. should be programmed similar as DW-DMA controller multiple-transaction-size SRC/DEST_MSIZE. 000 = 1 transfers 001 = 4 010 = 8 011 = 16 100 = 32 101 = 64 110 = 128 111 = 256 The unit for transfers is H_DATA_WIDTH parameter. A single transfer (dw_dma_single assertion in case of Non DW DMA interface) would be signalled based on this value. Value should be sub-multiple of (RX_WMark + 1) × (F_DATA_WIDTH/H_DATA_WIDTH) and (FIFO_DEPTH – TX_WMark) × (F_DATA_WIDTH/ H_DATA_WIDTH) For example, if FIFO_DEPTH = 16, FDATA_WIDTH == H_DATA_WIDTH Allowed combinations for MSize and TX_WMark are: MSize = 1, TX_WMARK = 1-15 MSize = 4, TX_WMark = 8 MSize = 4, TX_WMark = 4 MSize = 4, TX_WMark = 12 MSize = 8, TX_WMark = 8 MSize = 8, TX_WMark = 4 Allowed combinations for MSize and RX_WMark are: MSize = 1, RX_WMARK = 0-14 MSize = 4, RX_WMark = 3 MSize = 4, RX_WMark = 7 MSize = 4, RX_WMark = 11 MSize = 8, RX_WMark = 7 MSize = 8, RX_WMark = 11 Recommendation: MSize = 8, TX_WMark = 8, RX_WMark = 7	0
RX_WMark	[27:16]	RW	FIFO thresholds watermark level when it receives data to card. When FIFO data count reaches greater than this number, DMA/FIFO request is raised. During end of packet, request is generated regardless of threshold programming to complete any remaining data. In non-DMA mode, when it enables receiver FIFO threshold (RXDR) interrupt, then interrupt is generated instead of DMA request.	FIFO_DEPTH - 1 = h'7F

Name	Bit	Type	Description	Reset Value
			During end of packet, interrupt does not generate if threshold programming is larger than any remaining data. It is the responsibility of host to read remaining bytes on seeing Data Transfer Done interrupt. In DMA mode, at end of packet, even if remaining bytes are less than threshold, DMA request does single transfers. This is done to flush out any remaining bytes before Data Transfer Done interrupt is set. 12 bits = 1 bit less than FIFO-count of status register, which is 13 bits. Limitation: RX_WMark <= FIFO_DEPTH-2 Recommendation: (FIFO_DEPTH/2) – 1; (Means greater than (FIFO_DEPTH/2) – 1) NOTE: In DMA mode during CCS time-out, DMA does not generate request at the end of packet, even if remaining bytes are less than threshold. In this case, there will be some data left in FIFO. It is the responsibility of application to reset FIFO after CCS timeout.	
RSVD	[15:12]	–	Reserved	0
TX_WMark	[11:0]	RW	FIFO threshold watermark level when transmitting data to card. When FIFO data count is less than or equal to this number, DMA/FIFO request is raised. If Interrupt is enabled, then interrupt occurs. During end of packet, request or interrupt is generated, regardless of threshold programming. In non-DMA mode, when it enables transmit FIFO threshold (TXDR) interrupt, then it generates interrupt instead of DMA request. During end of packet, on last interrupt, host is responsible for filling FIFO with only required remaining bytes (not before FIFO is full or after CIU completes data transfers, because FIFO may not be empty). In DMA mode, at end of packet, if last transfer is less than burst size, DMA controller does single cycles until required bytes are transferred. 12 bits = 1 bit less than FIFO-count of status register, which is 13 bits. Limitation: TX_WMark >= 1; Recommendation: FIFO_DEPTH/2; (Means less than or equal to FIFO_DEPTH/2)	0

23.6.1.21 CDETECT

- Base Address: 0x1255_0000
- Address = Base Address + 0x0050, Reset Value = Value in card_detect signal

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	–
card_detect_n	[29:0]	RO	Value on card_detect_n input ports (1-bit per card); Read-only bits. 0 represents presence of card. Only NUM_CARDS number of bits are implemented.	card_detec_n inputs

23.6.1.22 WRTORT

- Base Address: 0x1255_0000
- Address = Base Address + 0x0054, Reset Value = Value in card_write_prt signal

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	–
write_protect	[29:0]	RO	Value on card_write_prt input ports (1-bit per card). 1 represents write protection. Only NUM_CARDS number of bits are implemented.	card_write_prt inputs

23.6.1.23 GPIO

- Base Address: 0x1255_0000
- Address = Base Address + 0x0058, Reset Value = {24'h0, gp_in signal}

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	–
gpo	[23:8]	RW	Value requires to be driven to gpo pins. This portion of register is Read/write. Valid only when AREA_OPTIMIZED parameter is 0.	0
gpi	[7:0]	RWX	Value on gpi input ports. This portion of register is Read-only. Valid only when AREA_OPTIMIZED parameter is 0.	gpi input

23.6.1.24 TCBCNT

- Base Address: 0x1255_0000
- Address = Base Address + 0x005C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
trans_card_byte_count	[31:0]	RO	Number of bytes transferred by CIU unit to card. In 32-bit or 64-bit AMBA data-bus-width modes, register should be accessed in full to avoid read-coherency problems. In 16-bit AMBA data-bus-width mode, it implements internal 16-bit coherency register. User should first read lower 16 bits and then higher 16 bits. When reading lower 16 bits, it stores higher 16 bits of counter in temporary register. When higher 16 bits are read, data from temporary register is supplied. TCBCNT and TBBCNT shares same coherency register. When AREA_OPTIMIZED parameter is 1, register should be Read only after data transfer completes. During data transfer, register returns 0.	0

23.6.1.25 TBBCNT

- Base Address: 0x1255_0000
- Address = Base Address + 0x0060, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
trans_fifo_byte_count	[31:0]	RO	Number of bytes transferred between Host/DMA memory and BIU FIFO. In 32-bit or 64-bit AMBA data-bus-width modes, register should be accessed in full to avoid read-coherency problems. In 16-bit AMBA data-bus-width mode, it implements internal 16-bit coherency register. You should first read lower 16 bits and then higher 16 bits. When reading lower 16 bits, it stores higher 16 bits of counter in temporary register. When higher 16 bits are read, data from temporary register is supplied. TCBCNT and TBBCNT shares same coherency register.	0

23.6.1.26 DEBNCE

- Base Address: 0x1255_0000
- Address = Base Address + 0x0064, Reset Value = 0x00FF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	–
debounce_ count	[23:0]	RW	Number of host clocks (clk) used by debounce filter logic. Typical debounce time is 5-25 ms.	0xFFFFFF

23.6.1.27 USRID

- Base Address: 0x1255_0000
- Address = Base Address + 0x0068, Reset Value = UID_REG configuration parameter

Name	Bit	Type	Description	Reset Value
USRID	[31:0]	RW	User identification register. value set by user. Default reset value can be picked by user while configuring core before synthesis. Can also be used as scratch pad register.	Configuration value

23.6.1.28 VERID

- Base Address: 0x1255_0000
- Address = Base Address + 0x006C, Reset Value = 0x5342_240A

Name	Bit	Type	Description	Reset Value
VERID	[31:0]	RO	Synopsys version identification register; register value is hard-wired. Can be read by firmware to support different versions of core.	0x5342_240A

23.6.1.29 HCON

- Base Address: 0x1255_0000
- Address = Base Address + 0x0070,
  Reset Value = Dependent on user-defined values in configuration parameters

Name	Bit	Type	Description	Reset Value
HCON	[31:0]	RO	Hardware configurations selected by user before synthesizing core. Register values can be used to develop configuration-independent software drivers. • bit[31:27]: Reserved (0) • bit[26]: AREA_OPTIMIZED 0 = No area optimization 1 = Area optimization • bit[25:24]: NUM_CLK_DIVIDER-1 • bit[23]: SET_CLK_FALSE_PATH 0 = No false path 1 = False path set • bit[22]: IMPLEMENT_HOLD_REG 0 = No hold register 1 = Hold register • bit[21]: FIFO_RAM_INSIDE 0 = Outside 1 = Inside • bit[20:18]: GE_DMA_DATA_WIDTH 000 = 16 bits 001 = 32 bits 010 = 64 bits Others = Reserved • bit[17:16]: DMA_INTERFACE 00 = None 01 = DW_DMA 10 = GENERIC_DMA 11 = Reserved • bit[15:10]: H_ADDR_WIDTH 0 to 7 – Reserved 8 = 9 bits 9 = 10 bits … 31 = 32 bits 32 to 63 = Reserved • bit[9:7]: H_DATA_WIDTH 000 = 16 bits 001 = 32 bits 010 = 64 bits Others = Reserved • bit[6]: H_BUS_TYPE 0 = APB 1 = AHB	Configuration Dependent

Name	Bit	Type	Description	Reset Value
			• bit[5:1]: NUM_CARDS – 1 • bit[0]: CARD_TYPE 0 = MMC_ONLY 1 = SD_MMC For FIFO_DEPTH parameter, power-on value of RX_WMark value of FIFO Threshold Watermark Register represents FIFO_DEPTH – 1.	

23.6.1.30 UHS_REG

- Base Address: 0x1255_0000
- Address = Base Address + 0x0074, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
DDR_REG	[31:16]	RW	DDR mode. Determines voltage fed to buffers by an external voltage regulator. 0 = Non-DDR mode 1 = DDR mode UHS_REG[16] should be set for card number 0, In case of DDR mode, CTYPE[16] should be set to 1. (16-bit mode)	0
VOLT_REG	[15:0]	RW	High Voltage mode. Determines voltage fed to buffers by an external voltage regulator. 0 = Buffers supplied with 3.3 V Vdd 1 = Buffers supplied with 1.8 V Vdd These bits function as output of host controller and are fed to an external voltage regulator. The voltage regulator must switch the voltage of buffers of a particular card to either 3.3 V or 1.8 V, depending on the value programmed in register. VOLT_REG[0] should be set to 1'b1 for card number 0 to make it operate for 1.8 V.	0

23.6.1.31 BMOD

- Base Address: 0x1255_0000
- Address = Base Address + 0x0080, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved	–
PBL	[10]	RW	Programmable Burst Length. These bits indicate maximum number of beats to be performed in one IDMAC transaction. The IDMAC will always attempt to burst as specified in PBL each time it starts a Burst transfer on host bus. The permissible values are 1, 4, 8, 16, 32, 64, 128, and 256. This value is mirror of MSIZE of FIFOTH register. To change this value, write the required value to FIFOTH register. This is an encode value as follows. 000 = 1 transfers 001 = 4 transfers 010 = 8 transfers 011 = 16 transfers 100 = 32 transfers 101 = 64 transfers 110 = 128 transfers 111 = 256 transfers Transfer unit is either 16, 32, or 64 bits, based on HDATA_WIDTH. PBL is a read-only value.	0
DE	[7]	RW	IDMAC Enable. When set, the IDMAC is enabled. DE is Read/Write.	0
DSL	[6:2]	RW	Descriptor Skip Length Specifies the number of HWord/Word/Dword (Depending on 16/32/64-bit bus) to skip between two unchained descriptors. This is applicable only for dual buffer structure. DSL is Read/Write.	0
FB	[1]	RW	Fixed Burst Controls whether AHB Master interface performs fixed burst transfers or not. When set, AHB will use only SINGLE, INCR4, INCR8, or INCR16 during start of normal burst transfers. When reset, AHB will use SINGLE and INCR burst transfer operations. FB is Read/Write.	0
SWR	[0]	RW	Software Reset. When set, DMA Controller resets all its internal registers. SWR is Read/Write. It is automatically cleared after 1 clock cycle.	0

23.6.1.32 PLDMND

- Base Address: 0x1255_0000
- Address = Base Address + 0x0084, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
PD	[31:0]	RWX	Poll Demand. If OWN bit of a descriptor is not set, FSM goes to Suspend state. The host needs to write any value into this register. It writes for IDMAC FSM to resume normal descriptor fetch operation. This is a Write only register. PD bit is Write-only.	0

23.6.1.33 DBADDR

- Base Address: 0x1255_0000
- Address = Base Address + 0x0088, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
SDL	[31:0]	RW	Start of Descriptor List. Contains base address of First Descriptor. The LSB bits[0/1/2:0] for 16/32/64-bit bus-width) are ignored and taken as all-zero by the IDMAC internally. Hence these LSB bits are Read-only.	0

23.6.1.34 IDSTS

- Base Address: 0x1255_0000
- Address = Base Address + 0x008C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	–
FSM	[16:13]	RW	DMAC FSM present state. 0 = DMA_IDLE 1 = DMA_SUSPEND 2 = DESC_RD 3 = DESC_CHK 4 = DMA_RD_REQ_WAIT 5 = DMA_WR_REQ_WAIT 6 = DMA_RD 7 = DMA_WR 8 = DESC_CLOSE This bit is Read-only.	0
EB	[12:10]	RW	Error Bits. Indicates the type of error that causes a Bus Error. Valid only with Fatal Bus Error bit (IDSTS[2]) set. This field does not generate an interrupt. 3'b001 = Host Abort received during transmission 3'b010 = Host Abort received during reception Others = Reserved EB is Read-only.	0
AIS	[9]	RW	Abnormal Interrupt Summary. Logical OR of the following: IDSTS[2] = Fatal Bus Interrupt IDSTS[4] = DU bit Interrupt IDSTS[5] = Card Error Summary Interrupt Only unmask bits affect this bit. This is a sticky bit. Clear it each time when corresponding bit that causes AIS to be set is cleared. Writing a 1 clears this bit.	0
NIS	[8]	RW	Normal Interrupt Summary. Logical OR of the following: IDSTS[0] = Transmits Interrupt IDSTS[1] = Receives Interrupt Only unmask bits affect this bit. This is a sticky bit. Cleared it each time a corresponding bit that causes NIS to be set is cleared. Writing a 1 clears this bit.	0
RSVD	[7:6]	–	Reserved	–
CES	[5]	RW	Card Error Summary. Indicates the status of transaction to/from card. It is also present in RINTSTS. Indicates the logical OR of the mentioned bits: EBE: End Bit Error RTO: Response Timeout/Boot Ack Timeout RCRC: Response CRC SBE: Start Bit Error DRTO: Data Read Timeout/BDS timeout	0

Name	Bit	Type	Description	Reset Value
			DCRC: Data CRC for Receive RE: Response Error Write of 1 clears this bit.	
DU	[4]	RW	Descriptor Unavailable Interrupt This bit is set when descriptor is unavailable due to OWN bit = 0 (DES0[31] =0). Write of 1 clears this bit.	0
RSVD	[3]	–	Reserved	0
FBE	[2]	RW	Fatal Bus Error Interrupt Indicates that a Bus Error occurred (IDSTS[12:10]). When this bit is set, DMA disables all its bus accesses. Write of 1 clears this bit.	0
RI	[1]	RW	Receive Interrupt Indicates completion of data reception for a descriptor. Write of 1 clears this bit.	0
TI	[0]	RW	Transmit Interrupt Indicates that it completes data transmission for a descriptor. Writ of 1 clears this bit.	0

23.6.1.35 IDINTEN

- Base Address: 0x1255_0000
- Address = Base Address + 0x0090, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	–
AI	[9]	RW	Abnormal Interrupt Summary Enable. When set, an abnormal interrupt is enabled. This bit enables the following bits: IDINTEN[2] = Fatal Bus Error Interrupt IDINTEN[4] = DU Interrupt IDINTEN[5] = Card Error Summary Interrupt	0
NI	[8]	RW	Normal Interrupt Summary Enable. When set, a normal interrupt is enabled. When reset, a normal interrupt is disabled. This bit enables the following bits: IDINTEN[0] = Transmit Interrupt IDINTEN[1] = Receive Interrupt	0
RSVD	[7:6]	–	Reserved	0
CES	[5]	RW	Card Error summary Interrupt Enable. When set, it enables Card Interrupt summary.	0
DU	[4]	RW	Descriptor Unavailable Interrupt. When set along with Abnormal Interrupt Summary Enable, the DU interrupt is enabled.	0
RSVD	[3]	–	Reserved	0
FBE	[2]	RW	Fatal Bus Error Enable. When set with Abnormal Interrupt Summary Enable, the Fatal Bus Error Interrupt is enabled. When reset, Fatal Bus Error Enable Interrupt is disabled.	0
RI	[1]	RW	Receive Interrupt Enable. When set with Normal Interrupt Summary Enable, Receive Interrupt is enabled. When reset, Receive Interrupt is disabled.	0
TI	[0]	RW	Transmit Interrupt Enable. When set with Normal Interrupt Summary Enable, Transmit Interrupt is enabled. When reset, Transmit Interrupt is disabled.	0

23.6.1.36 DSCADDR

- Base Address: 0x1255_0000
- Address = Base Address + 0x0094, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
HDA	[31:0]	RO	Host Descriptor Address Pointer. Clears on reset. Pointer updated by IDMAC during operation. This register points to the start address of the current descriptor read by the IDMAC.	0

23.6.1.37 BUFADDR

- Base Address: 0x1255_0000
- Address = Base Address + 0x0098, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
HBA	[31:0]	RO	Host Buffer Address Pointer. Clears on Reset. Pointer updated by IDMAC during operation. This register points to the current Data Buffer Address being accessed by the IDMAC.	0

23.6.1.38 CLKSEL

- Base Address: 0x1255_0000
- Address = Base Address + 0x009C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:19]	–	Reserved	0
SelClk_drv	[18:16]	RW	Select drv clock among 8 shifted clocks: 000 = No phase shifted clock 001 = 1st phase shifted clock 010 = 2nd phase shifted clock 011 = 3rd phase shifted clock 100 = 4th phase shifted clock 101 = 5th phase shifted clock 110 = 6th phase shifted clock 111 = 7th phase shifted clock	0
RSVD	[15:3]	–	Reserved	0
SelClk_sample	[2:0]	RW	Select sample clock among 8 shifted clocks: 000 = no phase shifted clock 001 = 1st phase shifted clock 010 = 2nd phase shifted clock 011 = 3rd phase shifted clock 100 = 4th phase shifted clock 101 = 5th phase shifted clock 110 = 6th phase shifted clock	0

			111 = 7th phase shifted clock	

24 Pulse Width Modulation Timer

24.1 Overview

Exynos 4412 SCP has five 32-bit Pulse Width Modulation (PWM) timers. These timers generate internal interrupts for the ARM subsystem. Additionally, timers 0, 1, 2, and 3 include a PWM function that drives an external I/O signal. The PWM in timer 0 has an optional dead-zone generator capability to support a large current device. Timer 4 is an internal timer without output pins.

The Timers use the APB-PCLK as source clock. Timers 0 and 1 share a programmable 8-bit prescaler that provides the first level of division for the PCLK. Timers 2, 3, and 4 share a different 8-bit prescaler. Each timer has its own private clock-divider that provides a second level of clock division (prescaler divided by 2, 4, 8, or 16).

Each timer has its 32-bit down-counter; the timer clock drives this counter. The Timer Count Buffer registers (TCNTBn) loads initial value of the down-counter. If the down-counter reaches zero, it generates the timer interrupt request to inform the CPU that the timer operation is complete. If the timer down-counter reaches zero, the value of corresponding TCNTBn automatically reloads into the down-counter to start a next cycle. However, if the timer stops, for example, by clearing the timer enable bit of TCONn during the timer running mode, the value of TCNTBn does not reload into the counter.

The PWM function uses the value of the TCMPBn register. The timer control logic changes the output level if down-counter value matches the value of the compare register in timer control logic. Therefore, the compare register determines the turn-on time or turn-off time of a PWM output.

Each timer is double-buffer structure with the TCNTBn and TCMPBn registers to allow the timer parameters to update in the middle of a cycle. The new values do not take effect until the current timer cycle completes.

Figure 24-1 illustrates the simple example of a PWM cycle.

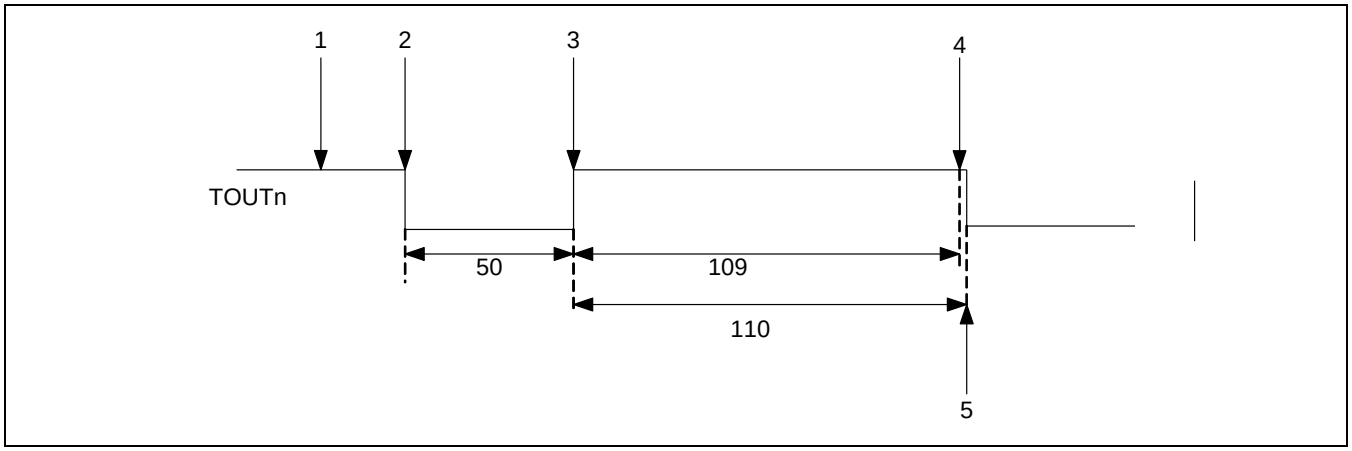


Figure 24-1 Simple Example of a PWM Cycle

Steps to use PWM as a pulse generator are:

5. Initialize the TCNTBn register with 159 (50 + 109) and TCMPBn with 109.
6. Start Timer: Set the start bit and manually update this bit to off.
7. The TCNTBn value of 159 is loaded into the down-counter, and then the output TOUTn is set to low.
8. If down-counter counts down the value from TCNTBn to value in the TCMPBn register 109, the output changes from low to high.
9. If the down-counter reaches 0, then it generates an interrupt request.
10. The down-counter automatically reloads TCNTBn. This restarts the cycle.

Figure 24-2 illustrates the clock generation scheme for individual PWM channels.

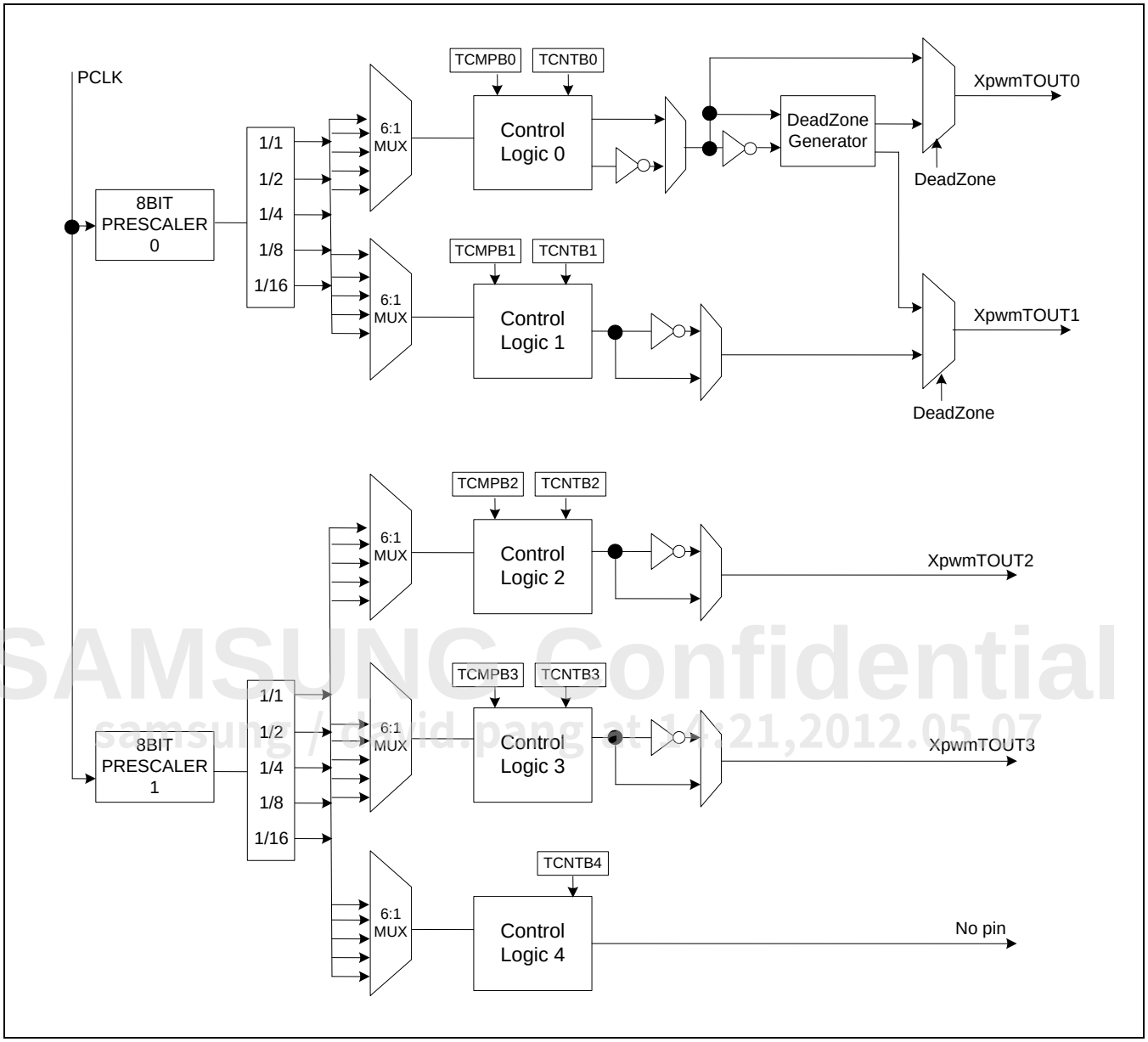


Figure 24-2 PWM TIMER Clock Tree Diagram

Each timer can generate level interrupts.

24.2 Features of PWM Timer

The features of PWM are:

- Five 32-bit timers.
- Two 8-bit Clock Prescalers providing first level of division for the PCLK. Five Clock Dividers and Multiplexers providing second level of division for the Prescaler clock.
- Programmable Clock Select Logic for individual PWM Channels
- Four Independent PWM Channels with Programmable Duty Control and Polarity
- Static Configuration: It stops PWM.
- Dynamic Configuration: PWM is running
- Auto-Reload and One-Shot Pulse Mode
- Dead Zone Generator on two PWM Outputs
- Level Interrupt Generation

The PWM has two operation modes. They are:

- Auto-Reload
- One-Shot Pulse:
- Auto-Reload Mode:
  In this mode, continuous PWM pulses are generated based on programmed duty cycle and polarity.
- One-Shot Pulse Mode:
  In this mode, only one PWM pulse is generated based on programmed duty cycle and polarity.

To control the functionality of PWM, 18 special function registers are provided. The PWM is an AMBA slave module which has programmable outputs and a clock input and the PWM connects to the Advanced Peripheral Bus (APB). These 18 special function registers within PWM are accessed via APB transactions.

24.3 PWM Operation

PWM timer of Exynos 4412 SCP can operate as a general timer and a pulse generator with TOUT signal.

24.3.1 Prescaler and Divider

An 8-bit prescaler and 3-bit divider generates these output frequencies:

Table 24-1 describes the minimum and maximum resolution based on prescaler and clock divider values.

Table 24-1 Minimum and Maximum Resolution Based on Prescaler and Clock Divider Values

4-bit Divider Settings	Minimum Resolution (Prescaler Value = 1)	Maximum Resolution (Prescaler Value = 255)	Maximum Interval (TCNTBn = 4294967295)
1/1 (PCLK = 66 MHz)	0.030us (33.0 MHz)	3.879us (257.8 kHz)	16659.27s
1/2 (PCLK = 66 MHz)	0.061us (16.5 MHz)	7.758us (128.9 kHz)	33318.53s
1/4 (PCLK = 66 MHz)	0.121us (8.25 MHz)	15.515us (64.5 kHz)	66637.07s
1/8 (PCLK = 66 MHz)	0.242us (4.13 MHz)	31.03us (32.2 kHz)	133274.14s
1/16 PCLK = 66 MHz)	0.485us (2.06 MHz)	62.061us (16.1 kHz)	266548.27s

24.3.2 Basic Timer Operation

Figure 24-3 illustrates the timer operations.

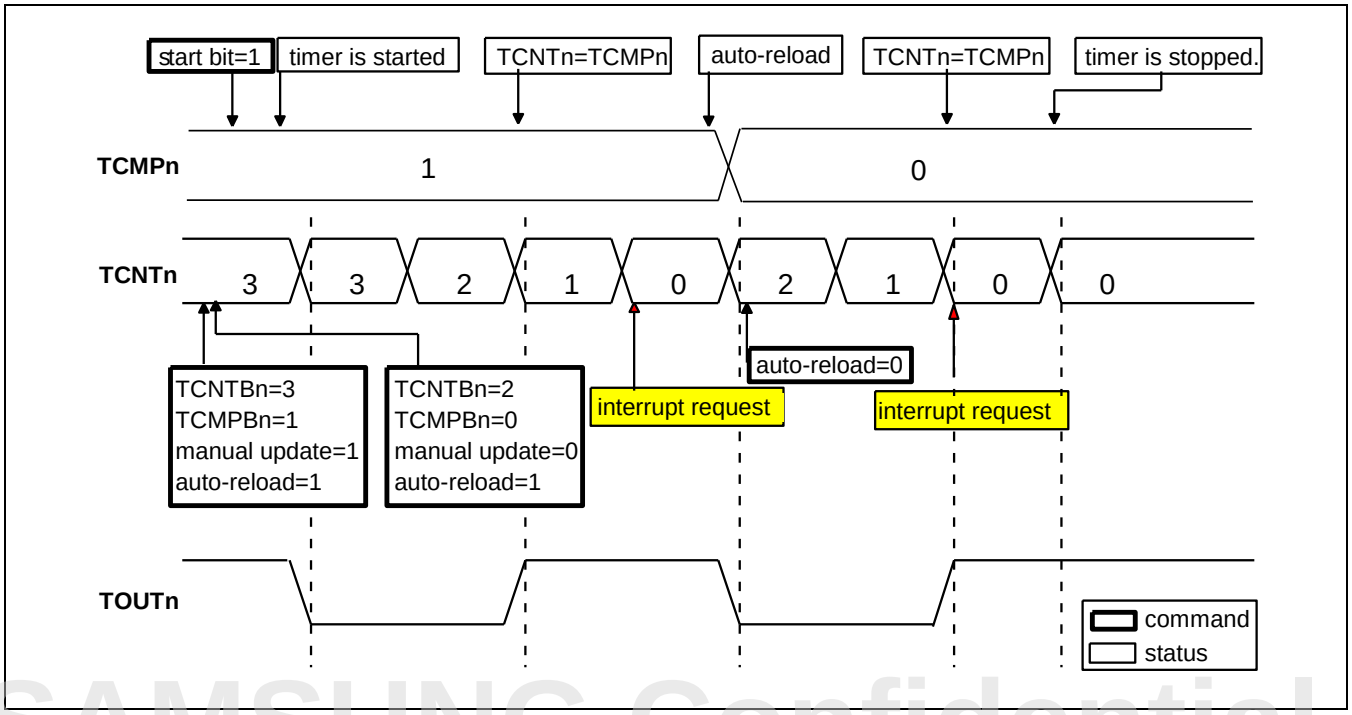


Figure 24-3 Timer Operations

The timer (except the timer channel 4) includes four registers. They are:

- TCNTBn
- TCNTn
- TCMPBn
- TCMPn

If the timer reaches 0, then TCNTBn and TCMPBn registers are loaded into TCNTn and TCMPn. If TCNTn reaches 0, then the interrupt request occurs if it enables the interrupt (TCNTn and TCMPn are the names of the internal registers. It reads the TCNTn register from the TCNTOn register).

To generate interrupt at intervals 3cycle of XpwmTOUTn, set TCNTBn, TCMPBn and TCON register as shown in *Figure 24-3*.

Steps to generate interrupt:

1. Set TCNTBn = 3 and TCMPBn = 1.
2. Set auto-reload = 1 and manual update = 1.
   If manual update bit is 1, then it loads TCNTBn and TCMPBn values to TCNTn and TCMPn.
3. Set TCNTBn = 2 and TCMPBn = 0 for the next operation.
4. Set auto-reload = 1 and manual update = 0.
   If you set manual update = 1 at this time, it changes TCNTn to 2 and it changes TCMP to 0.
   Therefore, it generates interrupt at interval two-cycle instead of three-cycle.
   You should set auto-reload = 1 automatically for the next operation.
5. Set start = 1 for starting the operation. Then TCNTn is down counting.
   If TCNTn is 0, it generates interrupt and if auto-reload is enable, it loads TCNTn 2 (TCNTBn value) and it loads TCMPn 0 (TCMPn value).
6. TCNTn is down counting before it stops.

24.3.3 Auto-reload and Double Buffering

PWM Timers includes a double buffering feature, which changes the reload value for the next timer operation without stopping the current timer operation.

The timer value is written into TCNTBn (Timer Count Buffer register) and the current counter value of the timer is read from TCNTOn (Timer Count Observation register). If TCNTBn is read, the read value does not reflect the current state of the counter but the reload value for the next timer duration.

Auto-reload is a copy function that a value of the TCNTBn is copied to the TCNTn when the TCNTn reaches 0. The value written to TCNTBn, is loaded to TCNTn if the TCNTn reaches to 0 and auto-reload is enabled. If the TCNTn is 0 and the auto-reload bit is 0, then TCNTn does not operate further.

Figure 24-4 illustrates the example of double buffering feature.

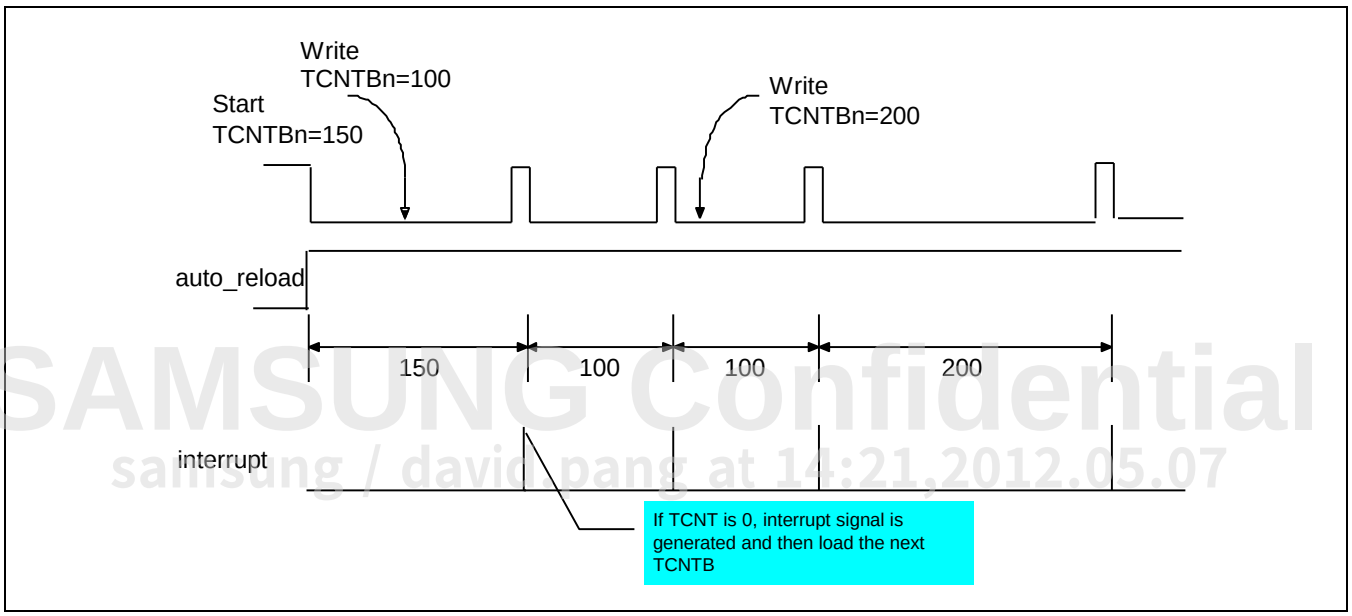


Figure 24-4 Example of Double Buffering Feature

24.3.4 Timer Operation Example

Figure 24-5 illustrates the example of a timer operation.

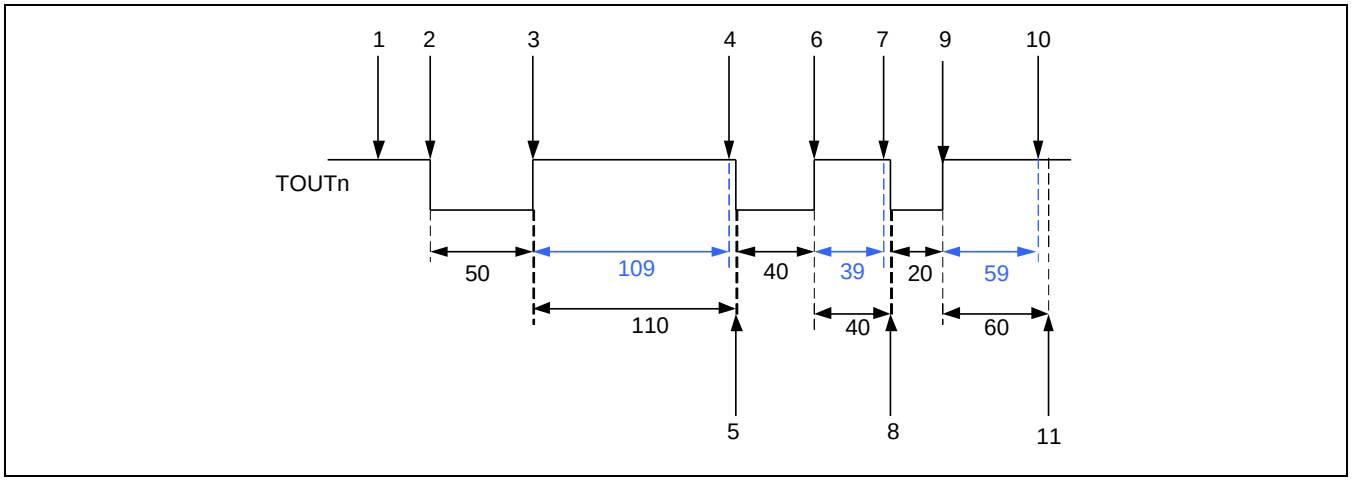


Figure 24-5 Example of a Timer Operation

Steps to use PWM as a timer:

1. 1Enable the auto-reload feature.
2. Set the TCNTBn as 159 (50 + 109) and TCMPBn as 109.
3. Set the manual update bit On and set the manual update bit Off.
4. Set the inverter On/Off bit. The manual update bit sets TCNTn and TCMPn to the value of TCNTBn and TCMPBn.
5. Set TCNTBn and TCMPBn as 79 (40 + 39) and 39.
6. Start Timer: Set the start bit in TCON.
7. If TCNTn and TCMPn have the same value, then it changes the logic level of TOUTn from low to high
8. When TCNTn reaches 0, it generates interrupt request.
9. It automatically reloads TCNTn and TCMPn with TCNTBn and TCMPBn as (79 (40 + 39)) and 39. In the Interrupt Service Routine (ISR), the TCNTBn and TCMPBn are set as 79 (20 + 59) and 59.
10. If TCNTn and TCMPn have the same value, then it changes the logic level of TOUTn from low to high
11. When TCNTn reaches to 0, it generates interrupt request.
12. It automatically reloads TCNTn and TCMPn with TCNTBn, TCMPBn as (79 (20 + 59)) and 59. It disables the auto-reload and interrupt request to stop the timer in the ISR.
13. If TCNTn and TCMPn have similar value, then it changes the logic level of TOUTn from low to high.
14. Even if TCNTn reaches to 0, it does not generate interrupt request.
15. Because auto-reload is disabled, it does not reload TCNTn and stop the timer.

24.3.5 Initialize Timer (Setting Manual-Up Data and Inverter)

You should define the starting value of the TCNTn, because an auto-reload operation of the timer occurs when the down counter reaches to "0" In this case, the starting value should be loaded by setting "1" to the manual update bit of TCON register.

1. Write the initial value into TCNTBn and TCMPBn.
2. Set the manual update bit and clear only manual update bit of the corresponding timer.

NOTE: We recommend you to set the inverter On/Off bit (whether inverter is used or not).

3. Set the start bit of the corresponding timer to start the timer.

24.3.6 PWM

Figure 24-6 illustrates the example of PWM.

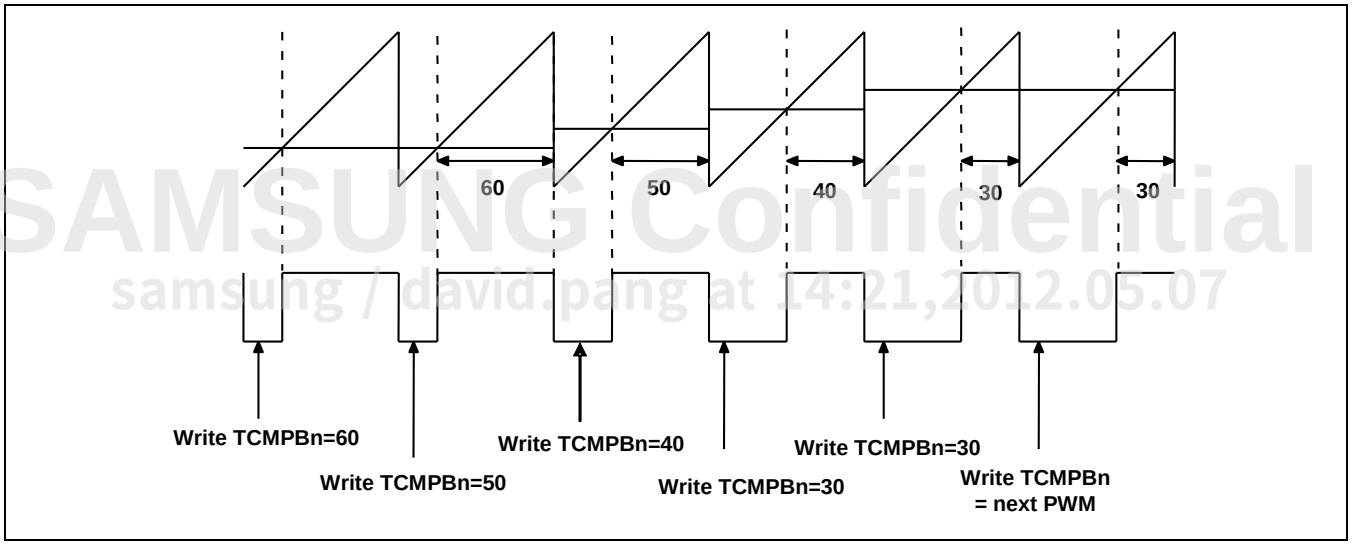


Figure 24-6 Example of PWM

Use TCMPBn to implement the PWM feature. TCNTBn determines PWM frequency. A As illustrated in *Figure 24-6* TCMPBn determines a PWM value.

For a higher PWM value, decrease the TCMPBn value. For a lower PWM value, increase the TCMPBn value. If you enable the output inverter, the increment/decrement can be disabled.

Due to the double buffering feature, you should write a counter value for next PWM cycle into the TCMPBn register.

24.3.7 During Current ISR. (Interrupt Service Routine) Output Level Control

Figure 24-7 illustrates the inverter On/Off.

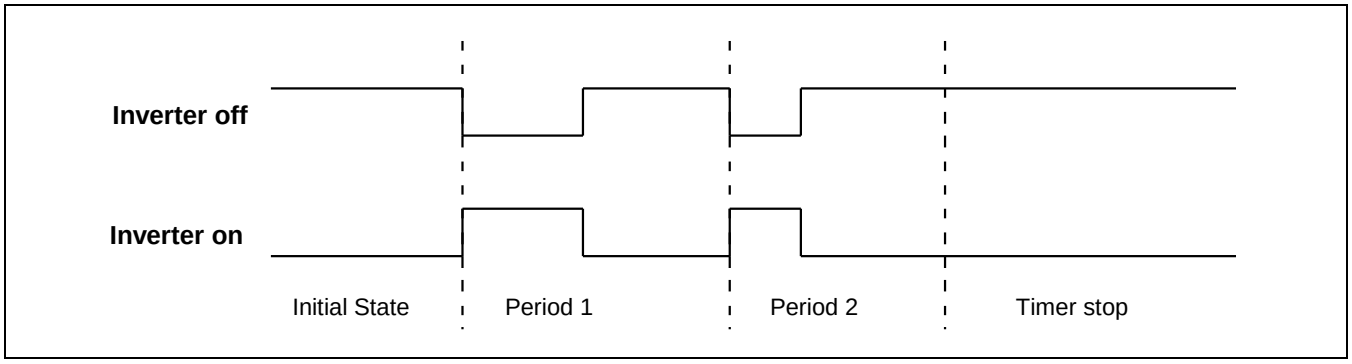


Figure 24-7 Inverter On/Off

Steps to maintain TOUT as high or low when inverter is turned Off:

1. Turn-Off the auto-reload bit. Then, TOUTn goes to high level and it stops the timer after TCNTn reaches to 0. This method is recommended.
2. Stop the timer by clearing the timer start/stop bit to 0. If TCNTn <= TCMPn, the output level is high. If TCNTn > TCMPn, the output level is low.
3. You can invert TOUTn signal by setting "1" to Inverter On/Off bit of TCON register. The inverter removes the additional circuit to adjust the output level.

24.3.8 Dead Zone Generator

Dead Zone Generator feature inserts the time gap between a turn-off and turn-on of two different switching devices. This time gap prohibits the two switching devices turning On simultaneously even for a very short duration.

TOUT_0 specifies the PWM output. nTOUT_0 specifies the inversion of the TOUT_0. If you enable the dead-zone, the output wave-form of TOUT_0 and nTOUT_0 become TOUT_0_DZ and nTOUT_0_DZ. Dead-zone interval cannot turn on TOUT0_DZ and nTOUT_0_DZ simultaneously. For functional accuracy, it should set the dead zone length smaller than compare counter value.

Figure 24-8 illustrates the waveform when it enables Dead Zone feature.

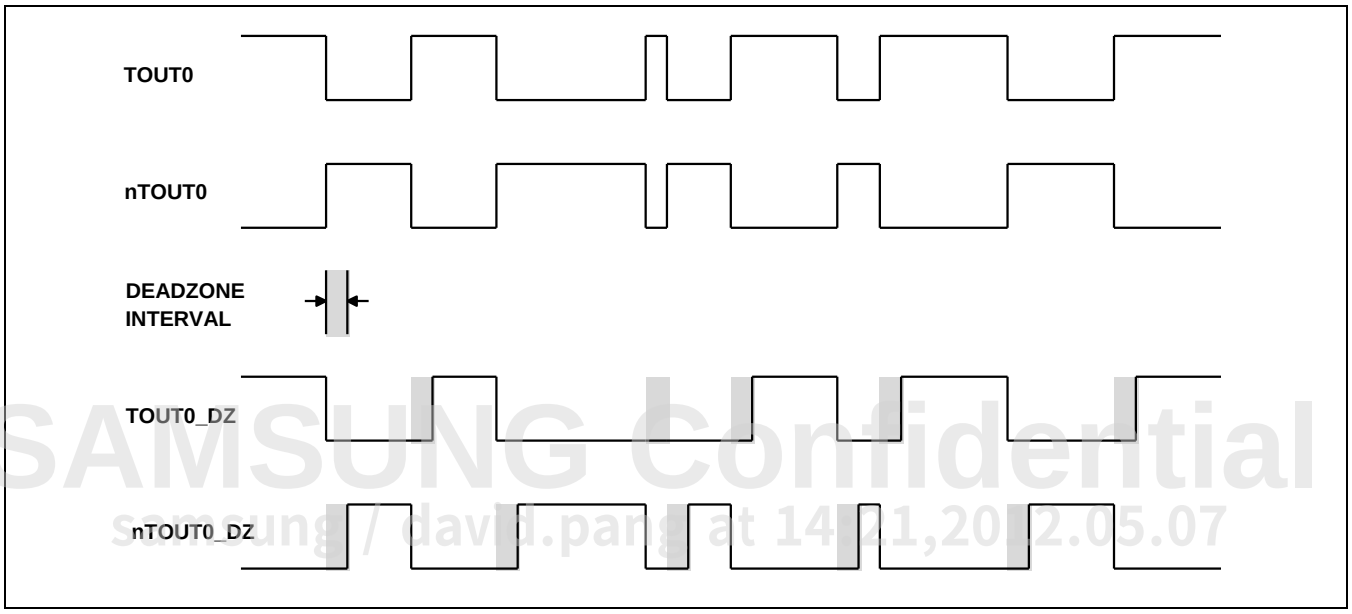


Figure 24-8 Waveform when a Dead Zone Feature is Enabled

24.4 I/O Description

Signal	I/O	Description	Pad	Type
TOUT_0	Output	PWMTIMER TOUT[0]	XpwmTOUT[0]	muxed
TOUT_1	Output	PWMTIMER TOUT[1]	XpwmTOUT[1]	muxed
TOUT_2	Output	PWMTIMER TOUT[2]	XpwmTOUT[2]	muxed
TOUT_3	Output	PWMTIMER TOUT[3]	XpwmTOUT[3]	muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

24.5 Register Description

24.5.1 Register Map Summary

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)

Register	Offset	Description	Reset Value
TCFG0	0x0000	Specifies the timer configuration register 0 that configures the two 8-bit prescaler and dead-zone or dead zone length	0x0000_0101
TCFG1	0x0004	Specifies the timer configuration register 1 that controls five MUX select bit	0x0000_0000
TCON	0x0008	Specifies the timer control register	0x0000_0000
TCNTB0	0x000C	Specifies the timer 0 count buffer register	0x0000_0000
TCMPB0	0x0010	Specifies the timer 0 compare buffer register	0x0000_0000
TCNTO0	0x0014	Specifies the timer 0 count observation register	0x0000_0000
TCNTB1	0x0018	Specifies the timer 1 count buffer register	0x0000_0000
TCMPB1	0x001C	Specifies the timer 1 compare buffer register	0x0000_0000
TCNTO1	0x0020	Specifies the timer 1 count observation register	0x0000_0000
TCNTB2	0x0024	Specifies the timer 2 count buffer register	0x0000_0000
TCMPB2	0x0028	Specifies the timer 2 compare buffer register	0x0000_0000
TCNTO2	0x002C	Specifies the timer 2 count observation register	0x0000_0000
TCNTB3	0x0030	Specifies the timer 3 count buffer register	0x0000_0000
TCMPB3	0x0034	Specifies the timer 3 compare buffer register	0x0000_0000
TCNTO3	0x0038	Specifies the timer 3 count observation register	0x0000_0000
TCNTB4	0x003C	Specifies the timer 4 count buffer register	0x0000_0000
TCNTO4	0x0040	Specifies the timer 4 count observation register	0x0000_0000
TINT_CSTAT	0x0044	Specifies the timer interrupt control and status register	0x0000_0000

24.5.1.1 TCFG0

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0000, Reset Value = 0x0000_0101

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved Bits	0x00
Dead zone length	[23:16]	RW	Dead zone length	0x00
Prescaler 1	[15:8]	RW	Prescaler 1 value for Timer 2, 3, and 4	0x01
Prescaler 0	[7:0]	RW	Prescaler 0 value for timer 0 and 1	0x01

Timer Input Clock Frequency = PCLK/({prescaler value + 1})/{divider value}
{prescaler value} = 1 to 255
{divider value} = 1, 2, 4, 8, 16
Dead zone length = 0 to 254

NOTE: If deadzone length is set as "n", real Dead Zone length is "n + 1" (n = 0 to 254).

24.5.1.2 TCFG1

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	0x000
Divider MUX4	[19:16]	RW	Selects Mux input for PWM timer 4 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16	0x0
Divider MUX3	[15:12]	RW	Selects Mux input for PWM timer 3 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16	0x0
Divider MUX2	[11:8]	RW	Selects Mux input for PWM timer 2 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16	0x0
Divider MUX1	[7:4]	RW	Selects Mux input for PWM timer 1 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16	0x0
Divider MUX0	[3:0]	RW	Selects Mux input for PWM timer 0 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16	0x0

24.5.1.3 TCON

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:23]	–	Reserved Bits	0x000
Timer 4 auto reload on/off	[22]	RW	0 = One-shot 1 = Interval mode (auto-reload)	0x0
Timer 4 manual update	[21]	RW	0 = No operation 1 = Updates TCNTB4	0x0
Timer 4 start/stop	[20]	RW	0 = Stops Timer 4 1 = Starts Timer 4	0x0
Timer 3 auto reload on/off	[19]	RW	0 = One-shot 1 = Interval mode(auto-reload)	0x0
Timer 3 output inverter on/off	[18]	RW	0 = Inverter Off 1 = TOUT_3 inverter-on	0x0
Timer 3 manual update	[17]	RW	0 = No operation 1 = Updates TCNTB3	0x0
Timer 3 start/stop	[16]	RW	0 = Stops Timer 3 1 = Starts Timer 3	0x0
Timer 2 auto reload on/off	[15]	RW	0 = One-shot 1 = Interval mode (auto-reload)	0x0
Timer 2 output inverter on/off	[14]	RW	0 = Inverter Off 1 = TOUT_2 inverter-on	0x0
Timer 2 manual update	[13]	RW	0 = No operation 1 = Updates TCNTB2,TCMPB2	0x0
Timer 2 start/stop	[12]	RW	0 = Stops Timer 2 1 = Starts Timer 2	0x0
Timer 1 auto reload on/off	[11]	RW	0 = One-shot 1 = Interval mode (auto-reload)	0x0
Timer 1 output inverter on/off	[10]	RW	0 = Inverter Off 1 = TOUT_1 inverter-on	0x0
Timer 1 manual update	[9]	RW	0 = No operation 1 = Updates TCNTB1 andTCMPB1	0x0
Timer 1 start/stop	[8]	RW	0 = Stops Timer 1 1 = Starts Timer 1	0x0
Reserved	[7:5]	–	Reserved Bits	0x0
Dead zone enable/disable	[4]	RW	Enables/Disables Dead zone generator	0x0
Timer 0 auto reload on/off	[3]	RW	0 = One-shot 1 = Interval mode (auto-reload)	0x0
Timer 0 output	[2]	RW	0 = Inverter Off	0x0

Name	Bit	Type	Description	Reset Value
inverter on/off			1 = TOUT_0 inverter-on	
Timer 0 manual update	[1]	RW	0 = No operation 1 = Updates TCNTB0 andTCMPB0	0x0
Timer 0 start/stop	[0]	RW	0 = Stops Timer 0 1 = Starts Timer 0	0x0

24.5.1.4 TCNTB0

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 0 count buffer	[31:0]	RW	Timer 0 Count Buffer register	0x0000_0000

24.5.1.5 TCMPB0

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 0 compare buffer	[31:0]	RW	Timer 0 Compare Buffer register	0x0000_0000

24.5.1.6 TCNTO0

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 0 count observation	[31:0]	R	Timer 0 Count Observation register	0x0000_0000

24.5.1.7 TCNTB1

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 1 count buffer	[31:0]	RW	Timer 1 Count Buffer register	0x0000_0000

24.5.1.8 TCMPB1

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 1 compare buffer	[31:0]	RW	Timer 1 Compare Buffer register	0x0000_0000

24.5.1.9 TCNTO1

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 1 count observation	[31:0]	R	Timer 1 Count Observation register	0x0000_0000

24.5.1.10 TCNTB2

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 2 count buffer	[31:0]	RW	Timer 2 Count Buffer register	0x0000_0000

24.5.1.11 TCMPB2

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 2 compare buffer	[31:0]	RW	Timer 2 Compare Buffer register	0x0000_0000

24.5.1.12 TCNTO2

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 2 count observation	[31:0]	R	Timer 2 Count Observation register	0x0000_0000

24.5.1.13 TCNTB3

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 3 count buffer	[31:0]	RW	Timer 3 Count Buffer register	0x0000_0000

24.5.1.14 TCMPB3

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 3 compare buffer	[31:0]	RW	Timer 3 Compare Buffer register	0x0000_0000

24.5.1.15 TCNTO3

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0038, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 3 count observation	[31:0]	R	Timer 3 Count Observation register	0x0000_0000

24.5.1.16 TCNTB4

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x003C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 4 count buffer	[31:0]	RW	Timer 4 Count Buffer register	0x0000_0000

24.5.1.17 TCNTO4

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Timer 4 count observation	[31:0]	R	Timer 4 Count Observation register	0x0000_0000

24.5.1.18 TINT_CSTAT

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved Bits	0x00000
Timer 4 interrupt status	[9]	RW	Timer 4 interrupt status bit. It clears by writing "1" on this bit.	0x0
Timer 3 interrupt status	[8]	RW	Timer 3 interrupt status bit. It clears by writing "1" on this bit.	0x0
Timer 2 interrupt status	[7]	RW	Timer 2 interrupt status bit. It clears by writing "1" on this bit.	0x0
Timer 1 interrupt status	[6]	RW	Timer 1 interrupt status bit. It clears by writing "1" on this bit.	0x0
Timer 0 interrupt status	[5]	RW	Timer 0 interrupt status bit. It clears by writing "1" on this bit.	0x0
Timer 4 interrupt enable	[4]	RW	Enables timer 4 interrupt 0 = Disables Timer 4 interrupt 1 = Enables Timer 4 interrupt	0x0
Timer 3 interrupt enable	[3]	RW	Enables timer 3 interrupt 0 = Disables Timer 3 interrupt 1 = Enables Timer 3 interrupt	0x0
Timer 2 interrupt enable	[2]	RW	Enables timer 2 interrupt 0 = Disables Timer 2 interrupt 1 = Enables Timer 2 interrupt	0x0
Timer 1 interrupt enable	[1]	RW	Enables timer 1 interrupt 0 = Disables Timer 1 interrupt 1 = Enables Timer 1 interrupt	0x0
Timer 0 interrupt enable	[0]	RW	Enables timer 0 interrupt. 0 = Disables Timer 0 interrupt 1 = Enables Timer 0 interrupt	0x0

25 Multi Core Timer (MCT)

25.1 Overview

Multi Core Timer (MCT) provides two features. They are:

- It provides time tick (1 ms) at any power mode except sleep mode.
- One global timer and four local timers for Multi core CPU.

Figure 25-1 illustrates the overall multi core timer block diagram.

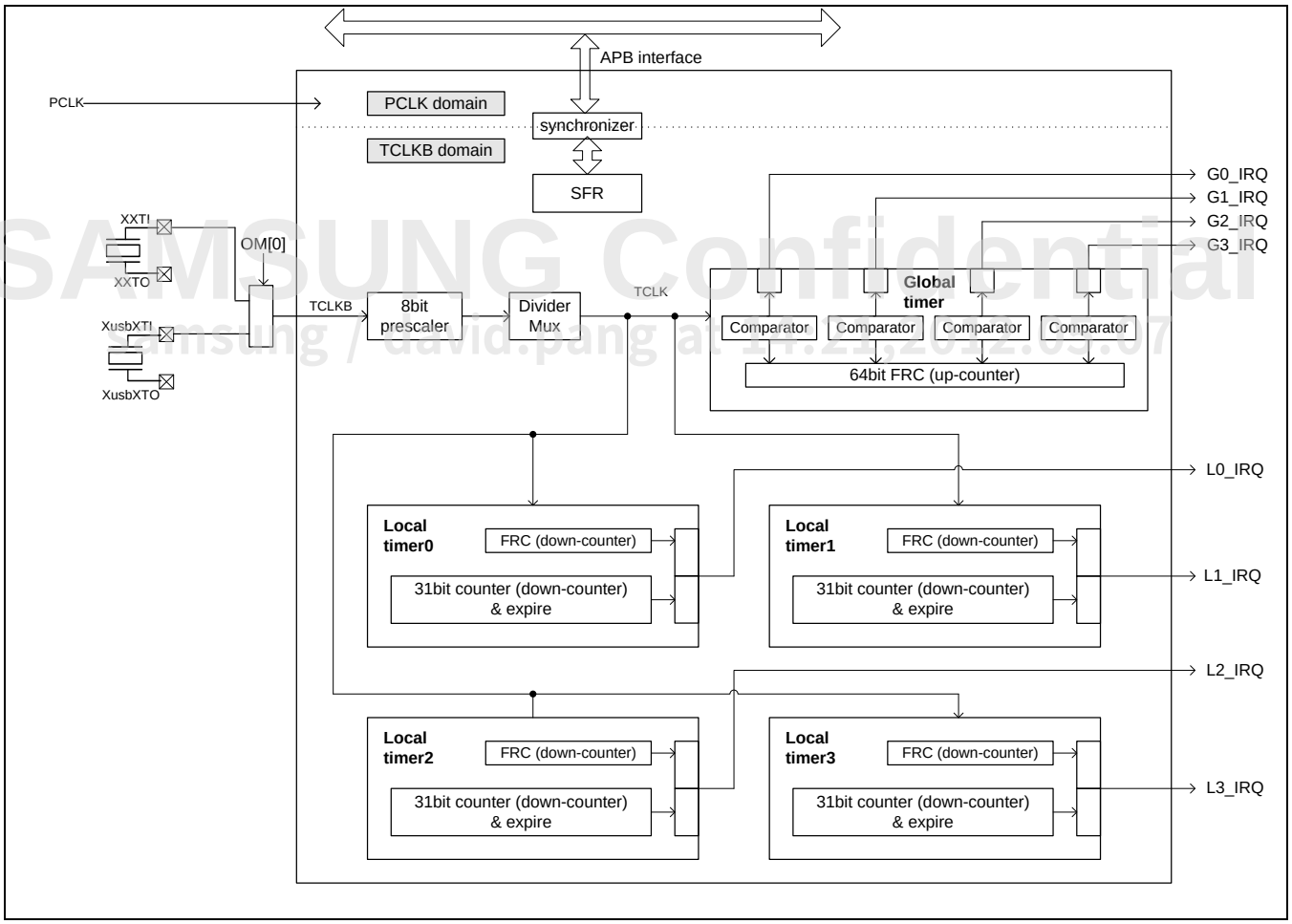


Figure 25-1 Overall Multi Core Timer Block Diagram

25.2 Features of MCT

Features of MCT are:

- The clock source of MCT is main OSC (XXTI) or USB OSC (XusbXTI).
- Counter bit: 64-bit for global timer and 31-bit for local timer
- Global timer: One Free Running Counter (FRC) (FRC, up-counter) and four comparators
- Local timer: Adjustable interrupt-interval without stopping reference tick and FRC (down-counter)
- Power mode: Used in all power modes except sleep mode

25.3 Internal Function of MCT

MCT includes one multi core timer and two local timers. These timers operate independently and generate their interrupts.

Table 25-1 describes the interrupt in MCT.

Table 25-1 Interrupt in MCT

IRQ signal	Interrupt source
G0_IRQ	Asserted, when FRC in global timer = comparator 0 in global timer
G1_IRQ	Asserted, when FRC in global timer = comparator 1 in global timer
G2_IRQ	Asserted, when FRC in global timer = comparator 2 in global timer
G3_IRQ	Asserted, when FRC in global timer = comparator 3 in global timer
L0_IRQ	Asserted, when FRC or 31bit counter in Local timer0 is expire.
L1_IRQ	Asserted, when FRC or 31bit counter in Local timer1 is expire.
L2_IRQ	Asserted, when FRC or 31bit counter in Local timer2 is expire.
L3_IRQ	Asserted, when FRC or 31bit counter in Local timer3 is expire.

Interrupt sources in *Table 25-1*, excluding expiration of FRC in local timer are used as wake-up sources.

25.3.1 Global Timer

There is one global timer exists in MCT.

25.3.1.1 Internal Function of Global Timer

FRC in global timer is an up-counter. After it starts, it increments until 64'hffff_ffff_ffff_ffff and returns to 0. Four independent comparators continuously see the value. When it matches with two values, interrupt occurs.

Figure 25-2 illustrates the function of global timer.

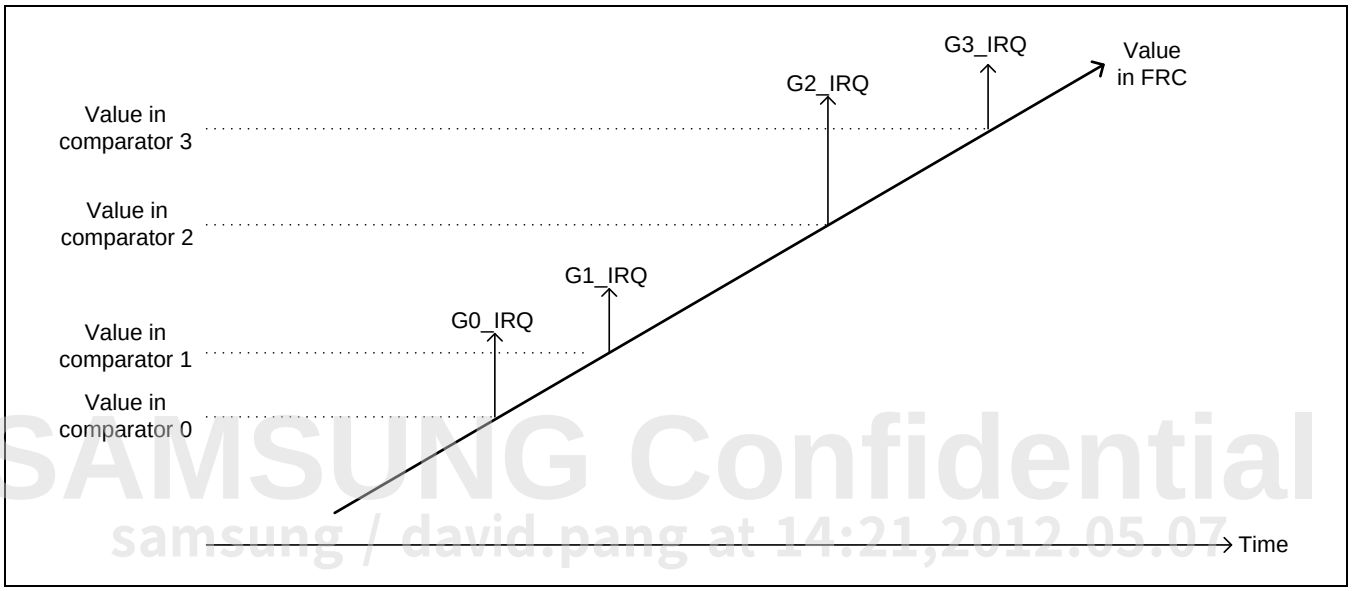


Figure 25-2 Function of Global Timer

If you does not set auto increment at each comparator, it asserts next IRQ after FRC is rounded (FRC current value → FRC = 0 → FRC current value).

If you sets auto increment at comparator with some value, it asserts IRQ periodically.

Figure 25-3 illustrates the periodically asserted interrupt.

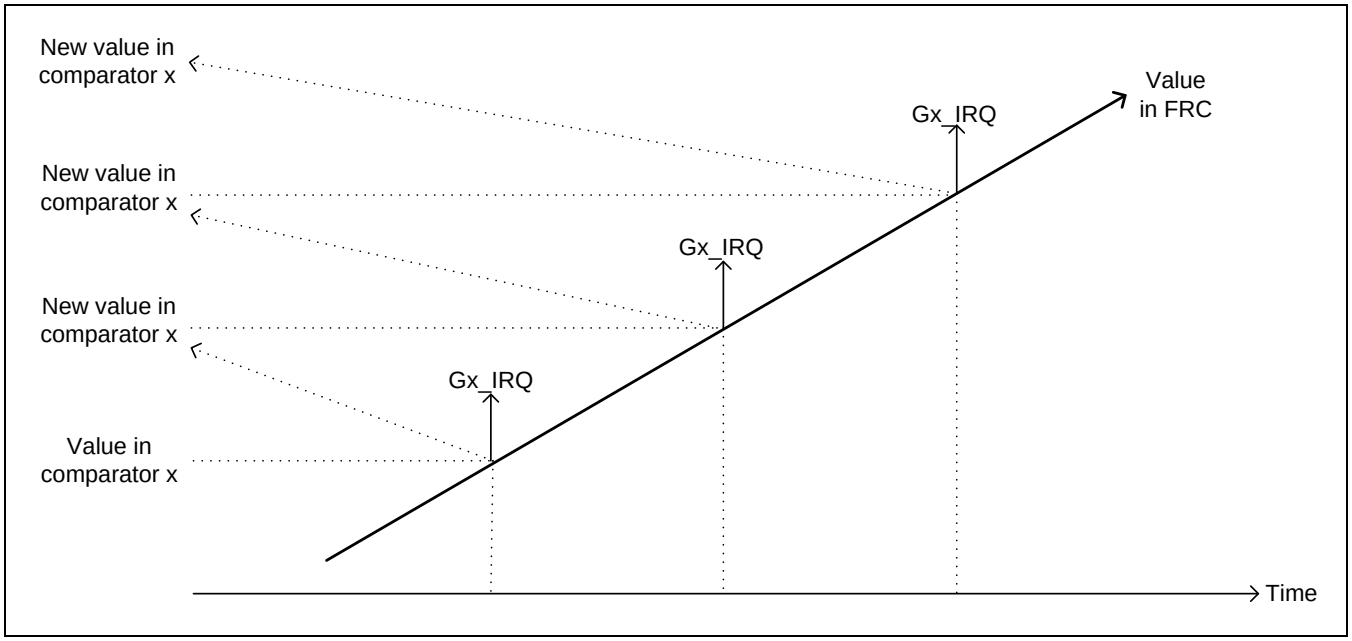


Figure 25-3 Periodically Asserted Interrupt

FRC and related four comparators have 64-bit data width. To read and write the values, you should access two consecutive 32-bit SFR. Because a lower 32-bit SFR is changed faster than an upper, you should better read an upper SFR first.

25.3.2 Local Timer

In MCT, 4 local timer is exists. This section represents each local timer.

25.3.2.1 Internal Function of Local Timer

Figure 25-4 illustrates the two separate timers for various tick.

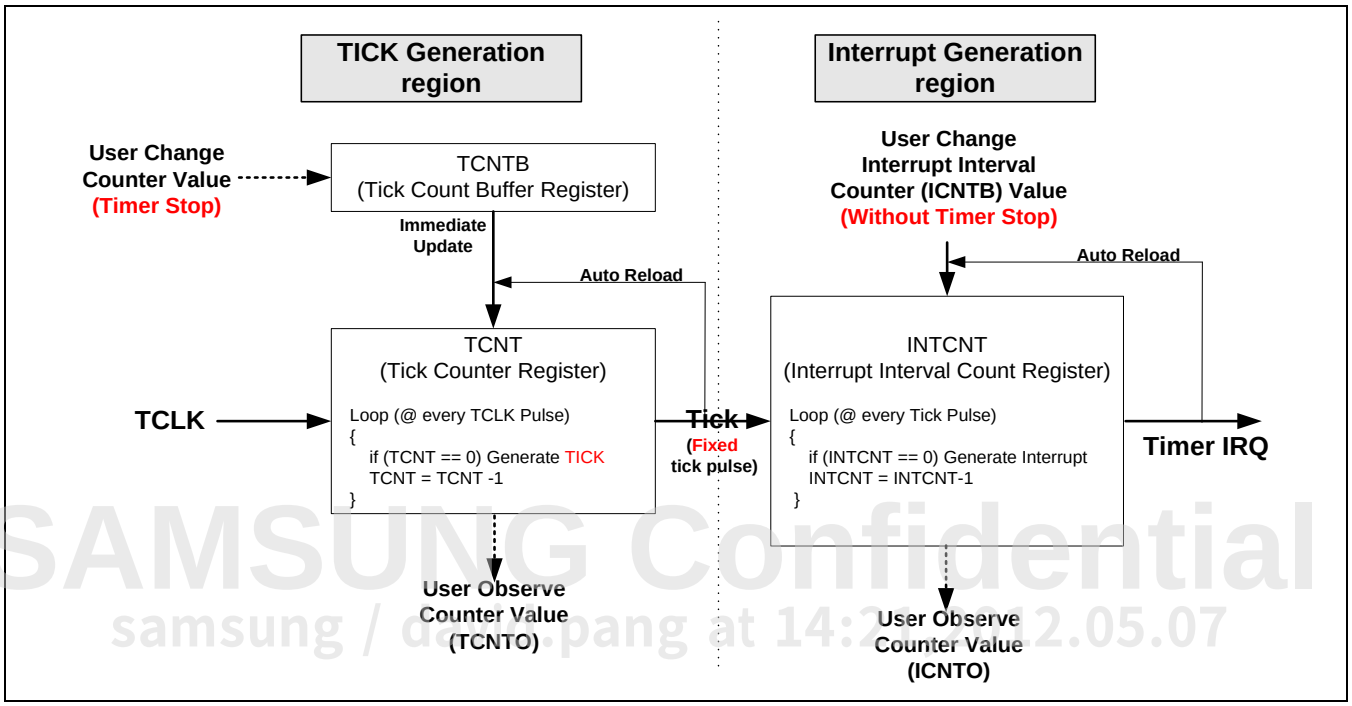


Figure 25-4 Two Separate Timers for Various Tick

There are four separate timers at a local timer in Exynos 4412 SCP for supporting various ticks. You uses the first timer for tick generation and the other for interrupt generation. You also uses four independent SFR sets and logic blocks for tick and interrupt region. Each logic block operates separately. Therefore, you can change interrupt interval independent to reference tick generation. This feature is useful for some power-saving modes.

Figure 25-5 illustrates the Free Running Counter (FRC).

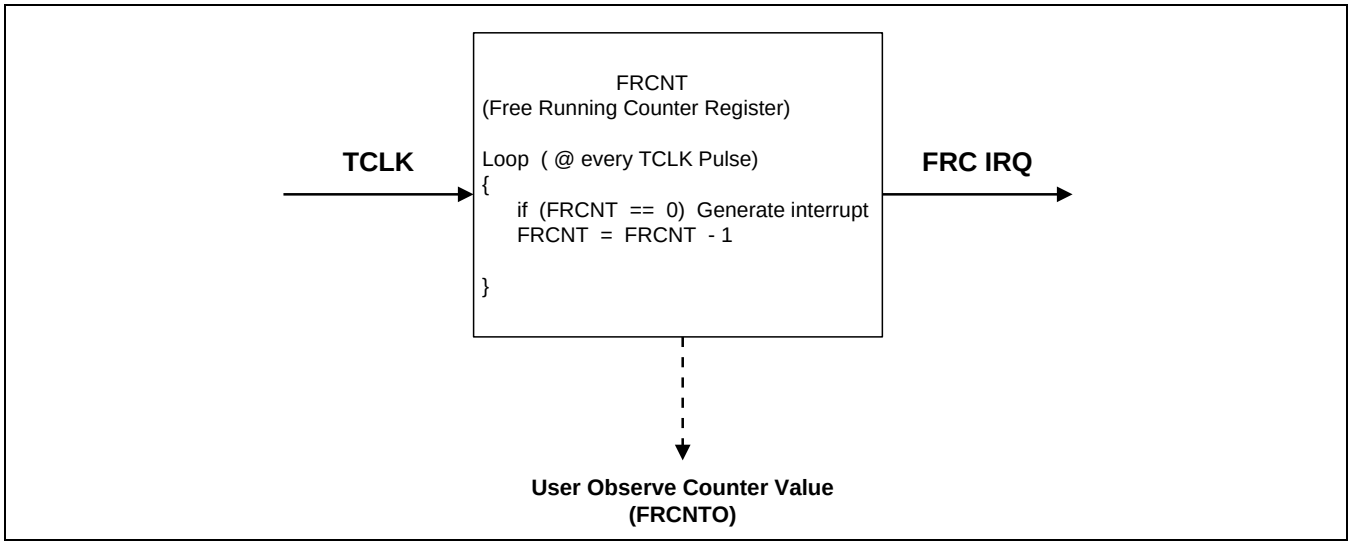


Figure 25-5 FRC Free Running Counter (FRC)

You can implement Linux High Resolution Timer (HRT) with FRC. *Figure 25-5* illustrates that its operation is independent to tick and interrupt timers.

25.3.2.2 Detailed Operation of Local Timer

Figure 25-6 illustrates the timer operation with always on of auto-reload.

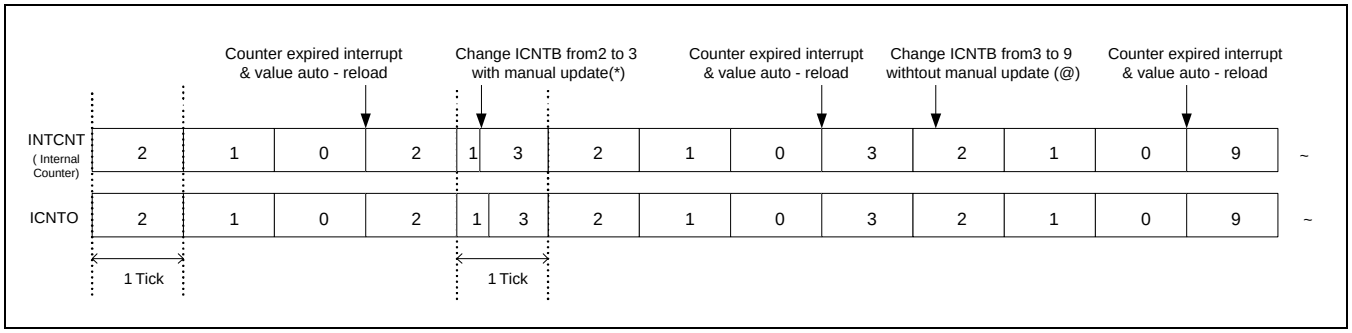


Figure 25-6 Timer Operation with Always on of Auto-reload

Usually, the tick interval is fixed after initial setting and you can change the interrupt interval at run-time. Figure 25-6 illustrates the detailed operation of Interrupt Counter (INTCNT) and interrupt counter observation SFR (ICNTO) with auto-reload. Each rectangular block shows one tick time. Although it changes the interrupt interval, one tick time is fixed because tick and interrupt counter are independent of each other.
Interrupt is asserted when INTCNT value is expires (INTCNT = 0). SW Reads ICNTO to know the elapsed time.

NOTE: As Figure 25-6 illustrates, when it changes ICNTB with interrupt manual update (Ln_ICNTB[31], n=0~3), during that time it applies the new changed value to INTCNT. When it changes ICNTB without interrupt manual update (Ln_ICNTB[31], n=0~3), you applies the new changed value to INTCNT after it expires.
When you requires auto reload, type the Interrupt Type (Ln_TCON[2], n=0~3) as 1.

Figure 25-7 illustrates the timer operation (One-Shot).

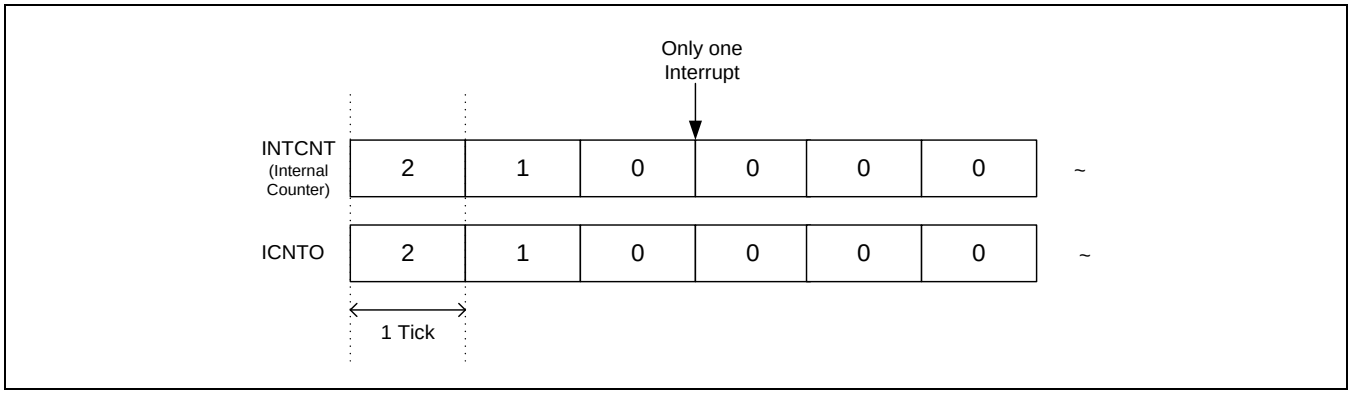


Figure 25-7 Timer Operation (One-Shot)

As Figure 25-7 illustrates, if you requires one-shot interrupt mode, set the Interrupt Type (Ln_TCON[2], n=0~3) to 0. After you assert one interrupt, it masks interrupt. When you configure Ln_TCON again, irrespective of the configuration, you clears the mask and asserts interrupt again.

Figure 25-8 illustrates the FRC operation (always auto-reload).

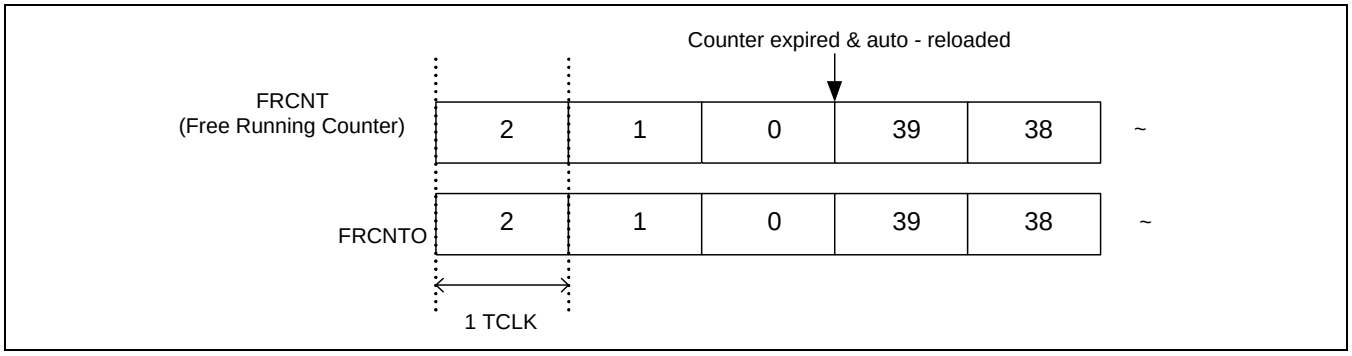


Figure 25-8 FRC Operation (Always Auto-reload)

FRC is down-counter. When it reaches to 0, it auto-reloads and interrupt occurs.

25.3.3 Usage Model

- This section includes the Usage model of timers in MCT.

25.3.3.1 Usage Model of Global Timer

To set the comparator0:

11. Interrupt enables by G_INT_ENB bit set high.
12. Set upper 32-bit value of FRC buffer to G_CNT_U SFR.
13. Poll to check G_CNT_WSTAT[1] = 1 and write 1 to G_CNT_WSTAT[1] to clear Write status bit.
14. Set lower 32-bit value of FRC buffer to G_CNT_L SFR.
15. Poll to check G_CNT_WSTAT[0] = 1 and write 1 to G_CNT_WSTAT[0] to clear Write status bit.
16. Set upper 32-bit value of comparator to G_COMP0_U SFR.
17. Poll to check G_ WSTAT[1] = 1 and write 1 to G_WSTAT[1] to clear Write status bit.
18. Set upper 32-bit value of comparator to G_COMP0_L SFR.
19. Poll to check G_ WSTAT[0] = 1 and write 1 to G_WSTAT[0] to clear Write status bit.
20. Set auto increment value to G_COMP0_ADD_INCR SFR.
21. Poll to check G_ WSTAT[2] = 1 and write 1 to G_WSTAT[2] to clear Write status bit.
22. Set G_TCON[8] as 1 to enable FRM timer, and G_TCON[0] and G_TCON[1] for comparator.
23. Poll to check G_WSTAT[16] = 1 and write 1 to G_WSTAT[16] to clear Write status bit.

Follow steps 6 through 13 to set other comparators.

25.3.3.2 Usage Model of Local Timer

SFR has two sets. You can differentiate them by prefix L0_ , L1_, L2_ and L3_.

To use the local timer:

- Set the count value as high as possible (to improve resolution).
- Do not change TCFG and tick interval at run-time. If you want to change them, stop timers and change.

25.3.3.2.1 Counter Setting

To set the counter:

1. TCNTB: Tick Counter Value = TICINTB + 1
2. ICNTB: Interrupt Counter Value = ICNTB + 1
   FRCNTB: Free Running Counter Value = FRCNTB + 1

25.3.3.2.2 Interrupt and Write Status Check

There are two interrupt sources in each local timer. INT_ENB SFR enables these sources selectively.

When you write a value to TCON, TCNTB, ICNTB, and FRCNTB SFR, the value may not update immediately to internal counter of system timer; because local timer uses XusbXTI (or XXTI) clock for counter. These clocks are slower than PCLK, the operating clock of SFR. If you read WSTAT register continuously, you know the update timing. Refer WSTAT SFR for more information..

25.3.3.2.3 Count Value Update

When TCNTB, and FRCNTB write interrupt are asserted after writing the counter value to TCNTB, and FRCNTB tick timer counter is updated.

Interrupt manual update (ICNTB[31]) updates the interrupt timer at that time. As *25.3.3.2.2 Interrupt and Write Status Check* section explains, check whether it updates internal counter.

The Interrupt Interval mode (TCON[2]) allows interrupt timer to update value in ICNTB automatically after the expiry of interrupt timer counter.

25.3.3.2.4 Start Timer

To set start timer for TCNT and ICNT:

1. Set TCFG SFR to make appropriate TCLK.
   A. Select clock source.
   B. Set pre-scaler and divider value.
2. Set tick counter by writing appropriate values to TCNTB SFR.
3. Poll to check WSTAT[0] = 1 and write 1 to WSTAT[0] to clear Write status bit.
4. Set 31-bit interrupt counter by writing to ICNTB SFR with ICNTB[31] = 1 to update internal interrupt counter .
5. Poll to check WSTAT[1] = 1 and write 1 to WSTAT[1] to clear Write status bit.
6. Write TCON[0] = 1'b1 to run timer.
7. Poll to check WSTAT[3] = 1 and write 1 to WSTAT[3] to clear Write status bit.
8. Write both TCON[3] = 1 to run interrupt and tick timer. Also set interrupt type by TCON[2] at this time.
9. Poll to check WSTAT[3] = 1 and write 1 to WSTAT[3] to clear Write status bit.

To set start timer for FRC:

1. Set free running counter by writing appropriate values to FRCNTB SFR.
2. Poll to check WSTAT[2] = 1 and write 1 to WSTAT[2] to clear Write status bit.
3. Write (TCON[3] = 1'b1) to run FRC.

25.3.3.2.5 STOP Timer

To Stop TCNT AND ICNT timer:

1. Write INT_ENB[0] = 1'b0 to disable Interrupt counter expired (INTCNT = 0) interrupt.
2. Write TCON[1] = 1'b0 to stop internal interrupt counter.
3. Write TCON[0] = 1'b0 To stop internal timer counter.

To stop FRC timer:

1. Write INT_ENB[1] = 1'b0 to disable Free running counter expired (FRCNT = 0) interrupt.
2. Write TCON[3] = 1'b0 to stop Free running counter.

25.3.3.2.6 Change Interval Interrupt at Run-time

To run system timer as *Figure 25-6* illustrates:

1. Set new 31-bit interrupt counter value by writing to ICNTB SFR with ICNTB[31] = 1 to update internal interrupt counter.
2. Poll to check WSTAT[1] = 1and type 1 to WSTAT[1] to clear Write status bit andit updates the new value after the interrupt counter expire.

25.4 Register Description

25.4.1 Register Map Summary

- Base Address: 0x1005_0000

Register	Offset	Description	Reset Value
MCT_CFG	0x0000	Specifies the configures 8-bit-prescaler and clock MUX	0
G_CNT_L	0x0100	Specifies the lower 32-bit value of FRC buffer register	0
G_CNT_U	0x0104	Specifies the upper 32-bit value of FRC buffer register	0
G_CNT_WSTAT	0x0110	Specifies the G_CNT_L&G_CNT_U SFR write status bit	0
G_COMP0_L	0x0200	Specifies the lower 32-bit value of comparator 0	0
G_COMP0_U	0x0204	Specifies the upper 32-bit value of comparator 0	0
G_COMP0_ ADD_INCR	0x0208	Specifies the auto increment value of comparator 0	0
G_COMP1_L	0x0210	Specifies the lower 32-bit value of comparator 1	0
G_COMP1_U	0x0214	Specifies the upper 32-bit value of comparator 1	0
G_COMP1_ ADD_INCR	0x0218	Specifies the auto increment value of comparator 1	0
G_COMP2_L	0x0220	Specifies the lower 32-bit value of comparator 2	0
G_COMP2_U	0x0224	Specifies the upper 32-bit value of comparator 2	0
G_COMP2_ ADD_INCR	0x0228	Specifies the auto increment value of comparator 2	0
G_COMP3_L	0x0230	Specifies the lower 32-bit value of comparator 3	0
G_COMP3_U	0x0234	Specifies the upper 32-bit value of comparator 3	0
G_COMP3_ ADD_INCR	0x0238	Specifies the auto increment value of comparator 3	0
G_TCON	0x0240	Specifies the global timer control register	0
G_INT_CSTAT	0x0244	Specifies the clears interrupt	0
G_INT_ENB	0x0248	Specifies the interrupt enable for G_IRQ0 to 3	0
G_WSTAT	0x024C	Specifies the write status for comparator0 to 3	0
L0_TCNTB	0x0300	Specifies the tick integer count buffer register	0
L0_TCNTO	0x0304	Specifies the tick integer count observation register	0
L0_ICNTB	0x0308	Specifies the interrupt count buffer register	0
L0_ICNTO	0x030C	Specifies the interrupt count observation register	0
L0_FRCNTB	0x0310	Specifies the free running count buffer register	0
L0_FRCNTO	0x0314	Specifies the free running count observation register	0
L0_TCON	0x0320	Specifies the timer control register	0
L0_INT_CSTAT	0x0330	Specifies the clears interrupt	0
L0_INT_ENB	0x0334	Specifies the interrupt enable for L_IRQ0	0
L0_WSTAT	0x0340	Specifies the write status	0

Register	Offset	Description	Reset Value
L1_TCNTB	0x0400	Specifies the Tick integer count buffer register	0
L1_TCNTO	0x0404	Specifies the tick integer count observation register	0
L1_ICNTB	0x0408	Specifies the interrupt count buffer register	0
L1_ICNTO	0x040C	Specifies the interrupt count observation register	0
L1_FRCNTB	0x0410	Specifies the free running count buffer register	0
L1_FRCNTO	0x0414	Specifies the free running count observation register	0
L1_TCON	0x0420	Specifies the timer control register	0
L1_INT_CSTAT	0x0430	Specifies the clears interrupt	0
L1_INT_ENB	0x0434	Specifies the interrupt enable for L_IRQ1	0
L1_WSTAT	0x0440	Specifies the write status	0
L2_TCNTB	0x0500	Specifies the Tick integer count buffer register	0
L2_TCNTO	0x0504	Specifies the tick integer count observation register	0
L2_ICNTB	0x0508	Specifies the interrupt count buffer register	0
L2_ICNTO	0x050C	Specifies the interrupt count observation register	0
L2_FRCNTB	0x0510	Specifies the free running count buffer register	0
L2_FRCNTO	0x0514	Specifies the free running count observation register	0
L2_TCON	0x0520	Specifies the timer control register	0
L2_INT_CSTAT	0x0530	Specifies the clears interrupt	0
L2_INT_ENB	0x0534	Specifies the interrupt enable for L_IRQ2	0
L2_WSTAT	0x0540	Specifies the write status	0
L3_TCNTB	0x0600	Specifies the Tick integer count buffer register	0
L3_TCNTO	0x0604	Specifies the tick integer count observation register	0
L3_ICNTB	0x0608	Specifies the interrupt count buffer register	0
L3_ICNTO	0x060C	Specifies the interrupt count observation register	0
L3_FRCNTB	0x0610	Specifies the free running count buffer register	0
L3_FRCNTO	0x0614	Specifies the free running count observation register	0
L3_TCON	0x0620	Specifies the timer control register	0
L3_INT_CSTAT	0x0630	Specifies the clears interrupt	0
L3_INT_ENB	0x0634	Specifies the interrupt enable for L_IRQ3	0
L3_WSTAT	0x0640	Specifies the write status	0

25.4.1.1 MCT_CFG

- Base Address = 0x1005_0000
- Address = Base Address + 0x0000, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0
INT_MON_SEL	[15:13]	RW	Selects Mux Input for Interrupt Monitor 000 = Interrupt0 at global timer 001 = Interrupt1 at global timer 010 = Interrupt2 at global timer 011 = Interrupt3 at global timer 100 = Interrupt at local timer 0 101 = Interrupt at local timer 1 110 = Interrupt at local timer 2 111 = Interrupt at local timer 3	0x0
TICK_MON_SEL	[12:11]	RW	Selects Mux Input for Tick Monitor 00 = FRC at global timer 01 = Tick at local timer 2 10 = Tick at local timer 0 11 = Tick at local timer 1	0x0
Divider Mux	[10:8]	RW	Selects Mux Input for Timer 000 = 1 001 = 1/2 010 = 1/4 011 = 1/8 100 = 1/16	0x0
Pre-scaler	[7:0]	RW	Prescaler Value for Timer	0x0

Timer Input Clock Frequency = TCLKB/({prescaler value + 1})/{divider value}
{prescaler value} = 1 to 255/{divider value} = 1, 2, 4, 8, 16

NOTE: While timer is running. User should not change clock setting.

25.4.1.2 G_CNT_L

- Base Address = 0x1005_0000
- Address = Base Address + 0x0100, Reset Value = 0

Name	Bit	Type	Description	Reset Value
FRC count buffer	[31:0]	RW	Lower 32-bit value of FRC Buffer register	0x0

25.4.1.3 G_CNT_U

- Base Address = 0x1005_0000
- Address = Base Address + 0x0104, Reset Value = 0

Name	Bit	Type	Description	Reset Value
FRC count buffer	[31:0]	RW	Upper 32-bit value of FRC Buffer register	0x0

25.4.1.4 G_CNT_WSTAT

- Base Address = 0x1005_0000
- Address = Base Address + 0x0110, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
G_CNT_U write status	[1]	RW	G_CNT_U Write Status Bit After user writes value to G_CNT_U, MCT asserts this bit. Write of "1" clears this bit.	0x0
G_CNT_L write status	[0]	RW	G_CNT_L Write Status Bit After user writes value to G_CNT_L, MCT asserts this bit. Write of "1" clears this bit.	0x0

25.4.1.5 G_COMP0_L

- Base Address = 0x1005_0000
- Address = Base Address + 0x0200, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Comparator 0 buffer	[31:0]	RW	Lower 32-bit value of comparator 0	0x0

25.4.1.6 G_COMP0_U

- Base Address = 0x1005_0000
- Address = Base Address + 0x0204, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Comparator 0 buffer	[31:0]	RW	Upper 32-bit value of comparator 0	0x0

25.4.1.7 G_COMP0_ADD_INCR

- Base Address = 0x1005_0000
- Address = Base Address + 0x0208, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Auto increment value	[31:0]	RW	When you enable auto-increment, MCT adds this value to comparator 0 for periodic interrupt.	0x0

25.4.1.8 G_COMP1_L

- Base Address = 0x1005_0000
- Address = Base Address + 0x0210, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Comparator 1 buffer	[31:0]	RW	Lower 32-bit value of Comparator 1	0x0

25.4.1.9 G_COMP1_U

- Base Address = 0x1005_0000
- Address = Base Address + 0x0214, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Comparator 1 buffer	[31:0]	RW	Upper 32-bit value of Comparator 1	0x0

25.4.1.10 G_COMP1_ADD_INCR

- Base Address = 0x1005_0000
- Address = Base Address + 0x0218, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Auto increment value	[31:0]	RW	When you enable auto-increment, MCT adds this value to comparator 1 for periodic interrupt.	0x0

25.4.1.11 G_COMP2_L

- Base Address = 0x1005_0000
- Address = Base Address + 0x0220, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Comparator 2 buffer	[31:0]	RW	Lower 32-bit value of Comparator 2	0x0

25.4.1.12 G_COMP2_U

- Base Address = 0x1005_0000
- Address = Base Address + 0x0224, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Comparator 2 buffer	[31:0]	RW	Upper 32-bit value of Comparator 2	0x0

25.4.1.13 G_COMP2_ADD_INCR

- Base Address = 0x1005_0000
- Address = Base Address + 0x0228, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Auto increment value	[31:0]	RW	When you enable auto-increment, MCT adds this value to comparator 2 for periodic interrupt.	0x0

25.4.1.14 G_COMP3_L

- Base Address = 0x1005_0000
- Address = Base Address + 0x0230, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Comparator 3 buffer	[31:0]	RW	Lower 32-bit value of Comparator 3	0x0

25.4.1.15 G_COMP3_U

- Base Address = 0x1005_0000
- Address = Base Address + 0x0234, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Comparator 3 buffer	[31:0]	RW	Upper 32-bit value of Comparator 3	0x0

25.4.1.16 G_COMP3_ADD_INCR

- Base Address = 0x1005_0000
- Address = Base Address + 0x0238, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Auto increment value	[31:0]	RW	When you enable auto-increment, MCT adds this value to comparator 3 for periodic interrupt.	0x0

25.4.1.17 G_TCON

- Base Address = 0x1005_0000
- Address = Base Address + 0x0240, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	0x0
Timer enable	[8]	RW	Timer enable 1'b0: Global FRC is disabled and the counter does not increment. You can read and write all registers. 1'b1: Global FRC is enabled and the counter increments normally.	0x0
Auto-increment 3	[7]	RW	1'b0: Single Shot Mode When the global FRC reaches G_COMP3 value, it sets the event flag. It is the responsibility of software to update the G_COMP3 value to get further events. 1'b1: Auto Increment Mode Each time the counter reaches G_COMP3 value, it increments G_COMP3 register with G_COMP3_ADD_INCR register. Therefore, MCT sets further events periodically without any software update.	0x0
Comp3 enable	[6]	RW	If this bit is set to one, it allows the comparison between the global FRC and the 64-bit G_COMP3 register.	0x0
Auto-increment2	[5]	RW	1'b0: Single Shot Mode When the global FRC reaches the G_COMP2 value, it sets the event flag. It is the responsibility of software to update the G_COMP2 value to get further events. 1'b1: Auto Increment Mode Each time the counter reaches the G_COMP2 value, MCT increments the G_COMP2 register with G_COMP2_ADD_INCR register. Therefore, MCT sets further events periodically without any software update.	0x0
Comp2 enable	[4]	RW	If this bit is set to one, it allows the comparison between the global FRC and the 64-bit G_COMP2 register.	0x0
Auto-increment 1	[3]	RW	1'b0: Single Shot Mode When the global FRC reaches the G_COMP1 value, it sets the event flag. It is the responsibility of software to update G_COMP1 value to get further events. 1'b1: Auto Increment Mode Each time the counter reaches the G_COMP1 value, MCT increments G_COMP1 register with G_COMP1_ADD_INCR register. Therefore, MCT sets further events periodically without any software update.	0x0
Comp1 enable	[2]	RW	If this bit is set to one,, it allows the comparison between the global FRC and the 64-bit G_COMP1 register.	0x0
Auto-increment0	[1]	RW	1'b0: Single Shot Mode When the global FRC reaches the G_COMP0 value, it sets the event flag. It is the responsibility of software to	0x0

Name	Bit	Type	Description	Reset Value
			update the G_COMP0 value to get further events. 1'b1: Auto Increment Mode Each time the counter reaches the G_COMP0 value, MCT increments G_COMP0 register with G_COMP0_ADD_INCR register. Therefore, MCT sets further events periodically without any software update.	
Comp0 enable	[0]	RW	If this bit is set to one,, it allows the comparison between the global FRC and the 64-bit G_COMP0 register.	0x0

When you sets Comp enable and auto-increment bits and FRC in Global Timer reaches to the value in comparator, Global Timer automatically adds the value in G_COMPx_ADD_INCR SFR. With this function, Global Timer can assert interrupt periodically.

User can read the automatically changed value through G_COMPx_L and G_COMPx_U SFR.

25.4.1.18 G_INT_CSTAT

- Base Address = 0x1005_0000
- Address = Base Address + 0x0244, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
Comparator3 match status	[3]	RW	When the value in comparator3 and FRC in global timer are matched, G_IRQ3 (when-enabled) and this bit are asserted. Clear by writing "1" on this bit.	0x0
Comparator2 match status	[2]	RW	When the value in comparator2 and FRC in global timer are matched, G_IRQ2 (when-enabled) and this bit are asserted. Clear by writing "1" on this bit.	0x0
Comparator1 match status	[1]	RW	When the value in comparator1 and FRC in global timer are matched, G_IRQ1 (when-enabled) and this bit are asserted. Clear by writing "1" on this bit.	0x0
Comparator0 match status	[0]	RW	When the value in comparator0 and FRC in global timer are matched, G_IRQ0 (when-enabled) and this bit are asserted. Clear by writing "1" on this bit.	0x0

25.4.1.19 G_INT_ENB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0248, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
C_INT3_ENABLE	[3]	RW	Interrupt Enable for G_IRQ3 0 = Disables G_IRQ3 1 = Enables G_IRQ3	0x0
C_INT2_ENABLE	[2]	RW	Interrupt Enable for G_IRQ2 0 = Disables G_IRQ3 1 = Enables G_IRQ3	0x0
C_INT1_ENABLE	[1]	RW	Interrupt Enable for G_IRQ1 0 = Disables G_IRQ3 1 = Enables G_IRQ3	0x0
C_INT0_ENABLE	[0]	RW	Interrupt Enable for G_IRQ1 0 = Disables G_IRQ3 1 = Enables G_IRQ3	0x0

25.4.1.20 G_WSTAT

- Base Address = 0x1005_0000
- Address = Base Address + 0x024C, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0x0
G_TCON write status	[16]	RW	G_TCON Write Status Bit After user writes value to G_TCON, MCTasserts this bit. Write of "1" clears this bit.	0x0
RSVD	[15]	–	Reserved	0x0
G_COMP3_ ADD_INCR write status	[14]	RW	G_COMP3_ADD_INCR Write Interrupt Status Bit After user writes value to G_COMP3_ADD_INCR, MCTasserts this bit. Write of "1" clears this bit.	0x0
G_COMP3_U write status	[13]	RW	G_COMP3_U Write Interrupt Status Bit After user writes value to G_COMP3_U, MCTasserts this bit. Write of "1" clears this bit.	0x0
G_COMP3_L write status	[12]	RW	G_COMP3_L Write Interrupt Status Bit After user writes value to G_COMP3_L, MCTasserts this bit. Write of "1" clears this bit.	0x0
RSVD	[11]	–	Reserved	0x0
G_COMP2_ ADD_INCR write status	[10]	RW	G_COMP2_ADD_INCR Write Interrupt Status Bit After user writes value to G_COMP2_ADD_INCR, MCTasserts this bit. Write of "1" clears this bit.	0x0
G_COMP2_U write status	[9]	RW	G_COMP2_U Write Interrupt Status Bit After user writes value to G_COMP2_U, MCT asserts this bit. Write of "1" clears this bit.	0x0
G_COMP2_L write status	[8]	RW	G_COMP2_L Write Interrupt Status Bit After user writes value to G_COMP2_L, MCT asserts this bit. Write of "1" clears this bit.	0x0
RSVD	[7]	–	Reserved	0x0
G_COMP1_ ADD_INCR write status	[6]	RW	G_COMP1_ADD_INCR Write Status Bit After user writes value to G_COMP1_ADD_INCR, MCT asserts this bit. Write of "1" clears this bit.	0x0
G_COMP1_U write status	[5]	RW	G_COMP1_U Write Status Bit After user writes value to G_COMP1_U, MCT asserts this bit. Write of "1" clears this bit.	0x0
G_COMP1_L write status	[4]	RW	G_COMP1_L Write Status Bit After user writes value to G_COMP1_L, MCT asserts this bit. Write of "1" clears this bit.	0x0
RSVD	[3]	–	Reserved	0x0
G_COMP0_ ADD_INCR write status	[2]	RW	G_COMP0_ADD_INCR Write Status Bit After user writes value to G_COMP0_ADD_INCR, MCT asserts this bit. Write of "1" clears this bit.	0x0

Name	Bit	Type	Description	Reset Value
G_COMP0_U write status	[1]	RW	G_COMP0_U Write Status Bit After user writes value to G_COMP0_U, MCT asserts this bit. Write of "1" clears this bit.	0x0
G_COMP0_L write status	[0]	RW	G_COMP0_L Write Status Bit After user writes value to G_COMP0_L, MCT asserts this bit. Write of "1" clears this bit.	0x0

25.4.1.21 L0_TCNTB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0300, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Tick count buffer	[31:0]	RW	Tick count buffer register for local timer 0	0x0

NOTE: Real Timer Counter Value = L0_TCNTB + 1, do not use 0 for L0_TCNTB

25.4.1.22 L0_TCNTO

- Base Address = 0x1005_0000
- Address = Base Address + 0x0304, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Tick count observation	[31:0]	R	Tick Count Observation register for local timer 0	0x0

25.4.1.23 L0_ICNTB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0308, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Interrupt manual update	[31]	W	0 = No operation 1 = Update L0_ICNTB This bit is auto-cleared.	0x0
Interrupt count buffer	[30:0]	RW	Interrupt Count Buffer register for local timer 0	0x0

NOTE: Real Interrupt Counter Value = L0_ICNTB + 1.
If L0_ICNTB value is 0, interrupt occurs at every TICK.

25.4.1.24 L0_ICNTO

- Base Address = 0x1005_0000
- Address = Base Address + 0x030C, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
Tick count observation	[30:0]	R	Interrupt count observation register for local timer 0	0x0

25.4.1.25 L0_FRCNTB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0310, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Free running count buffer	[31:0]	RW	Free Running Count Buffer register for local timer 0	0x0

NOTE: Real Interrupt Counter Value = L0_FRCNTB + 1.

25.4.1.26 L0_FRCNTO

- Base Address = 0x1005_0000
- Address = Base Address + 0x0314, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Free running count observation	[31:0]	R	Free Running Count Observation register for local timer 0	0x0

25.4.1.27 L0_TCON

- Base Address = 0x1005_0000
- Address = Base Address + 0x0320, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
FRC start/stop	[3]	R/W	0 = Stops timer 1 = Starts timer	0x0
Interrupt Type	[2]	R/W	0 = One shot mode 1 = Interval mode (auto-reload)	0x0
Interrupt start/stop	[1]	R/W	0 = Stops timer 1 = Starts timer	0x0
Timer start/stop	[0]	R/W	0 = Stops timer 1 = Starts timer	0x0

If you want to use the one shot mode for interrupt, set L0_TCON[2] as 0. In that case, the interrupt occurs only once and it automatically de-asserts Interrupt Start/Stop bit.

If you want to use interval mode for interrupt, set L0_TCON[2] as 1. In that case, it automatically reloads the value in ICNTB to INTCNT counter when INTCNT expires.

25.4.1.28 L0_INT_CSTAT

- Base Address = 0x1005_0000
- Address = Base Address + 0x0330, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
FRCCNT counter expired Status	[1]	RW	Free running Counter Expired (L0_FRCCNT = 0) Interrupt Status Bit When the timer interrupt occurs, MCT asserts this bit. Write of "1" clears this bit. Interrupt signal (when enabled): L0_IRQ	0x0
INTCNT counter expired Status	[0]	RW	Interrupt counter expired (L0_INTCNT = 0) Interrupt Status Bit. When the timer interrupt occurs, MCT asserts this bit. Write of "1" clears this bit. Interrupt signal (when enabled): L0_IRQ	0x0

25.4.1.29 L0_INT_ENB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0334, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
FRCEIE	[1]	RW	Free running counter expired (L0_FRCCNT = 0) Interrupt Enable 0 = Disables interrupt 1 = Enables interrupt	0x0
ICNTEIE	[0]	RW	Interrupt counter expired (L0_INTCNT = 0) Interrupt Enable 0 = Disables interrupt 1 = Enables interrupt	0x0

25.4.1.30 L0_WSTAT

- Base Address = 0x1005_0000
- Address = Base Address + 0x0340, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
L0_TCON write status	[3]	RW	L0_TCON Write Interrupt Status Bit After user writes value to L0_TCON: MCT asserts this bit. Write of "1" clears this bit.	0x0
L0_FRCCNTB write status	[2]	RW	L0_TCON Write Interrupt Status Bit After user writes value to TCON: MCT asserts this bit. Write of "1" clears this bit.	0x0
L0_ICNTB write status	[1]	RW	L0_ICNTB Write Interrupt Status Bit After user writes value to ICNTB: MCT asserts this bit. Write of "1" clears this bit.	0x0
L0_TCNTB write status	[0]	RW	L0_TCTNB Write Interrupt Status Bit After user writes value to TCNTB: MCT asserts this bit. Write of "1" clears this bit.	0x0

25.4.1.31 L1_TCNTB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0400, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Tick count buffer	[31:0]	RW	Tick Count Buffer register for local timer 1	0x0

NOTE: Real Timer Counter Value = L1_TCNTB + 1, Do not use 0 for L1_TCNTB

25.4.1.32 L1_TCNTO

- Base Address = 0x1005_0000
- Address = Base Address + 0x0404, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Tick count observation	[31:0]	R	Tick Count Observation register for local timer 1	0x0

25.4.1.33 L1_ICNTB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0408, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Interrupt manual update	[31]	W	0 = No operation 1 = Update L1_ICNTB This bit is auto-cleared.	0x0
Interrupt count buffer	[30:0]	RW	Interrupt Count Buffer register for local timer 1	0x0

NOTE: Real Interrupt Counter Value = L1_ICNTB + 1. If L1_ICNTB value is 0, interrupt occurs at every TICK.

25.4.1.34 L1_ICNTO

- Base Address = 0x1005_0000
- Address = Base Address + 0x040C, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
Tick count observation	[30:0]	R	Interrupt Count Observation register for local timer 1	0x0

25.4.1.35 L1_FRCNTB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0410, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Free running count buffer	[31:0]	RW	Free Running Count Buffer register for local timer 1	0x0

NOTE: Real Interrupt Counter Value = L1_FRCNTB + 1.

25.4.1.36 L1_FRCNTO

- Base Address = 0x1005_0000
- Address = Base Address + 0x0414, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Free running count observation	[31:0]	R	Free Running Count Observation register for local timer 1	0x0

25.4.1.37 L1_TCON

- Base Address = 0x1005_0000
- Address = Base Address + 0x0420, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
FRC start/stop	[3]	RW	0 = Stops timer 1 = Starts timer	0x0
Interrupt type	[2]	RW	0 = One shot mode 1 = Interval mode (auto-reload)	0x0
Interrupt start/stop	[1]	RW	0 = Stops timer 1 = Starts timer	0x0
Timer start/stop	[0]	RW	0 = Stops timer 1 = Starts timer	0x0

If you want to use the one shot mode for interrupt, set L1_TCON[2] as 0. In that case, the interrupt occurs only once and it automatically de-asserts Interrupt Start/Stop bit.

If you want to use interval mode for interrupt, set L1_TCON[2] as 1. In that case, it automatically reloads the value in ICNTB to INTCNT counter when INTCNT expires.

25.4.1.38 L1_INT_CSTAT

- Base Address = 0x1005_0000
- Address = Base Address + 0x0430, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
FRCCNT counter expired status	[1]	RW	Free running counter expired (L1_FRCCNT = 0) Interrupt Status Bit: MCT asserts this bit. Write of "1" clears this bit. Interrupt signal (when enabled): L1_IRQ	0x0
INTCNT counter expired status	[0]	RW	Iinterrupt counter expired (L1_INTCNT = 0) Interrupt Status Bit: MCT asserts this bit. Write of "1" clears this bit. Interrupt signal (when enabled): L1_IRQ	0x0

25.4.1.39 L1_INT_ENB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0434, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
FRCEIE	[1]	RW	Free running counter expired (L1_FRCCNT = 0) Interrupt Enable 0 = Disables interrupt 1 = Enables Interrupt	0x0
ICNTEIE	[0]	RW	Interrupt counter expired (L1_INTCNT = 0) Interrupt Enable. 0 = Disables interrupt 1 = Enables Interrupt	0x0

25.4.1.40 L1_WSTAT

- Base Address = 0x1005_0000
- Address = Base Address + 0x0440, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
L1_TCON write status	[3]	RW	L1_TCON Write Interrupt Status Bit After user writes value to L1_TCON, MCT asserts this bit. Write of "1" clears this bit.	0x0
L1_FRCCNTB write status	[2]	R/W	L1_TCON Write Interrupt Status Bit After user writes value to TCON, MCT asserts this bit. Write of "1" clears this bit.	0x0
L1_ICNTB write status	[1]	R/W	L1_ICNTB Write Interrupt Status Bit After user writes value to ICNTB, MCT asserts this bit. Write of "1" clears this bit.	0x0
L1_TCNTB write status	[0]	R/W	L1_TCTNB Write Interrupt Status Bit After user writes value to TCNTB, MCT asserts this bit. Write of "1" clears this bit.	0x0

25.4.1.41 L2_TCNTB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0500, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Tick count buffer	[31:0]	RW	Tick count buffer register for local timer 2	0x0

NOTE: Real Timer Counter Value = L2_TCNTB + 1, do not use 0 for L2_TCNTB

25.4.1.42 L2_TCNTO

- Base Address = 0x1005_0000
- Address = Base Address + 0x0504, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Tick count observation	[31:0]	R	Tick Count Observation register for local timer 2	0x0

25.4.1.43 L2_ICNTB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0508, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Interrupt manual update	[31]	W	0 = No operation 1 = Update L2_ICNTB This bit is auto-cleared.	0x0
Interrupt count buffer	[30:0]	RW	Interrupt Count Buffer register for local timer 2	0x0

NOTE: Real Interrupt Counter Value = L2_ICNTB + 1.
If L2_ICNTB value is 0, interrupt occurs at every TICK.

25.4.1.44 L2_ICNTO

- Base Address = 0x1005_0000
- Address = Base Address + 0x050C, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
Tick count observation	[30:0]	R	Interrupt count observation register for local timer 2	0x0

25.4.1.45 L2_FRCNTB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0510, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Free running count buffer	[31:0]	RW	Free Running Count Buffer register for local timer 2	0x0

NOTE: Real Interrupt Counter Value = L2_FRCNTB + 1.

25.4.1.46 L2_FRCNTO

- Base Address = 0x1005_0000
- Address = Base Address + 0x0514, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Free running count observation	[31:0]	R	Free Running Count Observation register for local timer 2	0x0

25.4.1.47 L2_TCON

- Base Address = 0x1005_0000
- Address = Base Address + 0x0520, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
FRC start/stop	[3]	R/W	0 = Stops timer 1 = Starts timer	0x0
Interrupt Type	[2]	R/W	0 = One shot mode 1 = Interval mode (auto-reload)	0x0
Interrupt start/stop	[1]	R/W	0 = Stops timer 1 = Starts timer	0x0
Timer start/stop	[0]	R/W	0 = Stops timer 1 = Starts timer	0x0

If you want to use the one shot mode for interrupt, set L2_TCON[2] as 0. In that case, the interrupt occurs only once and it automatically de-asserts Interrupt Start/Stop bit.

If you want to use interval mode for interrupt, set L2_TCON[2] as 1. In that case, it automatically reloads the value in ICNTB to INTCNT counter when INTCNT expires.

25.4.1.48 L2_INT_CSTAT

- Base Address = 0x1005_0000
- Address = Base Address + 0x0530, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
FRCCNT counter expired Status	[1]	RW	Free running Counter Expired (L2_FRCCNT = 0) Interrupt Status Bit When the timer interrupt occurs, MCT asserts this bit. Write of "1" clears this bit. Interrupt signal (when enabled): L2_IRQ	0x0
INTCNT counter expired Status	[0]	RW	Interrupt counter expired (L2_INTCNT = 0) Interrupt Status Bit. When the timer interrupt occurs, MCT asserts this bit. Write of "1" clears this bit. Interrupt signal (when enabled): L2_IRQ	0x0

25.4.1.49 L2_INT_ENB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0534, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
FRCEIE	[1]	RW	Free running counter expired (L2_FRCCNT = 0) Interrupt Enable 0 = Disables interrupt 1 = Enables interrupt	0x0
ICNTEIE	[0]	RW	Interrupt counter expired (L2_INTCNT = 0) Interrupt Enable 0 = Disables interrupt 1 = Enables interrupt	0x0

25.4.1.50 L2_WSTAT

- Base Address = 0x1005_0000
- Address = Base Address + 0x0540, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
L2_TCON write status	[3]	RW	L2_TCON Write Interrupt Status Bit After user writes value to L2_TCON: MCT asserts this bit. Write of "1" clears this bit.	0x0
L2_FRCCNTB write status	[2]	RW	L2_TCON Write Interrupt Status Bit After user writes value to TCON: MCT asserts this bit. Write of "1" clears this bit.	0x0
L2_ICNTB write status	[1]	RW	L2_ICNTB Write Interrupt Status Bit After user writes value to ICNTB: MCT asserts this bit. Write of "1" clears this bit.	0x0
L2_TCNTB write status	[0]	RW	L2_TCTNB Write Interrupt Status Bit After user writes value to TCNTB: MCT asserts this bit. Write of "1" clears this bit.	0x0

25.4.1.51 L3_TCNTB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0600, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Tick count buffer	[31:0]	RW	Tick count buffer register for local timer 3	0x0

NOTE: Real Timer Counter Value = L3_TCNTB + 1, do not use 0 for L0_TCNTB

25.4.1.52 L3_TCNTO

- Base Address = 0x1005_0000
- Address = Base Address + 0x0604, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Tick count observation	[31:0]	R	Tick Count Observation register for local timer 3	0x0

25.4.1.53 L3_ICNTB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0608, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Interrupt manual update	[31]	W	0 = No operation 1 = Update L3_ICNTB This bit is auto-cleared.	0x0
Interrupt count buffer	[30:0]	RW	Interrupt Count Buffer register for local timer 3	0x0

NOTE: Real Interrupt Counter Value = L3_ICNTB + 1.
If L3_ICNTB value is 0, interrupt occurs at every TICK.

25.4.1.54 L3_ICNTO

- Base Address = 0x1005_0000
- Address = Base Address + 0x060C, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0x0
Tick count observation	[30:0]	R	Interrupt count observation register for local timer 3	0x0

25.4.1.55 L3_FRCNTB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0610, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Free running count buffer	[31:0]	RW	Free Running Count Buffer register for local timer 3	0x0

NOTE: Real Interrupt Counter Value = L3_FRCNTB + 1.

25.4.1.56 L3_FRCNTO

- Base Address = 0x1005_0000
- Address = Base Address + 0x0614, Reset Value = 0

Name	Bit	Type	Description	Reset Value
Free running count observation	[31:0]	R	Free Running Count Observation register for local timer 3	0x0

25.4.1.57 L3_TCON

- Base Address = 0x1005_0000
- Address = Base Address + 0x0620, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
FRC start/stop	[3]	R/W	0 = Stops timer 1 = Starts timer	0x0
Interrupt Type	[2]	R/W	0 = One shot mode 1 = Interval mode (auto-reload)	0x0
Interrupt start/stop	[1]	R/W	0 = Stops timer 1 = Starts timer	0x0
Timer start/stop	[0]	R/W	0 = Stops timer 1 = Starts timer	0x0

If you want to use the one shot mode for interrupt, set L3_TCON[2] as 0. In that case, the interrupt occurs only once and it automatically de-asserts Interrupt Start/Stop bit.

If you want to use interval mode for interrupt, set L3_TCON[2] as 1. In that case, it automatically reloads the value in ICNTB to INTCNT counter when INTCNT expires.

25.4.1.58 L3_INT_CSTAT

- Base Address = 0x1005_0000
- Address = Base Address + 0x0630, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
FRCCNT counter expired Status	[1]	RW	Free running Counter Expired (L3_FRCCNT = 0) Interrupt Status Bit When the timer interrupt occurs, MCT asserts this bit. Write of "1" clears this bit. Interrupt signal (when enabled): L3_IRQ	0x0
INTCNT counter expired Status	[0]	RW	Interrupt counter expired (L3_INTCNT = 0) Interrupt Status Bit. When the timer interrupt occurs, MCT asserts this bit. Write of "1" clears this bit. Interrupt signal (when enabled): L3_IRQ	0x0

25.4.1.59 L3_INT_ENB

- Base Address = 0x1005_0000
- Address = Base Address + 0x0634, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0x0
FRCEIE	[1]	RW	Free running counter expired (L3_FRCCNT = 0) Interrupt Enable 0 = Disables interrupt 1 = Enables interrupt	0x0
ICNTEIE	[0]	RW	Interrupt counter expired (L3_INTCNT = 0) Interrupt Enable 0 = Disables interrupt 1 = Enables interrupt	0x0

25.4.1.60 L3_WSTAT

- Base Address = 0x1005_0000
- Address = Base Address + 0x0640, Reset Value = 0

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0x0
L3_TCON write status	[3]	RW	L3_TCON Write Interrupt Status Bit After user writes value to L3_TCON: MCT asserts this bit. Write of "1" clears this bit.	0x0

L3_FRCCNTB write status	[2]	RW	L3_TCON Write Interrupt Status Bit After user writes value to TCON: MCT asserts this bit. Write of "1" clears this bit.	0x0
L3_ICNTB write status	[1]	RW	L3_ICNTB Write Interrupt Status Bit After user writes value to ICNTB: MCT asserts this bit. Write of "1" clears this bit.	0x0
L3_TCNTB write status	[0]	RW	L3_TCTNB Write Interrupt Status Bit After user writes value to TCNTB: MCT asserts this bit. Write of "1" clears this bit.	0x0

26 Watchdog Timer

26.1 Overview

Watchdog Timer (WDT) in Exynos 4412 SCP is a timing device. You can use this device to resume the controller operation after malfunctioning due to noise and system errors. You can use WDT as a normal 16-bit interval timer to request interrupt service. WDT generates the reset signal.

26.2 Features of WDT

The features of WDT are:

- Supports normal interval timer mode with interrupt request.
- Activates internal reset signal if the timer count value reaches 0 (time-out).
- Supports level-triggered interrupt mechanism.

26.3 Functional Description

This section includes:

- WDT operation
- WTDAT and WTCNT
- WDT Start
- Consideration of debugging environment

26.3.1 WDT Operation

WDT uses PCLK as its source clock. The 8-bit Prescaler prescales the PCLK frequency to generate the corresponding WDT and it divides the resulting frequency again.

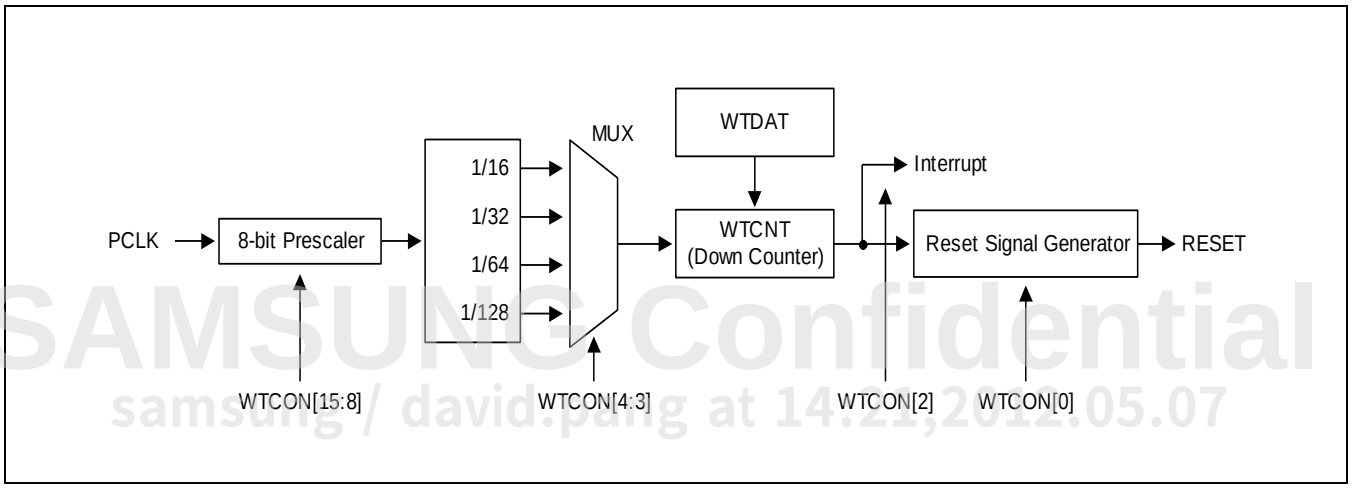


Figure 26-1 Watchdog Timer Block Diagram

Figure 26-1 illustrates the functional block diagram of WDT.

The Watchdog Timer Control (WTCON) specifies the prescaler value and frequency division factor. Valid prescaler values range from 0 to (2^8–1). You can select the frequency division factor as: 16, 32, 64, or 128.

Use this equation to calculate the WDT clock frequency and the duration of each timer clock cycle:

t_watchdog = 1/(PCLK/(Prescaler value + 1)/Division_factor)

26.3.2 WTDAT and WTCNT

After you enable the WDT, you cannot reload the value of the Watchdog Timer Data (WTDAT) register automatically into the Watchdog Timer Counter (WTCNT) register. Therefore, you must write an initial value to the WTCN register before WDT starts.

26.3.3 WDT Start

To start WDT, set WTCON[0] and WTCON[5] as 1.

26.3.4 Consideration of Debugging Environment

WDT should not operate if the Exynos 4412 SCP is in debug mode that uses Embedded In Circuit Debugger (ICE).

WDT determines whether CPU core is currently in the debug mode from the CPU core signal (DBGACK signal). After CPU core asserts the DBGACK signal, it does not activate the reset output of WDT as WDT expires.

26.4 Register Description

26.4.1 Register Map Summary

- Base Address: 0x1006_000

Register	Offset	Description	Reset Value
WTCON	0x0000	Watchdog timer control register	0x0000_8021
WTDAT	0x0004	Watchdog timer data register	0x0000_8000
WTCNT	0x0008	Watchdog timer count register	0x0000_8000
WTCLRINT	0x000C	Watchdog timer interrupt clear register	Undefined

26.4.1.1 WTCON

- Base Address: 0x1006_000
- Address = Base Address + 0x0000, Reset Value = 0x0000_8021

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
Prescaler value	[15:8]	RW	Prescaler value. The valid range is from 0 to (2^8–1).	0x80
RSVD	[7:6]	–	Reserved These two bits should be 00 in normal operation.	00
WDT timer	[5]	RW	Enables or disables WDT bit. 0 = Disables WDT bit 1 = Enables WDT bit	1
Clock select	[4:3]	RW	Determines the clock division factor. 00 = 16 01 = 32 10 = 64 11 = 128	00
Interrupt generation	[2]	RW	Enables or disables interrupt bit. 0 = Disables interrupt bit 1 = Enables interrupt bit	0
RSVD	[1]	–	Reserved. This bit should be 0 in normal operation.	0
Reset enable/disable	[0]	RW	Enables or disables WDT output bit for reset signal. 0 = Disables the reset function of the watchdog timer. 1 = Asserts reset signal of the Exynos 4412 SCP at watchdog time-out.	1

The WTCON register:

- Allows you to enable/disable the watchdog timer
- Selects the clock signal from four different sources
- Enables/disables interrupts
- Enables/disables the watchdog timer output

You can use WDT to restart the Exynos 4412 SCP to recover from malfunction. If you do not want to restart the controller, disable WDT.

If you want to use the normal timer that WDT provides, enable the interrupt and disable the WDT.

26.4.1.2 WTDAT

- Base Address: 0x1006_000
- Address = Base Address + 0x0004, Reset Value = 0x0000_8000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
Count reload value	[15:0]	RW	WDT count value for reload.	0x8000

The WTDAT register specifies the time-out duration. You cannot load the content of WTDAT into the timer counter at initial WDT operation. However, by using 0x8000 (initial value) drives the WDT counter first time-out. In this case, WDT counter logic reloads the value of WTDAT automatically into WTCNT.

26.4.1.3 WTCNT

- Base Address: 0x1006_000
- Address = Base Address + 0x0008, Reset Value = 0x0000_8000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
Count value	[15:0]	RW	The current count value of the WDT	0x8000

The WTCNT register contains the current count values for the WDT during normal operation. WDT counter logic cannot automatically load the content of WTDAT register into the timer count register if it enables the WDT initially. Therefore, you should set the WTCNT register to an initial value before enabling it.

26.4.1.4 WTCLRINT

- Base Address: 0x1006_000
- Address = Base Address + 0x000C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
Interrupt clear	[31:0]	RW	Write any value to clear the interrupt	–

You can use the WTCLRINT register to clear the interrupt. Interrupt service routine is responsible to clear the relevant interrupt after the interrupt service is complete. Writing any values on this register clears the interrupt. Reading on this register is not allowed.

27 Real Time Clock (RTC)

27.1 Overview

Real Time Clock (RTC) unit can operate using the backup battery while the system power is off. Although power is off, backup battery can store the time by Second, Minute, Hour, Day of the week, Day, Month, and Year data. The RTC unit works with an external 32.768 kHz crystal and performs the function of alarm.

27.2 Features of RTC

The features of RTC are:

- Supports BCD Number, that is Second, Minute, Hour, Day of the week, Day, Month, and Year.
- Supports Leap Year Generator
- Supports Alarm Function that is, Alarm-Interrupt or Wake-up from power down modes. The power down mode are: idle, deep idle, stop, deep stop, and sleep.
- Supports Tick Counter Function that is, Tick-Interrupt or Wake-up from power down modes (idle, deep idle, stop, deep stop, and sleep)
- Supports Independent Power Pin (RTCVDD)
- Supports millisecond tick time interrupt for RTOS kernel time tick.

27.2.1 Block Diagram

Figure 27-1 illustrates block diagram of RTC.

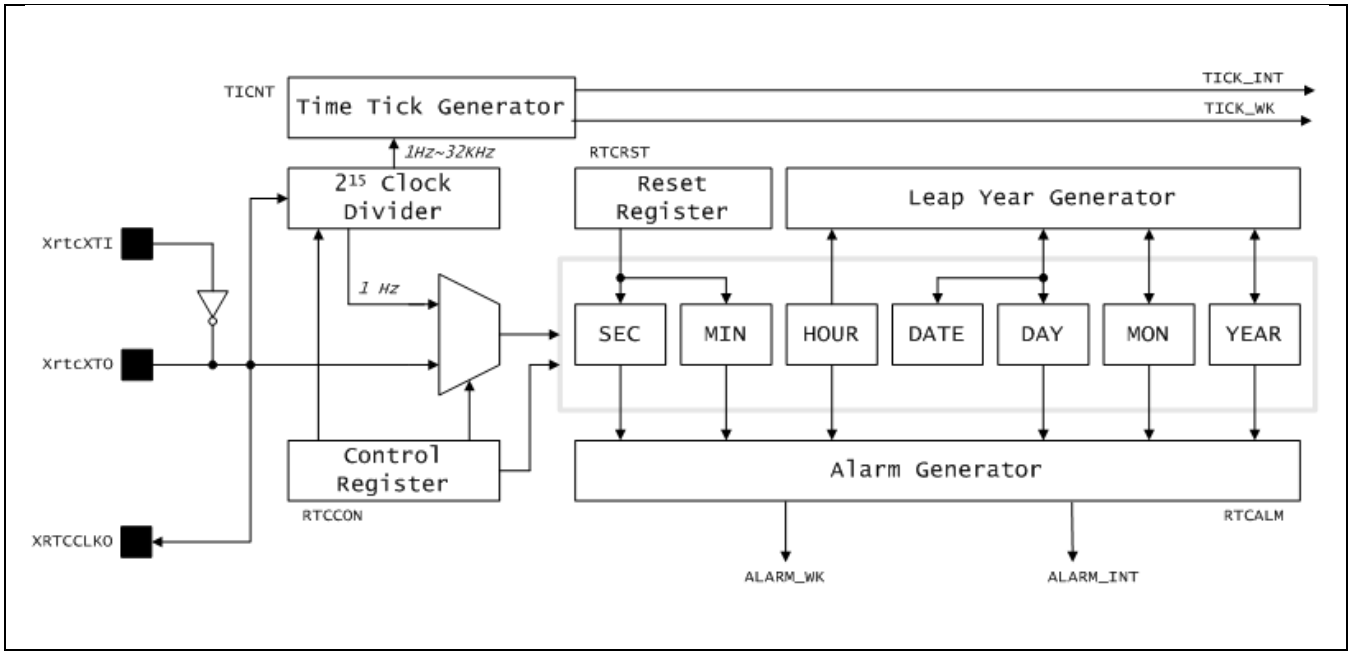


Figure 27-1 RTC Block Diagram

27.3 Leap Year Generator

The leap year generator determines the last day of each month out of 28, 29, 30, or 31. You can calculate this based on the data from BCDDAY, BCDMON, and BCDYEAR. This block considers leap year while deciding on the last day of a month.

NOTE: The BCDYEAR register is 12-bit wide. It can represent maximum three BCD digits. The implicit number of thousands place is 2. Therefore, it can represent years from 400 × n to 400 × n + 999 (n = 0, 1, 2, 3, 4, 5 …).

省略2, 所以最起始的年份是2000

27.4 Read/Write Register

To write the BCD register in RTC block, set the Bit 0 of the RTCCON register. To display the second, minute, hour, day of the week, date, month, and year, the CPU should read the data in BCDSEC, BCDMIN, BCDHOUR, BCDDAYWEEK, BCDDAY, BCDMON, and BCDYEAR registers, respectively, in the RTC block. However, a one-second deviation can exist because it reads multiple registers.

For example, if you read the registers from BCDYEAR to BCDMIN, you can assume the as 2059 (Year), 12 (Month), 31 (Day), 23 (Hour) and 59 (Minute). When you read the BCDSEC register and the value ranges from 1 to 59 (Second), there is no problem. However, if the value is 0 second, the year, month, day, hour, and minute can change to 2060 (Year), 1 (Month), 1 (Day), 0 (Hour) and 0 (Minute) because of the one second deviation. In this case, you must read again from BCDYEAR to BCDSEC if BCDSEC is zero.

27.4.1 Backup Battery Operation

The backup battery can drive the RTC logic. The backup battery supplies the power through the RTCVDD pin into the RTC block, even if the system power is off. If the system is off, you should block the interfaces of the CPU and RTC logic. To minimize power dissipation the backup battery alone drives the oscillation circuit and the BCD counters.

27.5 Alarm Function

The RTC generates ALARM_INT (alarm interrupt) and ALARM_WK (alarm wake-up) at a specific time in the power-off mode or normal operation mode. In normal operation mode, it activates the ALARM_INT. In the power-off mode, it activates the ALARM_WK signal as well as the ALARM_INT. The RTC alarm register (RTCALM) determines the alarm enable/disable status and the condition of the alarm time setting.

27.6 Tick Time Interrupt

The RTC tick timer is an up-counter and raises tick interrupt. The TICNT register contains 32-bits target count value for the interrupt, and the CURTICCNT register contains 32-bits current tick count. If the current tick count reaches the target value specified in TICNT, the tick time interrupt occurs. Then the period of interrupt is as follows:

Period = (n + 1)/(Tick clock source frequency) second (n = tick counter value)

Table 27-1 describes the tick interrupt resolution.

Table 27-1 Tick Interrupt Resolution

Tick Counter Clock Source Selection	Tick Clock Source Frequency (Hz)	Clock Range (s)	Resolution (ms)
4'b0011	4096 (2^12)	0 to 1048575	0.24
4'b0100	2048 (2^11)	0 to 2097151	0.49
4'b0101	1024 (2^10)	0 to 4194303	0.97
4'b0110	512 (2^9)	0 to 8388607	1.95
4'b0111	256 (2^8)	0 to 16777215	3.90
4'b1000	128 (2^7)	0 to 33554431	7.81
4'b1001	64 (2^6)	0 to 67108863	15.62
4'b1010	32 (2^5)	0 to 134217727	31.25
4'b1011	16 (2^4)	0 to 268435455	62.50
4'b1100	8 (2^3)	0 to 536870911	125
4'b1101	4 (2^2)	0 to 1073741823	250
4'b1110	2	0 to 2147483647	500
4'b1111	1	0 to 4294967295	1000

NOTE: You can select the appropriate tick time clock source to extend the tick time resolution.

You can use this RTC time tick for Real Time Operating System (RTOS) kernel time tick. When RTC time tick generates time tick, it synchronizes the time related function of RTOS in real time.

27.7 32.768 kHz X-Tal Connection Example

Figure 27-2 illustrates a circuit of the RTC unit oscillation at 32.768 kHz. The capacitance 20 pF of the load capacitor is an example value. You should adjust this according to the crystal load capacitance.

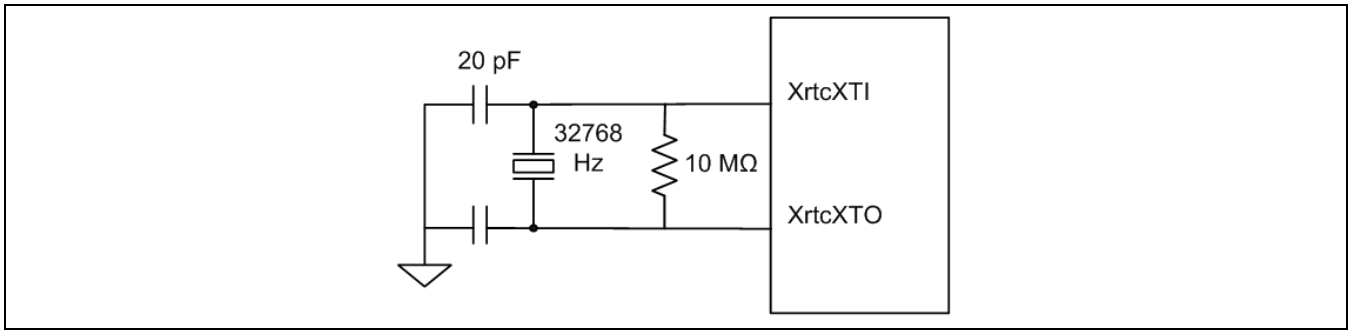


Figure 27-2 Main Oscillator Circuit Example

27.8 I/O Description

Signal	I/O	Description	Pad	Type
XT_RTC_I	Input	32.768 kHz RTC Oscillator Clock Input	XrtcXTI	Dedicated
XT_RTC_O	Output	32.768 kHz RTC Oscillator Clock Output	XrtcXTO	Dedicated
XRTCCLKO	Output	32.768 kHz RTC Clock Output (1.8 to 3.3 V). This signal is turned off by default. You can enable this signal by setting 1 in CLKOUTEN field of RTCCON register. NOTE: To use XRTCCLO, it should supply ALIVE power.	XRTCCLKO	Dedicated

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

27.9 Register Description

27.9.1 Register Map Summary

- Base Address: 0x1007_0000

Register	Offset	Description	Reset Value
INTP	0x0030	Specifies the interrupt pending register	0x0000_0000
RTCCON	0x0040	Specifies the RTC control register	0x0000_0000
TICCNT	0x0044	Specifies the tick time count register	0x0000_0000
RTCALM	0x0050	Specifies the RTC alarm control register	0x0000_0000
ALMSEC	0x0054	Specifies the alarm second data register	0x0000_0000
ALMMIN	0x0058	Specifies the alarm minute data register	0x0000_0000
ALMHOUR	0x005C	Specifies the alarm hour data register	0x0000_0000
ALMDAY	0x0060	Specifies the alarm day data register	0x0000_0000
ALMMON	0x0064	Specifies the alarm month data register	0x0000_0000
ALMYEAR	0x0068	Specifies the alarm year data register	0x0000_0000
BCDSEC	0x0070	Specifies the BCD second register	Undefined
BCDMIN	0x0074	Specifies the BCD minute register	Undefined
BCDHOUR	0x0078	Specifies the BCD hour register	Undefined
BCDDAYWEEK	0x007C	Specifies the BCD day of the week register	Undefined
BCDDAY	0x0080	Specifies the BCD day register	Undefined
BCDMON	0x0084	Specifies the BCD month register	Undefined
BCDYEAR	0x0088	Specifies the BCD year register	Undefined
CURTICCNT	0x0090	Specifies the current tick time counter register	0x0000_0000

这里弄反了

27.9.1.1 INTP

- Base Address: 0x1007_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0
ALARM	[1]	RW	Alarm interrupt pending bit 0 = Interrupt did not occur 1 = Interrupt occurred	0
Time TIC	[0]	RW	Time TIC interrupt pending bit 0 = No interrupt occurred 1 = Interrupt occurred.	0

NOTE: You can clear specific bits of INTP register by writing 1's to the bits that you want to clear regardless of CTLEN field value in RTCCON register.

27.9.1.2 RTCCON

- Base Address: 0x1007_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	0
CLKOUTEN	[9]	RW	Enables RTC clock output on XRTCCLKO pad 0 = Disables RTC clock output on XRTCCLKO pad 1 = Enables RTC clock output on XRTCCLKO pad	0
TICEN	[8]	RW	Enables Tick timer 0 = Disables tick timer 1 = Enables tick timer	0
TICCKSEL	[7:4]	RW	Tick timer sub clock selection 4'b0000 = 32768 Hz 4'b0001 = 16384 Hz 4'b0010 = 8192 Hz 4'b0011 = 4096 Hz 4'b0100 = 2048 Hz 4'b0101 = 1024 Hz 4'b0110 = 512 Hz 4'b0111 = 256 Hz 4'b1000 = 128 Hz 4'b1001 = 64 Hz 4'b1010 = 32 Hz 4'b1011 = 16 Hz 4'b1100 = 8 Hz 4'b1101 = 4 Hz 4'b1110 = 2 Hz 4'b1111 = 1 Hz	4'b0000
CLKRST	[3]	RW	RTC clock count reset 0 = Enables RTC counter (2^{15} clock divider) 1 = Resets and disables RTC counter NOTE: When it enables CTLEN, CLKRST affects RTC.	0
CNTSEL	[2]	RW	BCD count select 0 = Merges BCD counters 1 = Reserved (Separate BCD counters) NOTE: When it enables CTLEN, CNTSEL affects RTC.	0
CLKSEL	[1]	RW	BCD clock select 0 = XTAL $1/2^{15}$ divided clock 1 = Reserved (XTAL clock only for test) NOTE: When it enables CTLEN, CLKSEL affects RTC.	0
CTLEN	[0]	RW	Enables RTC control 0 = Disables RTC control 1 = Enables RTC control NOTE: When it enables CTLEN, you can change the BCD time count setting, 2^{15} clock divider reset, BCD counter select, and BCD clock select.	0

Name	Bit	Type	Description	Reset Value
			Note: RTC power consumption increases if CTLEN is 1. This bit should be 0 to achieve low power consumption if no more RTC control is required.	

To start RTC, set RTCCON[0] as 1.

NOTE:

1. The RTCCON register consists of 10 bits such as the CTLEN that controls the Read/Write enable of the BCD SEL, CNTSEL, CLKRST, TICCKSEL, TICEN for testing, and CLKOUTEN for RTC clock output control.
2. CTLEN bit controls all interfaces between the CPU and the RTC. Therefore, you should set it to "1" in an RTC control routine to enable data write after a system reset. To prevent inadvertent writing into BCD counter registers, it should clear the CTLEN bit to 0 before power off.
3. CLKRST is counter reset for 2^{15} clock divider. Before RTC clock setting, it should reset 2^{15} clock divider for exact RTC operation.

27.9.1.3 TICNT

- Base Address: 0x1007_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
TICK_TIME_ COUNT	[31:0]	RW	32-bit tick time count value. Tick timer is an up-counter. If the current tick count reaches this value, tick time interrupt occurs. NOTE: This value must be greater than 3.	32'b0

27.9.1.4 RTCALM

- Base Address: 0x1007_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0
ALMEN	[6]	RW	Enables Alarm global 0 = Disables alarm global 1 = Enables alarm global NOTE: For using ALARM_INT and ALARM_WK, set ALMEN as 1'b1.	0
YEAREN	[5]	RW	Enables Year alarm 0 = Disables year alarm 1 = Enables year alarm	0
MONEN	[4]	RW	Enables Month alarm 0 = Disables month alarm 1 = Enables month alarm	0
DAYEN	[3]	RW	Enables Day alarm 0 = Disables day alarm 1 = Enables day alarm	0
HOUREN	[2]	RW	Enables Hour alarm 0 = Disables hour alarm 1 = Enables hour alarm	0
MINEN	[1]	RW	Enables Minute alarm 0 = Disables minute alarm 1 = Enables minute alarm	0
SECEN	[0]	RW	Enables Second alarm 0 = Disables second alarm 1 = Enables second alarm	0

The RTCALM register determines the alarm enable and the alarm time. Note that, the RTCALM register generates the alarm signal through both ALARM_INT and ALARM_WK in power down mode, but only through ALARM_INT in the normal operation mode. Enable ALMEN to use ALARM_INT and ALARM_WK.

When compare value is year, then you should enable ALMEN and YEAREN. When compare values are year,

month, day, hour, minute and second, then you should enable ALMEN, YEAREN, MONEN, DAYEN, HOUREN, MINEN, and SECEN.

27.9.1.5 ALMSEC

- Base Address: 0x1007_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0
SECDATA	[6:4]	RW	BCD value for alarm second. 0 to 5	000
	[3:0]		0 to 9	0000

27.9.1.6 ALMMIN

- Base Address: 0x1007_0000
- Address = Base Address + 0x0058, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0
MINDATA	[6:4]	RW	BCD value for alarm minute. 0 to 5	000
	[3:0]		0 to 9	0000

27.9.1.7 ALMHOUR

- Base Address: 0x1007_0000
- Address = Base Address + 0x005C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	0
HOURDATA	[5:4]	RW	BCD value for alarm hour. 0 to 2	00
	[3:0]		0 to 9	0000

27.9.1.8 ALMDAY

- Base Address: 0x1007_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	0
DAYDATA	[5:4]	RW	BCD value for alarm day, from 0 to 28, 29, 30, 31. 0 to 3	00
	[3:0]		0 to 9	0000

27.9.1.9 ALMMON

- Base Address: 0x1007_0000
- Address = Base Address + 0x0064, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0
MONDATA	[4]	RW	BCD value for alarm month. 0 to 1	0
	[3:0]		0 to 9	0000

27.9.1.10 ALMYEAR

- Base Address: 0x1007_0000
- Address = Base Address + 0x0068, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
YEARDATA	[11:8]	RW	BCD value for alarm year. 0 to 9	0000
	[7:4]		0 to 9	0000
	[3:0]		0 to 9	0000

27.9.1.11 BCDSEC

- Base Address: 0x1007_0000
- Address = Base Address + 0x0070, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	–
SECDATA	[6:4]	RW	BCD value for second. 0 to 5	–
	[3:0]		0 to 9	–

27.9.1.12 BCDMIN

- Base Address: 0x1007_0000
- Address = Base Address + 0x0074, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	–
MINDATA	[6:4]	RW	BCD value for minute. 0 to 5	–
	[3:0]		0 to 9	–

27.9.1.13 BCDHOUR

- Base Address: 0x1007_0000
- Address = Base Address + 0x0078, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	–
HOURDATA	[5:4]	RW	BCD value for hour. 0 to 2	–
	[3:0]		0 to 9	–

27.9.1.14 BCDDAYWEEK

- Base Address: 0x1007_0000
- Address = Base Address + 0x007C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	–
DAYWEEKDATA	[2:0]	RW	BCD value for a day of the week. 1 to 7	–

27.9.1.15 BCDDAY

- Base Address: 0x1007_0000
- Address = Base Address + 0x0080, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	–
DAYDATA	[5:4]	RW	BCD value for day. 0 to 3	–
	[3:0]		0 to 9	–

27.9.1.16 BCDMON

- Base Address: 0x1007_0000
- Address = Base Address + 0x0084, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	–
MONDATA	[4]	RW	BCD value for month. 0 to 1	–
	[3:0]		0 to 9	–

27.9.1.17 BCDYEAR

- Base Address: 0x1007_0000
- Address = Base Address + 0x0088, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
YEARDATA	[11:8]	RW	BCD value for year. 0 to 9	–
	[7:4]		0 to 9	–
	[3:0]		0 to 9	–

27.9.1.18 CURTICCNT

- Base Address: 0x1007_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Tick counter observation	[31:0]	R	Current tick count value	–

28 Universal Asynchronous Receiver and Transmitter

28.1 Overview

A Universal Asynchronous Receiver and Transmitter (UART) in Exynos 4412 SCP provide four independent channels with asynchronous and serial input/output (I/O) ports for general purpose (Ch0 to 3). It also provides a dedicated channel for communication with Global Positioning System (GPS) (Ch4). All the ports operate either in an interrupt-based or a DMA-based mode. UART generates either an interrupt or a DMA request to transfer data to and from CPU and UART. UART supports bit rates up to 4 Mbps. Each UART channel contains two First In First Outs (FIFOs) to receive and transmit data as in:

- 256 bytes in Ch0
- 64 bytes in Ch1 and Ch4
- 16 bytes in Ch2 and Ch3

UART includes:

- Programmable Baud rates
- Infrared (IR) transmitter/receiver
- One or two stop bit insertion
- 5-bit, 6-bit, 7-bit, or 8-bit data width and parity checking

As shown in *Figure 28-1*, each UART contains:

- Baud-rate generator
- Transmitter
- Receiver
- Control unit

The Baud-rate generator uses SCLK_UART. The transmitter and the receiver contain FIFOs and data shifters. The data to be transmitted is written to Tx FIFO, and copied to the transmit shifter. The data is then shifted out by the transmit data pin (TxDn). The received data is shifted from the receive data pin (RxDn), and copied to Rx FIFO from the shifter.

28.2 Features

Features of UART are:

- RxD0, TxD0, RxD1, TxD1, RxD2, TxD2, RxD3, and TxD3 with either DMA-based or interrupt-based operation
- UART Ch 0, 1, 2, and 3 with IrDA 1.0
- UART Ch 0 with 256 byte FIFO, Ch 1 and 4 with 64 byte FIFO, Ch 2 and 3 with 16 byte FIFO
- UART Ch 0, 1, 2 with nRTS0, nCTS0, nRTS1, nCTS1, nCTS2, and nRTS2 for Auto Flow Control (AFC)
- UART Ch 4 communicates with GPS and it supports AFC
- UART supports handshakes transmit/receive.

28.3 UART Description

This section includes UART operations such as:

- Data transmission
- Data reception
- Interrupt generation
- Baud-rate generation
- Loop-back mode
- Infrared modes
- AFC

Figure 28-1 illustrates the block diagram of UART.

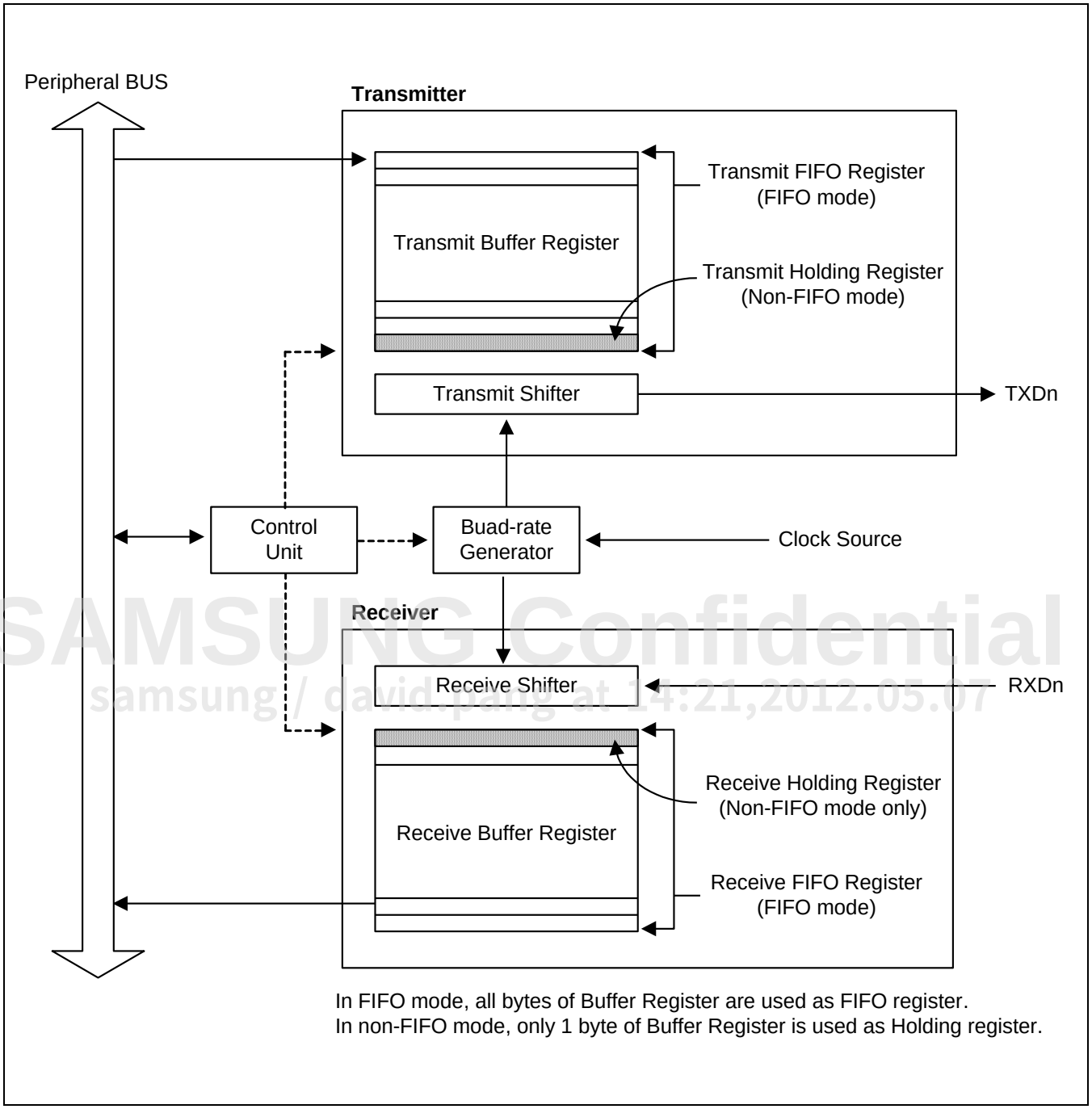


Figure 28-1 Block Diagram of UART

28.3.1 Data Transmission

The data frame for transmission is programmable. It consists of these bits that are specified by the line control register (ULCONn):

- A Start bit
- Five to eight data bits
- An optional parity bit
- One to two stop bits

The transmitter also produces a break condition that forces the serial output to logic 0 state for one-frame transmission time. This block transmits the break signals after it completely transmits the present transmission word. After the break signal transmission, the transmitter continuously transmits data to Tx FIFO (Tx holding register, in case of non-FIFO mode).

28.3.2 Data Reception

The data frame for reception is also programmable. It consists of a start bit, five to eight data bits, an optional parity bit, and one to two stop bits in the line control register (ULCONn). The receiver detects these errors and each of these errors sets an error flag:

- Overrun error: This error indicates that new data has overwritten the old data before the old data was read.
- Parity error: This error indicates that the receiver has detected an unexpected parity condition.
- Frame error: This error indicates that the received data does not have a valid stop bit.
- Break condition: This indicates that the RxDn input is held in the logic 0 state for more than one-frame transmission time.

Receive time-out condition occurs when the Rx FIFO is not empty in the FIFO mode and does not receive any data during the frame time specified in UCON.

28.3.3 AFC

UART0 and UART1 in Exynos 4412 SCP support AFC by using nRTS and nCTS signals.

To connect UART to a Modem, disable the AFC bit in UMCONn register and control the signal of nRTS by using software. The UART4 supports AFC, but it is dedicated for communication with GPS.

In AFC, the nRTS signal depends on the condition of the receiver, whereas the nCTS signals control the operation of transmitter. The transmitter of UART transfers the data to FIFO when nCTS signals are activated In AFC, nCTS signals means that other FIFO of UART is ready to receive data.

Before the UART receives data, nRTS has to be activated when Rx FIFO has a spare more than 2-byte and has to be inactivated when its receive FIFO has a spare under 1-byte. In AFC, the nRTS signals means that its RX FIFO is ready to receive data).

Figure 28-2 illustrates the UART AFC interface.

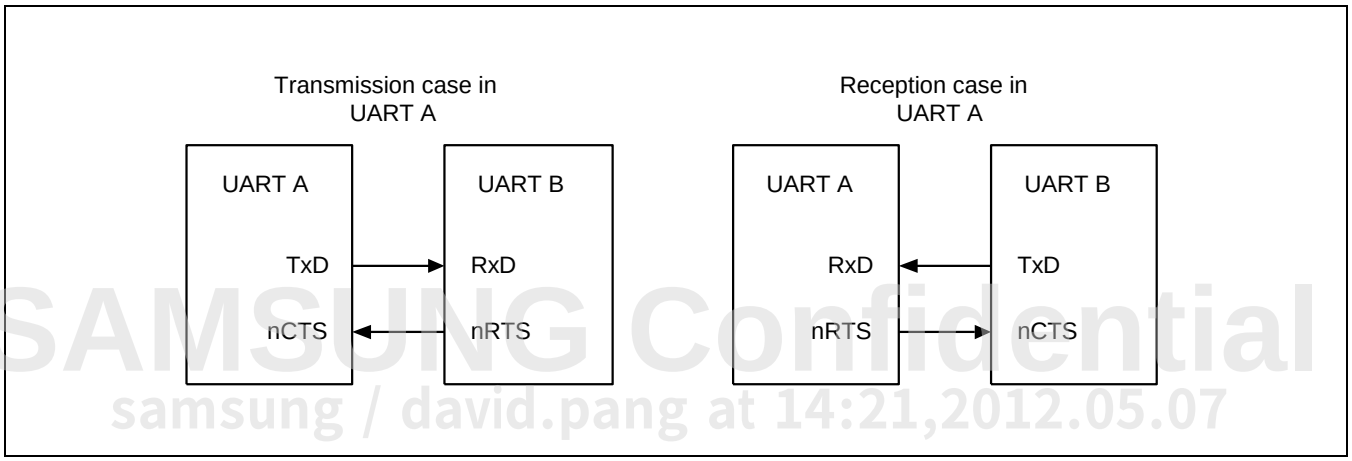


Figure 28-2 UART AFC Interface

28.3.4 Example of Non AFC (Controlling nRTS and nCTS by Software)

This section includes:

- Rx operation with FIFO
- Tx operation with FIFO

28.3.4.1 Rx Operation with FIFO

1. Select the transmit mode (either interrupt or DMA mode).
2. Verify the value of Rx FIFO count in the UFSTATn register. When the value is less than 16, set the value of UMCONn[0] to "1" (activate nRTS). However, when the value equal to or larger than 16, set the value to "0" (inactivate nRTS).
3. Repeat the Step 2 to receive next data.

28.3.4.2 Tx Operation with FIFO

1. Select the transmit mode (either interrupt or DMA mode).
2. Verify the value of UMSTATn[0]. When the value is "1" (activate nCTS), Write data to Tx FIFO register.
3. Repeat the Step 2 to send next data.

28.3.5 Trigger Level of Tx/Rx FIFO and DMA Burst Size in DMA Mode

DMA transaction starts when Tx/Rx data reaches the trigger level of Tx/Rx FIFO of UFCONn register in DMA mode. A single DMA transaction transfers a data whose size is specified as the DMA burst size of UCONn register. The DMA transactions are repeated until Tx/Rx FIFO count is less than the DMA burst size. Thus, DMA burst size should be less than or equal to the trigger level of Tx/Rx FIFO. In general ensure that the trigger level of Tx/Rx FIFO and DMA burst size matches.

28.3.6 RS-232C Interface

To connect UART to the modem interface (instead of null modem), nRTS, nCTS, nDSR, nDTR, DCD, and nRI signals are required. You can control these signals with general I/O ports by using software as the AFC does not support the RS-232C interface.

28.3.7 Interrupt/DMA Request Generation

Each UART in Exynos 4412 SCP consists of seven status (Tx/Rx/Error) signals, namely:

- Overrun error
- Parity error
- Frame error
- Break condition
- Receive buffer data ready
- Transmit buffer empty
- Transmit shifter empty

The corresponding UART status register (UTRSTATn/UERSTATn) indicates these conditions.

The overrun error, parity error, frame error, and break condition specify the receive error status.

When you set receive-error-status-interrupt-enable bit to 1 in the control register (UCONn), the receive error status generates receive-error-status-interrupt.

When a receive-error-status-interrupt-request is detected, you can identify the source of interrupt by reading the value of UERSTATn.

When you set Receive mode in control register (UCONn) as interrupt request or polling mode, Rx interrupt is generated in this case. When you set Receive mode in control register (UCONn) as interrupt request or polling mode, Rx interrupt is generated in this case. When the receiver transfers data of the receive shifter to the receive FIFO register in FIFO mode, and the number of received data is greater than or equal to the trigger level of Rx FIFO.

In non-FIFO mode, transferring the data of receive shifter to receive holding register causes Rx interrupt in the interrupt request and polling modes.

When the transmitter transfers data from its transmit FIFO register to transmit shifter and the number of data left in transmit FIFO is less than or equal to the trigger level of Tx FIFO, Tx interrupt is generated This occurs when Transmit mode in control register is selected as Interrupt request or polling mode.

In non-FIFO mode, transferring the data from transmit holding register to transmit shifter causes Tx interrupt in the interrupt request and polling mode.

Remember that the Tx interrupt is always requested when the number of data in the transmit FIFO is smaller than the trigger level. This means that an interrupt is requested as soon as you enable the Tx interrupt, unless you fill the Tx buffer. Fill the Tx buffer first and then enable the Tx interrupt.

The interrupt controllers of Exynos 4412 SCP are of the level-triggered type. Set the interrupt type as "Level" when you program the UART control registers.

When you select Receive and Transmit modes in control register as DMA request mode, DMAn request occurs instead of Rx or Tx interrupt in the above situation.

Table 28-1 describes the interrupts in connection with FIFO.

Table 28-1 Interrupts in Connection with FIFO

Type	FIFO Mode	Non-FIFO Mode
Rx interrupt	Generated when Rx FIFO count is greater than or equal to the trigger level of received FIFO. Generated when the number of data in FIFO does not reach the trigger level of Rx FIFO and does not receive any data during the specified time (receive time out) in UCON.	Generated by receive holding register whenever receive buffer becomes full.
Tx interrupt	Generated when Tx FIFO count is less than or equal to the trigger level of transmit FIFO (trigger level of Tx FIFO).	Generated by transmit holding register whenever transmit buffer becomes empty.
Error interrupt	Generated if frame error, parity error, or break signal are detected. Generated if UART receives new data when Rx FIFO is full (overrun error).	Generated by all errors. However when another error occurs at the same time, only one interrupt is generated.

28.3.8 UART Error Status FIFO

UART contains the error status FIFO besides the Rx FIFO register. The error status FIFO indicates which data among FIFO registers receives an error. An error interrupt is issued only when the data that contains an error is ready to read out. To clear the error status FIFO, you must read out URXHn with an error and UERSTATn.

For example, assume that the UART Rx FIFO receives A, B, C, D, and E characters sequentially and the frame error occurs while receiving "B" and the parity error occurs while receiving "D".

The actual UART receive error does not generate any error interrupt, since it does not read out the character received with an error. The error interrupt occurs if the character is read out.

Time	Sequence Flow	Error Interrupt	Note
#0	When no character is read out	–	–
#1	Receives A, B, C, D, and E	–	–
#2	After CPU reads out A	Frame error (in B) interrupt occurs.	"B" has to be read out.
#3	After CPU reads out B	–	–
#4	After CPU reads out C	Parity error (in D) interrupt occurs.	"D" has to be read out.
#5	After CPU reads out D	–	–
#6	After CPU reads out E	–	–

Figure 28-3 illustrates that UART receives the five characters including two errors.

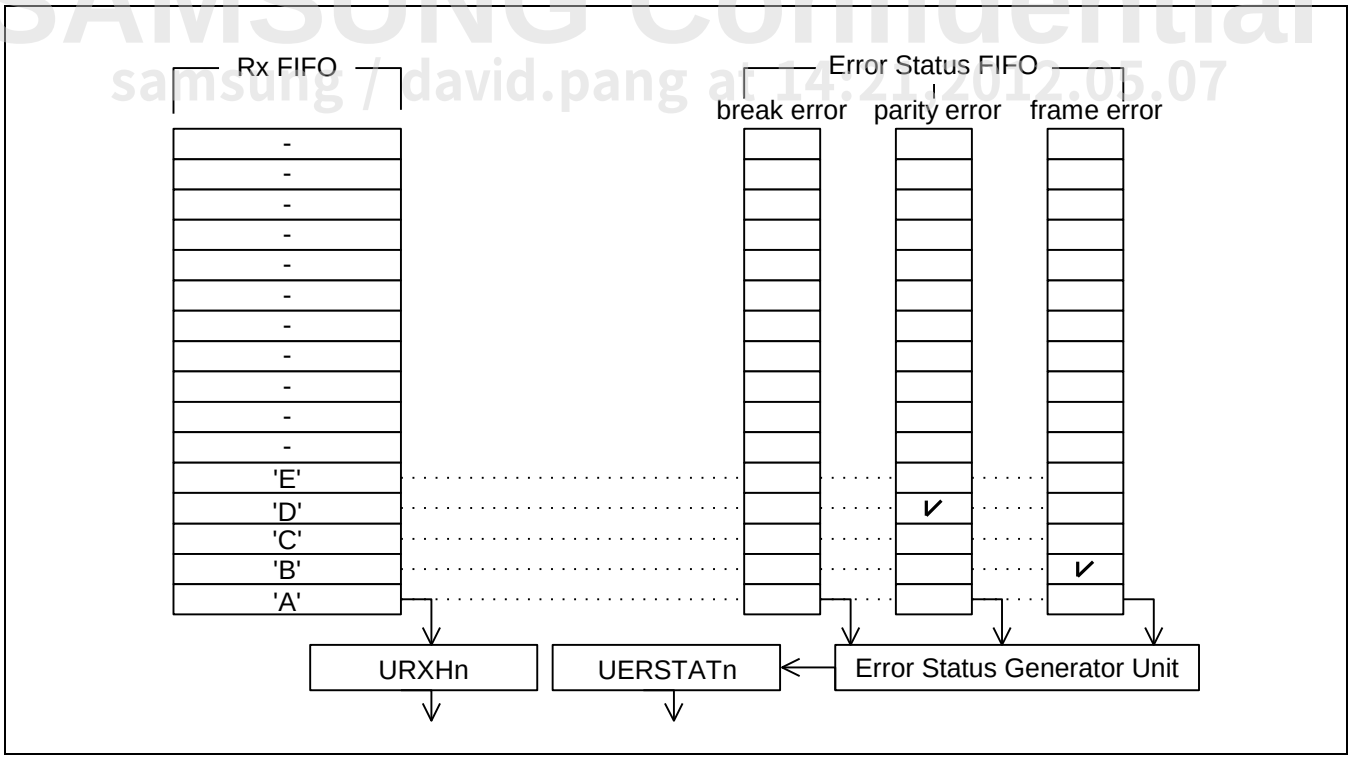


Figure 28-3 UART Receives the Five Characters Including Two Errors

28.3.8.1 Infra-Red Mode

The Exynos 4412 SCP UART block supports both infra-red (IR) transmission and reception. You can select the IR mode by setting the IR-mode bit in the UART line control register (ULCONn). *Figure 28-4* illustrates how to implement the IR mode.

In IR transmit mode, the transmit pulse moves at the rate of 3/16, that is, normal serial transmit rate when the transmit data bit is set to 0. However, in IR receive mode, the receiver must detect the 3/16 pulsed period to recognize a 0 value Refer to frame timing diagrams shown in *Figure 28-5* and *Figure 28-7* for more information.

Figure 28-4 illustrates IrDA function block diagram.

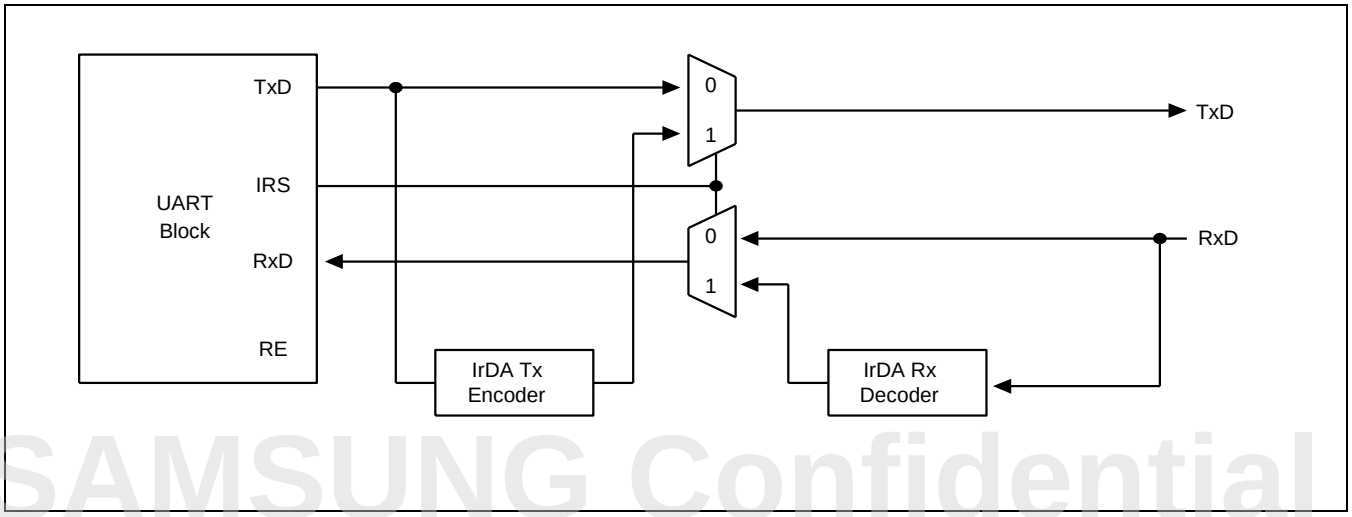


Figure 28-4 IrDA Function Block Diagram

Figure 28-5 illustrates the serial I/O frame timing diagram (Normal UART)

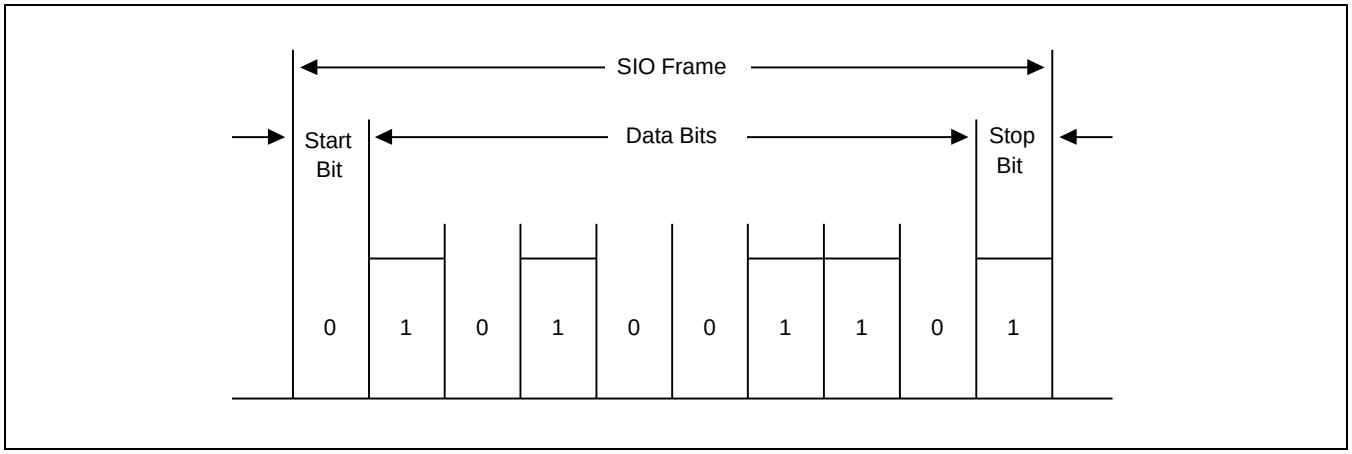


Figure 28-5 Serial I/O Frame Timing Diagram (Normal UART)

Figure 28-6 illustrates the infra-red transmit mode frame timing diagram.

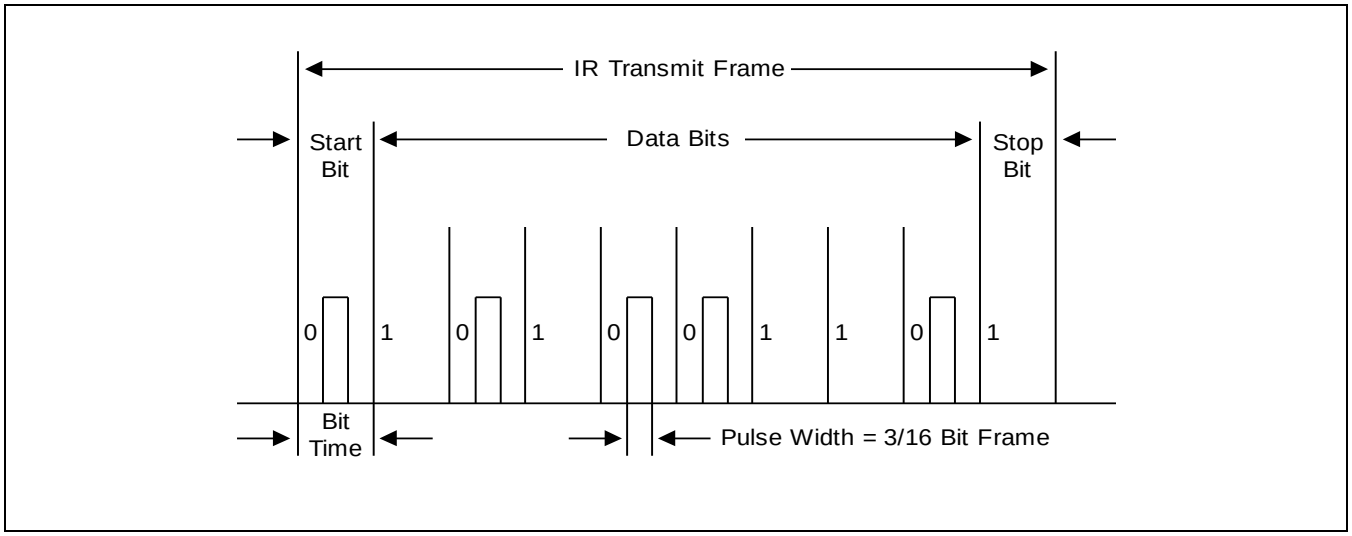


Figure 28-6 Infra-Red Transmit Mode Frame Timing Diagram

Figure 28-7 illustrates the infra-red receive mode frame timing diagram.

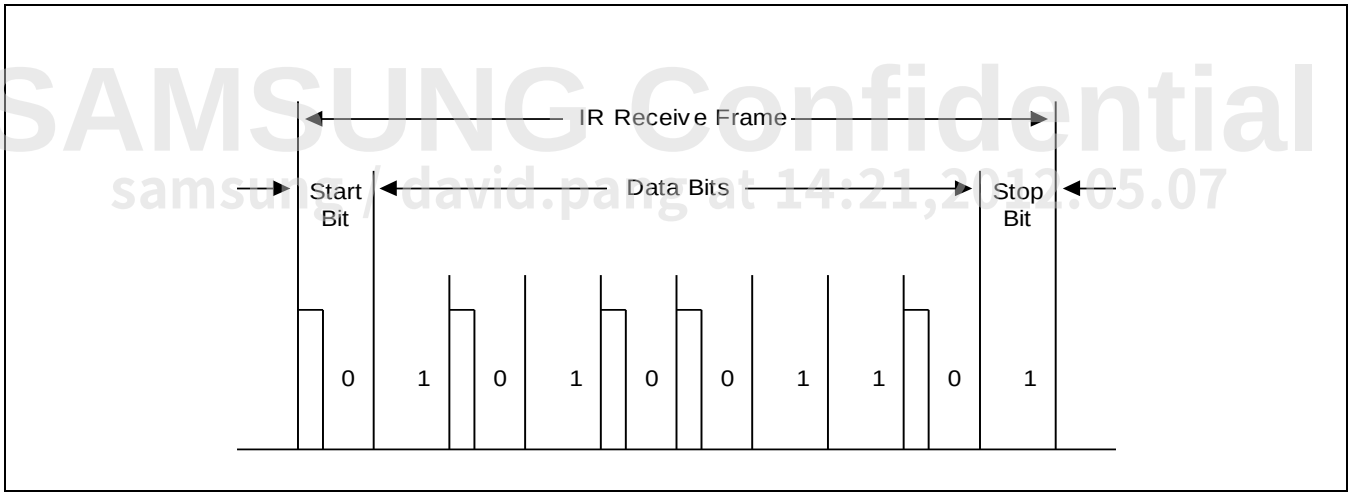


Figure 28-7 Infra-Red Receive Mode Frame Timing Diagram

28.4 UART Input Clock Description

Figure 28-8 illustrates the input clock diagram for UART.

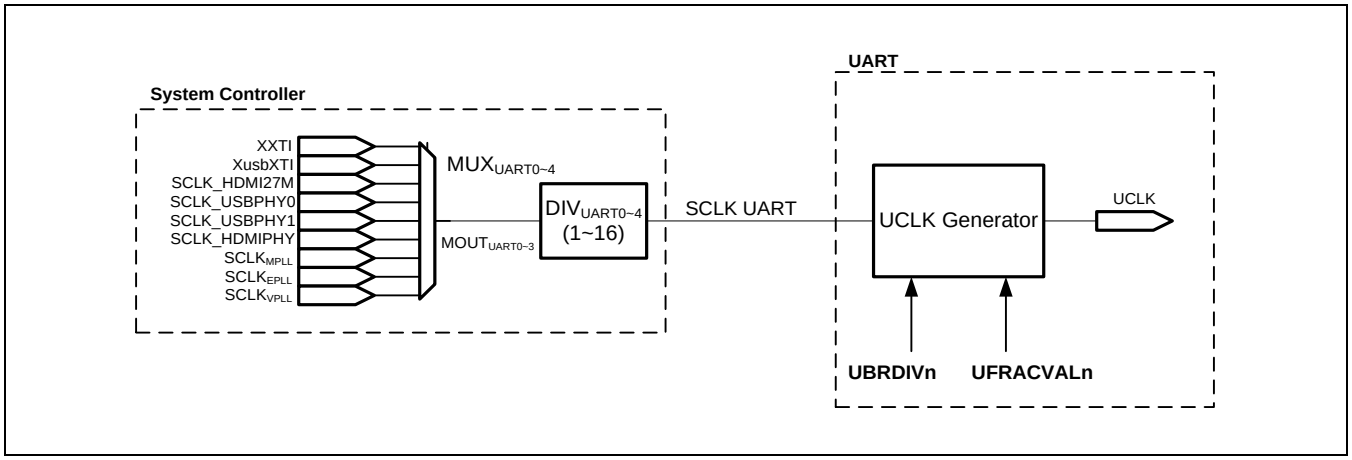


Figure 28-8 Input Clock Diagram for UART

Exynos 4412 SCP provides UART with a variety of clocks. *Figure 28-8* illustrates that UART uses SCLK_UART clock, which is from clock controller. You can also select SCLK_UART from various clock sources. Refer to Chapter 7, Clock Controller, for more information.

28.5 I/O Description

Signal	I/O	Description	Pad	Type
UART_0_RXD	Input	Receives data for UART0	XuRXD_0	muxed
UART_0_TXD	Output	Transmits data for UART0	XuTXD_0	muxed
UART_0_CTSn	Input	Clears to send (active low) for UART0	XuCTSn_0	muxed
UART_0_RTSn	Output	Requests to send (active low) for UART0	XuRTSn_0	muxed
UART_1_RXD	Input	Receives data for UART1	XuRXD_1	muxed
UART_1_TXD	Output	Transmits data for UART1	XuTXD_1	muxed
UART_1_CTSn	Input	Clears to send (active low) for UART1	XuCTSn_1	muxed
UART_1_RTSn	Output	Requests to send (active low) for UART1	XuRTSn_1	muxed
UART_2_RXD	Input	Receives data for UART2	XuRXD_2	muxed
UART_2_TXD	Output	Transmits data for UART2	XuTXD_2	muxed
UART_2_CTSn	Input	Clears to send (active low) for UART2	XuCTSn_2	muxed
UART_2_RTSn	Output	Requests to send (active low) for UART2	XuRTSn_2	muxed
UART_3_RXD	Input	Receives data for UART3	XuRXD_3	muxed
UART_3_TXD	Output	Transmits data for UART3	XuTXD_3	muxed

NOTE:

1. Type filed indicates whether the signal connects to the dedicated pad or muliplexed signal pad s. UART shares external pads with IrDA. To use these pads, set GPIO before the start of UART Refer to Chapter 6, GPIO, for more information.
2. UART4 has no I/O ports. It communicates with the internal GPS module.

28.6 Register Description

28.6.1 Register Map Summary

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000

Register	Offset	Description	Reset Value
ULCONn	0x0000	Specifies line control	0x0000_0000
UCONn	0x0004	Specifies control	0x0000_3000
UFCONn	0x0008	Specifies FIFO control	0x0000_0000
UMCONn	0x000C	Specifies modem control	0x0000_0000
UTRSTATn	0x0010	Specifies Tx/Rx status	0x0000_0006
UERSTATn	0x0014	Specifies Rx error status	0x0000_0000
UFSTATn	0x0018	Specifies FIFO status	0x0000_0000
UMSTATn	0x001C	Specifies modem status	0x0000_0000
UTXHn	0x0020	Specifies transmit buffer	Undefined
URXHn	0x0024	Specifies receive buffer	0x0000_0000
UBRDIVn	0x0028	Specifies baud rate divisor	0x0000_0000
UFRACVALn	0x002C	Specifies divisor fractional value	0x0000_0000
UINTPn	0x0030	Specifies interrupt pending	0x0000_0000
UINTSPn	0x0034	Specifies interrupt source pending	0x0000_0000
UINTMn	0x0038	Specifies interrupt mask	0x0000_0000

28.6.1.1 ULCONn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0
Infrared Mode	[6]	RW	Determines whether to use the infra-red mode. 0 = Normal mode operation 1 = Infra-red Tx/Rx mode	0
Parity Mode	[5:3]	RW	Specifies the type of parity that UART generates and checks during UART transmit and receive operation. 0xx = No parity 100 = Odd parity 101 = Even parity 110 = Parity forced/ checked as 1 111 = Parity forced/ checked as 0	
Number of Stop Bit	[2]	RW	Specifies how many stop bits UART uses to signal end-of-frame signal. 0 = One stop bit per frame 1 = Two stop bit per frame	0
Word Length	[1:0]	RW	Indicates the number of data bits UART transmits or receives per frame. 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits	0

28.6.1.2 UCONn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0
RSVD	[23]	WO	Reserved	0
Tx DMA Burst Size	[22:20]	RW	Tx DMA Burst Size It is the data transfer size of one DMA transaction. Tx DMA request triggers the DMA transaction. You must program the DMA program to transfer the same data size as this is the value for a single Tx DMA request. 000 = 1 byte (Single) 001 = 4 bytes 010 = 8 bytes 011 = 16 bytes 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved	0
RSVD	[19]	WO	Reserved	0
Rx DMA Burst Size	[18:16]	RW	Rx DMA Burst Size It is the data transfer size of one DMA transaction. Rx DMA request triggers the DMA transaction. You must program the DMA program to transfer the same data size as this is the value for a single Rx DMA request. 000 = 1 byte (Single) 001 = 4 bytes 010 = 8 bytes 011 = 16 bytes 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved	0
Rx Timeout Interrupt Interval	[15:12]	RW	Rx Timeout Interrupt Interval Rx interrupt occurs if UART receives no data during 8 × (N + 1) frame time. The default value of this field is 3. It means that the timeout interval is 32 frame time.	0x3
Rx Time-out with empty Rx FIFO (4)	[11]	R/W	Enables Rx time-out feature when Rx FIFO counter is 0. This bit is valid only when UCONn[7] is 1. 0 = Disables Rx time-out feature when Rx FIFO is empty. 1 = Enables Rx time-out feature when Rx FIFO is empty.	0
Rx Time-out DMA suspend enable	[10]	R/W	Enables the suspension of Rx DMA FSM when Rx Time-out occurs. 0 = Disables suspension of Rx DMA FSM 1 = Enables suspension of Rx DMA FSM	0

Name	Bit	Type	Description	Reset Value
Tx Interrupt Type	[9]	RW	Interrupt request type. (2) 0 = Pulse (UART requests interrupt when the Tx buffer is empty in the non-FIFO mode or when it reaches the trigger level of Tx FIFO in the FIFO mode.) 1 = Level (Interrupt is requested when Tx buffer is empty in the non-FIFO mode or when it reaches the trigger level of Tx FIFO in the FIFO mode.)	0
Rx Interrupt Type	[8]	RW	Interrupt request type. (2) 0 = Pulse (UART requests interrupt when instant Rx buffer receives data in the non-FIFO mode or when it reaches the trigger level of Rx FIFO in the FIFO mode.) 1 = Level (UART requests interrupt when Rx buffer receives data in the non-FIFO mode or when it reaches the trigger level of Rx FIFO in the FIFO mode.)	0
Rx Time Out Enable	[7]	RW	Enables/disables Rx time-out interrupts when you enable UART FIFO. The interrupt is a receive interrupt. 0 = Disables 1 = Enables	0
Rx Error Status Interrupt Enable	[6]	RW	Enables the UART to generate an interrupt upon an exception, such as a break, frame error, parity error, or overrun error during a receive operation. 0 = Does not generate receive error status interrupt. 1 = Generates receive error status interrupt.	0
Loop-back Mode	[5]	RW	To set this bit to 1 triggers the UART to enter the loop-back mode. This mode is for test purposes only. 0 = Normal operation 1 = Loop-back mode	0
Send Break Signal	[4]	RWX	To set this bit to 1 triggers UART to send a break during 1 frame time. This bit is automatically cleared after sending the break signal. 0 = Normal transmit 1 = Sends the break signal	0
Transmit Mode	[3:2]	RW	Determines which function is able to Write Tx data to the UART transmit buffer. 00 = Disables 01 = Interrupt request or polling mode 10 = DMA mode 11 = Reserved	00
Receive Mode	[1:0]	RW	Determines which function is able to Read data from UART receive buffer. 00 = Disables 01 = Interrupt request or polling mode 10 = DMA mode 11 = Reserved	00

NOTE:

1. DIV_VAL = UBRDIVn + UFRACVAL/16. Refer to *28.6.1.11 UBRDIVn* and *28.6.1.12 UFRACVALn* for more information.
2. Exynos 4412 SCP uses a level-triggered interrupt controller. Therefore, you must set these bits to 1 for every transfer.
3. If UART does not reach the trigger level of FIFO and does not receive data during the time specified at the "Rx Timeout Interrupt Interval" field in DMA receive mode with FIFO, UART generates the Rx interrupt (receive time out). Ensure to verify the FIFO status and read out the rest.
4. Both UCONn[11] and UCONn[7] should be set to 1 if you want to enable Rx time-out feature when Rx FIFO counter is set to 0.

28.6.1.3 UFCONn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved	0
Tx FIFO Trigger Level	[10:8]	RW	Determines the trigger level of Tx FIFO. When data count of Tx FIFO is less than or equal to the trigger level, Tx interrupt occurs. [Channel 0] 000 = 0 byte 001 = 32 bytes 010 = 64 bytes 011 = 96 bytes 100 = 128 bytes 101 = 160 bytes 110 = 192 bytes 111 = 224 bytes [Channel 1, 4] 000 = 0 byte 001 = 8 bytes 010 = 16 bytes 011 = 24 bytes 100 = 32 bytes 101 = 40 bytes 110 = 48 bytes 111 = 56 bytes [Channel 2, 3] 000 = 0 byte 001 = 2 bytes 010 = 4 bytes 011 = 6 bytes 100 = 8 bytes 101 = 10 bytes 110 = 12 bytes 111 = 14 bytes	000
RSVD	[7]	–	Reserved	0
Rx FIFO Trigger Level	[6:4]	RW	Determines the trigger level of Rx FIFO. When data count of Rx FIFO is more than or equal to the trigger level, Rx interrupt occurs. [Channel 0] 000 = 32 byte 001 = 64 bytes 010 = 96 bytes 011 = 128 bytes 100 = 160 bytes 101 = 192 bytes 110 = 224 bytes 111 = 256 bytes	000

Name	Bit	Type	Description	Reset Value
			[Channel 1, 4] 000 = 8 byte 001 = 16 bytes 010 = 24 bytes 011 = 32 bytes 100 = 40 bytes 101 = 48 bytes 110 = 56 bytes 111 = 64 bytes [Channel 2, 3] 000 = 2 byte 001 = 4 bytes 010 = 6 bytes 011 = 8 bytes 100 = 10 bytes 101 = 12 bytes 110 = 14 bytes 111 = 16 bytes	
RSVD	[3]	–	Reserved	0
Tx FIFO Reset	[2]	S	Automatically clears after resetting FIFO 0 = Normal 1 = Tx FIFO reset	0
Rx FIFO Reset	[1]	S	Automatically clears after resetting FIFO 0 = Normal 1 = Rx FIFO reset	0
FIFO Enable	[0]	RW	0 = Disables 1 = Enables	0

NOTE: When UART does not reach the trigger level of FIFO, it does not receive data during the specified timeout interval in DMA receive mode with FIFO. It generates the Rx interrupt (receive time out). Ensure to verify the FIFO status and read out the rest.

28.6.1.4 UMCONn (n = 0, 1, 2, 4)

- Base Address = 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1384_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
RTS trigger Level	[7:5]	RW	Determines the trigger level of Rx FIFO to control nRTS signal. When it enables AFC bit and Rx FIFO have bytes that are greater than or equal to the trigger level, it deactivates nRTS signal. [Channel 0] 000 = 255 bytes 001 = 224 bytes 010 = 192 bytes 011 = 160 bytes 100 = 128 bytes 101 = 96 bytes 110 = 64 bytes 111 = 32 bytes [Channel 1, 4] 000 = 63 bytes 001 = 56 bytes 010 = 48 bytes 011 = 40 bytes 100 = 32 bytes 101 = 24 bytes 110 = 16 bytes 111 = 8 bytes [Channel 2] 000 = 15 bytes 001 = 14 bytes 010 = 12 bytes 011 = 10 bytes 100 = 8 bytes 101 = 6 bytes 110 = 4 bytes 111 = 2 bytes	0
Auto Flow Control (AFC)	[4]	RW	0 = Disables 1 = Enables	0
Modem Interrupt Enable	[3]	RW	0 = Disables 1 = Enables	0
RSVD	[2:1]	–	Reserved (These bits must be 0)	0
Request to Send	[0]	RW	If AFC bit is enabled, this value will be ignored. In this case, the Exynos 4412 SCP controls nRTS signals automatically. If AFC bit is disabled, the software must control nRTS signal. 0 = "H" level (inactivate nRTS) 1 = "L" level (activate nRTS)	0

NOTE:

1. UART 3 does not support AFC function because the Exynos 4412 SCP has no nRTS3 and nCTS3.
2. In AFC mode, set the trigger level of Rx FIFO lower than the trigger level of RTS because transmitter stops data transfer when it deactivates the nRST signal.

28.6.1.5 UTRSTATn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0
RX FIFO count in RX time-out status	[23:16]	R	Capture value of Rx FIFO counter when Rx time-out occurs	0x00
TX DMA FSM State	[15:12]	R	Current State of Tx DMA FSM 0x0 = IDLE 0x1 = Burst Request 0x2 = Burst Acknowledgement 0x3 = Burst Next (intermediate state for next request) 0x4 = Single Request 0x5 = Single Acknowledgement 0x6 = Single Next (intermediate state for next request) 0x7 = Last Burst Request 0x8 = Last Burst Acknowledgement 0x9 = Last Single Request 0x10 = Last Single Acknowledgement	0x0
RX DMA FSM State	[11:8]	R	Current State of Rx DMA FSM 0x0 = IDLE 0x1 = Burst Request 0x2 = Burst Acknowledgement 0x3 = Burst Next (intermediate state for next request) 0x4 = Single Request 0x5 = Single Acknowledgement 0x6 = Single Next (intermediate state for next request) 0x7 = Last Burst Request 0x8 = Last Burst Acknowledgement 0x9 = Last Single Request 0x10 = Last Single Acknowledgement	0x0
RSVD	[7:4]	–	Reserved	0
RX Time-out status/Clear[1]	[3]	RWX	RX Time-out status when read. 0 = Rx Time out did not occur 1 = Rx Time out. Clears Rx Time-out status when write. 0 = No operation 1 = Clears Rx Time-out status NOTE: When UCONn[10] is set to 1. writing 1 to this bit resumes Rx DMA FSM that was suspended when Rx time-out occurred.	0
Transmitter empty	[2]	R	This bit is automatically set to 1 when the transmit buffer has no valid data to transmit, and the transmit shift is empty. 0 = Not empty 1 = Transmitter (includes transmit buffer and shifter) empty	1

Name	Bit	Type	Description	Reset Value
Transmit buffer empty	[1]	R	This bit is automatically set to 1 when transmit buffer is empty. 0 = Buffer is not empty 1 = Buffer is empty (in non-FIFO mode, it requests interrupt or DMA). In FIFO mode, it requests interrupt or DMA, when the trigger level of Tx FIFO is set to 00 (Empty). When UART uses FIFO, check for Tx FIFO Count bits and Tx FIFO Full bit in UFSTAT instead of this bit.	1
Receive buffer data ready	[0]	R	It automatically sets this bit to 1 when receive buffer contains valid data, which is received over the RXDn port. 0 = Buffer is empty 1 = Buffer has a received data (In Non-FIFO mode, it requests interrupt or DMA) When UART uses the FIFO, check for Rx FIFO Count bits and Rx FIFO Full bit in UFSTAT instead of this bit.	0

28.6.1.6 UERSTATn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0
Break Detect	[3]	R	It automatically sets this bit to 1 to indicate that a break signal has been received. 0 = No break signal is received 1 = Break signal is received (Interrupt is requested.)	0
Frame Error	[2]	R	It automatically sets this bit to 1 when a frame error occurs during the receive operation. 0 = No frame error occurs during the receive operation 1 = Frame error occurs (Interrupt is requested.) during the receive operation	0
Parity Error	[1]	R	It automatically sets this bit to 1 when a parity error occurs during the receive operation. 0 = No parity error occurs during receive the receive operation 1 = Parity error occurs (Interrupt is requested.) during the receive operation	0
Overrun Error	[0]	R	It automatically sets this bit to 1 automatically if an overrun error occurs during the receive operation. 0 = No overrun error occurs during the receive operation 1 = Overrun error occurs (Interrupt is requested.) during the receive operation	0

NOTE: It clears these bits (UERSATn[3:0]) to 0 when UART error status is Read.

28.6.1.7 UFSTATn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0
Tx FIFO Full	[24]	R	It automatically sets this bit to 1 when the transmitted FIFO is full during transmit operation 0 = Not full 1 = Full	0
Tx FIFO Count	[23:16]	R	Number of data in Tx FIFO NOTE: This field is set to 0 when Tx FIFO is full.	0
RSVD	[15:10]	–	Reserved	0
Rx FIFO Error	[9]	R	This bit is set to 1 when Rx FIFO contains invalid data that results from frame error, parity error, or break signal.	0
Rx FIFO Full	[8]	R	It automatically sets this bit to 1 when the received FIFO is full during receive operation 0 = Not full 1 = Full	0
Rx FIFO Count	[7:0]	R	Number of data in Rx FIFO NOTE: This field is set to 0 when Rx FIFO is full.	0

28.6.1.8 UMSTATn (n = 0, 1, 2, 4)

- Base Address = 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1384_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0
Delta CTS	[4]	R	This bit indicates that the nCTS input to the Exynos 4412 SCP has changed its state since the last time CPU read it. Refer to *Figure 28-9* for more information. 0 = Has not changed 1 = Has changed NOTE: In UMSTAT4, reset value of this bit is undefined. It depends on the GPIO configuration of GPS.	0
RSVD	[3:1]	–	Reserved	-
Clear to Send	[0]	R	0 = Does not activates CTS signal (nCTS pin is high) 1 = Activates CTS signal (nCTS pin is low) NOTE: In UMSTAT4, reset value of this bit is undefined. It depends on the GPIO configuration of GPS.	0

Figure 28-9 illustrates the nCTS and delta Clear to Send (CTS) timing diagram.

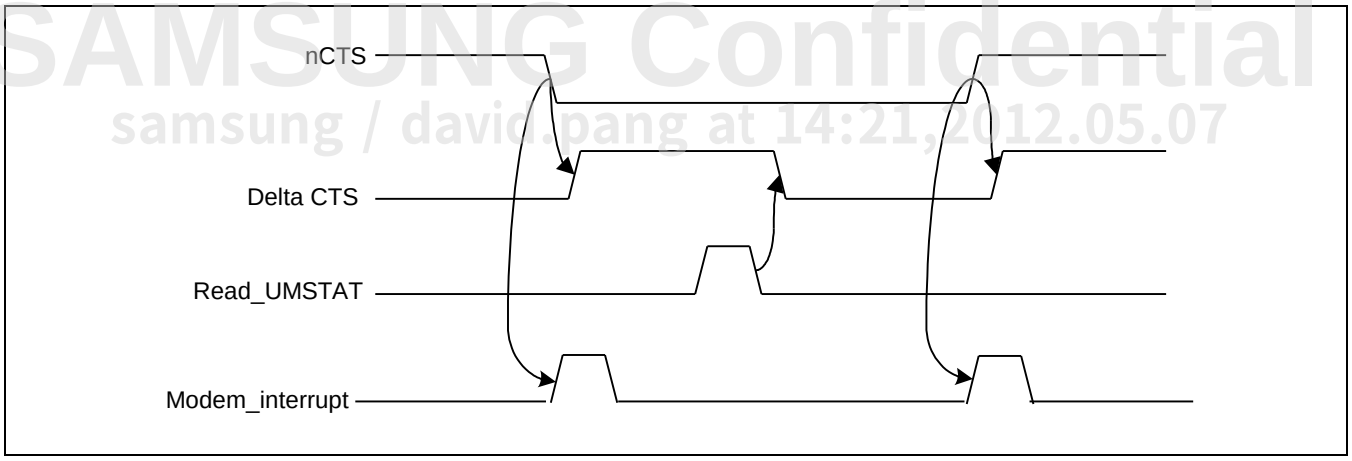


Figure 28-9 nCTS and Delta CTS Timing Diagram

28.6.1.9 UTXHn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
UTXHn	[7:0]	RWX	Transmits data for UARTn	–

28.6.1.10 URXHn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
URXHn	[7:0]	R	Receives data for UARTn	0x00

NOTE: When an overrun error occurs, CPU must Read URXHn. If not, the next received data makes an overrun error, even though it clears the overrun bit of UERSTATn.

28.6.1.11 UBRDIVn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
UBRDIVn	[15:0]	RW	Baud-rate division value	0x0000

NOTE: When UBRDIV value is set to 0, UFRACVAL value does not affect UART Baud rate.

28.6.1.12 UFRACVALn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0
UFRACVALn	[3:0]	RW	Determines the fractional part of Baud-rate divisor.	0x0

1. UART Baud-Rate Configuration

You can use the value stored in the Baud-rate divisor (UBRDIVn) and divisor fractional value (UFRACVALn) to determine the serial Tx/Rx clock rate (Baud rate) as:

DIV_VAL = UBRDIVn + UFRACVALn/16

or

DIV_VAL = (SCLK_UART/(bps × 16)) – 1

Where, the divisor should be from 1 to (216 – 1).

By using UFRACVALn, you can generate the Baud rate more accurately.

For example, if the Baud rate is 115200 bps and SCLK_UART is 40 MHz, UBRDIVn and UFRACVALn are:

DIV_VAL = (40000000/(115200 × 16)) – 1

= 21.7 – 1

= 20.7

UBRDIVn = 20 (integer part of DIV_VAL)

UFRACVALn/16 = 0.7

Therefore, UFRACVALn = 11

2. Baud-Rate Error Tolerance

UART Frame error should be less than 1.87 % (3/160)

tUPCLK = (UBRDIVn + 1 + UFRACVAL/16) × 16 × 1Frame/SCLK_UART

tUPCLK = Real UART Clock

tEXTUARTCLK = 1Frame/baud-rate

tEXTUARTCLK = Ideal UART Clock

UART error = (tUPCLK – tEXTUARTCLK)/tEXTUARTCLK × 100 %

* 1Frame = start bit + data bit + parity bit + stop bit.

3. UART Clock and PCLK Relation

There is a constraint on the ratio of clock frequencies for PCLK to UARTCLK.

The frequency of UARTCLK must be no more than 5.5/3 times faster than the frequency of PCLK:

FUARTCLK ⇐ 5.5/3 × FPCLK

FUARTCLK = baudrate × 16

This allows sufficient time to Write the received data to the receive FIFO.

28.6.1.13 UINTPn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0
MODEM	[3]	S	Generates modem interrupt.	0
TXD	[2]	S	Generates transmit interrupt.	0
ERROR	[1]	S	Generates error interrupt.	0
RXD	[0]	S	Generates receive interrupt.	0

Interrupt pending contains the information of the generated interrupts.

If one of the 4 bits is logical high ("1"), each UART channel generates interrupt.

NOTE: You must clear this in the interrupt service routine after clearing interrupt pending in Interrupt Controller (INTC). Clear specific bits of UINTP by writing 1's to the bits that you want to clear.

28.6.1.14 UINTSPn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0
MODEM	[3]	S	Generates modem interrupt.	0
TXD	[2]	S	Generates transmit interrupt.	0
ERROR	[1]	S	Generates error interrupt.	0
RXD	[0]	S	Generates receive interrupt.	0

NOTE: Interrupt Source Pending contains the information of the generated interrupt regardless of the value of Interrupt Mask.

28.6.1.15 UINTMn (n = 0 to 4)

- Base Address: 0x1380_0000, 0x1381_0000, 0x1382_0000, 0x1383_0000, 0x1384_0000
- Address = Base Address + 0x0038, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0
MODEM	[3]	RW	Masks modem interrupt.	0
TXD	[2]	RW	Masks transmit interrupt.	0
ERROR	[1]	RW	Masks error interrupt.	0
RXD	[0]	RW	Masks receive interrupt.	0

Figure 28-10 illustrates the block diagram of UINTSP, UINTP, and UINTM.

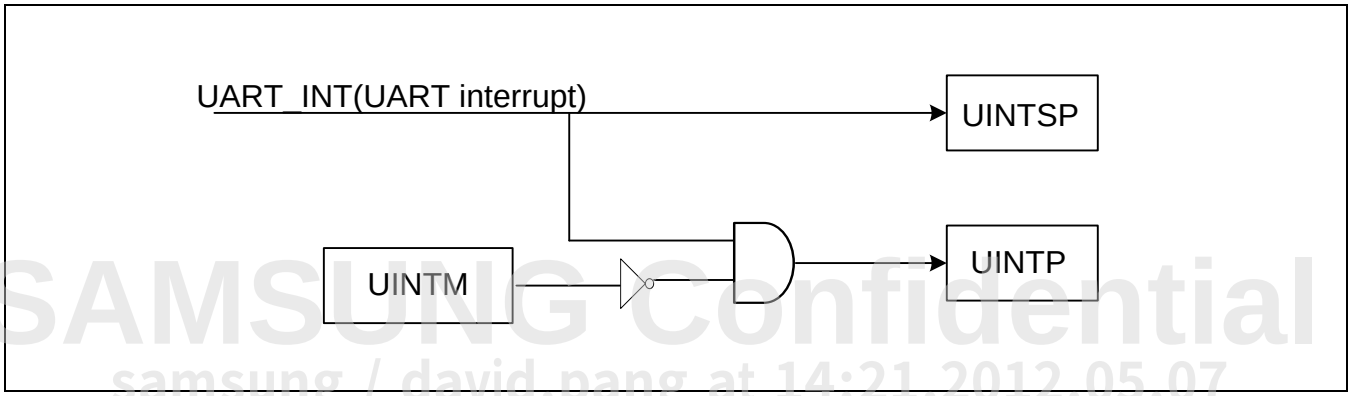


Figure 28-10 Block diagram of UINTSP, UINTP, and UINTM

Interrupt mask contains the information about masked interrupt sources. When a specific bit is set to 1, UART does not generate interrupt request signal to the Interrupt Controller even though it generates corresponding interrupt.

NOTE: In such cases, the corresponding bit of UINTSPn is set to 1. When the mask bit is set to 0, CPU services the interrupt requests from the corresponding interrupt source.

29 Inter-Integrated Circuit

29.1 Overview

The Exynos 4412 SCP Reduced Instruction Set Computer (RISC) microprocessor supports four multi-master Inter-Integrated Circuit (I2C) bus serial interfaces. To transmit information between bus masters and peripheral devices, which are connected to the I2C bus, a dedicated Serial Data Line (SDA) and Serial Clock Line (SCL) is used. Both SDA and SCL lines are bi-directional.

In multi-master I2C-bus mode, multiple Exynos 4412 SCP RISC microprocessors either receive or transmit serial data to or from slave devices. The master Exynos 4412 SCP initiates and terminates a data transfer over the I2C bus. The I2C bus in the Exynos 4412 SCP uses a standard I2C bus arbitration procedure to realize multi-master and multi-slave transfer.

To control multi-master I2C-bus operations, you must write values to these registers:

- Multi-master I2C-bus control register – I2CCON
- Multi-master I2C-bus control/status register – I2CSTAT
- Multi-master I2C-bus Tx/Rx data shift register – I2CDS
- Multi-master I2C-bus address register – I2CADD

If the I2C-bus is idle, both SDA and SCL lines should be at High level. A High-to-Low transition of SDA initiates a Start condition. A Low-to-High transition of SDA initiates a Stop condition, while SCL remains steady at High level.

The master device always generates Start and Stop conditions. Front 7 bits address value in the data byte is transferred through SDA line after the start condition has been initiated. This address value determines the slave device which the bus master device has selected. The 8th bit determines the direction of the transfer (Read or Write).

Every data byte put on the SDA line should be 8 bits in total. There is no limit either to send or receive bytes during the bus transfer operation. I2C master and slave devices always send the data from the Most Significant Bit (MSB) first, and then acknowledge (ACK) bit immediately follows every byte.

29.2 Features

Features of I2C bus interface are:

- 9 channels multi-master, Slave I2C bus interfaces
  (8 channels for general purpose, 1 channel dedicated for High Definition Multimedia Interface (HDMI).)
- 7-bit addressing mode
- Serial, 8-bit oriented, and bi-directional data transfer
- Supports up to 100 kbit/s in the Standard mode
- Supports up to 400 kbit/s in the Fast mode.
- Supports master transmit, master receive, slave transmit, and slave receive operation
- Supports interrupt or polling events

29.3 Block Diagram

Figure 29-1 illustrates the block diagram of I2C bus.

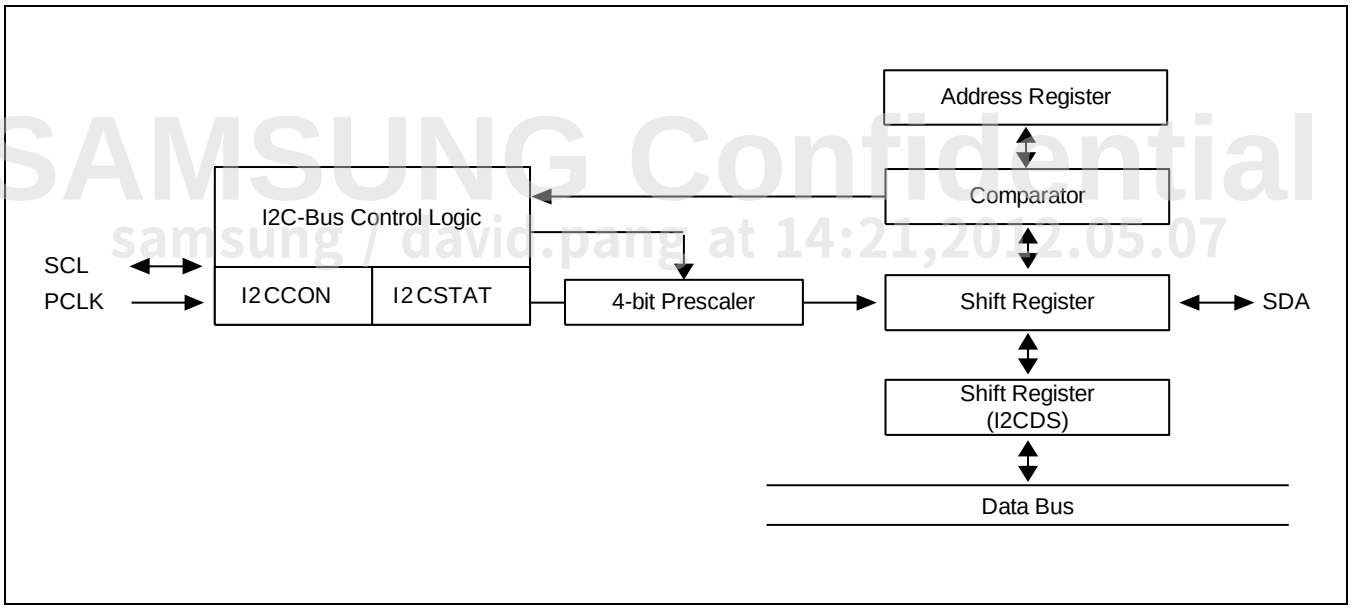


Figure 29-1 I2C-Bus Block Diagram

29.4 I2C-Bus Interface Operation

The four operation modes of the Exynos 4412 SCP I2C-bus interface are:

- Master Transmitter Mode:
- Master Receive Mode
- Slave Transmitter Mode
- Slave Receive Mode

The functional relationships among these operating modes are described in these sections:

- Start and Stop conditions
- Data transfer format
- ACK signal transmission
- Read-Write operation
- Bus arbitration procedures
- Abort conditions
- Configuring IIC-bus

29.4.1 Start and Stop Conditions

When the I2C-bus interface is inactive, it is usually in Slave mode. Alternatively, the interface should be in Slave mode before detecting a Start condition on the SDA line (a Start condition is initiated with a High-to-Low transition of the SDA line, when the clock signal of SCL is High). When controller changes the interface state to master-mode, SDA line initiates data transfer and generates SCL signal.

A Start condition transfers 1-byte serial data through SDA line, and a Stop condition terminates the data transfer. A Stop condition is a Low-to-High transition of the SDA line, while SCL is High. The master generates Start and Stop conditions. I2C bus goes into the busy state when a master or slave device generates a start condition. Alternatively, a Stop condition makes the I2C bus idle state.

When a master initiates a Start condition, it should send a slave address to notify the slave device. 1 byte of address field includes a 7-bit address and 1-bit transfer direction indicator, which shows Write or Read. When bit 8 is 0, it indicates a Write operation (Transmit Operation); when bit 8 is 1, it indicates a request for data Read (Receive Operation).

The master transmits Stop condition to complete the transfer operation. If the master wants to continue the data transmission to the bus, it should generate another Start condition and a slave address. In this manner, there can be various formats of the read-write operation.

Figure 29-2 illustrates the Start and Stop condition.

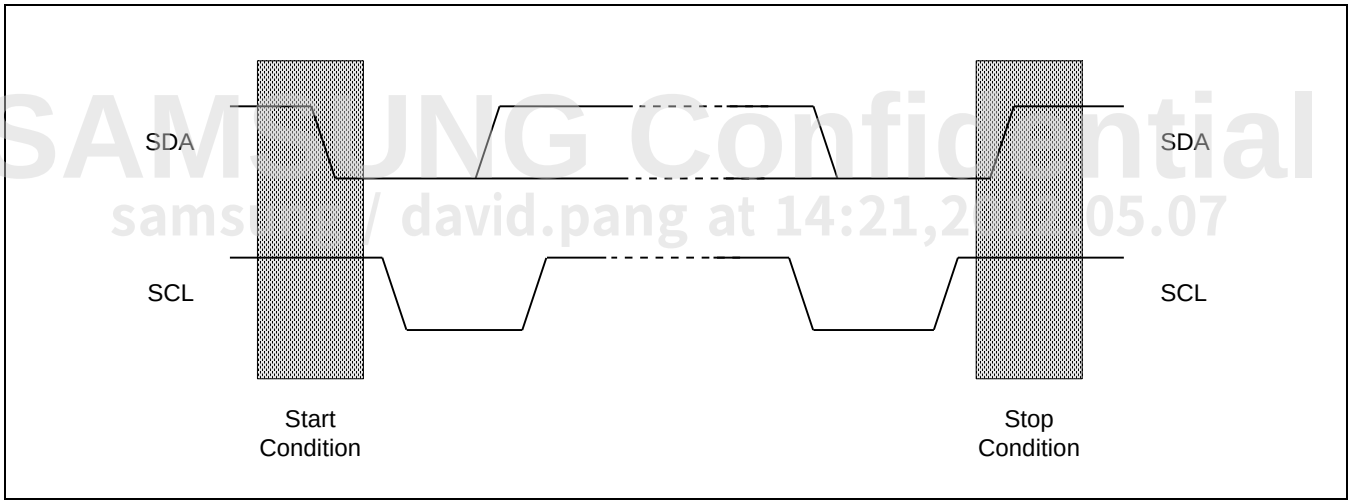


Figure 29-2 Start and Stop Condition

29.4.2 Data Transfer Format

Every byte placed on the SDA line should be 8 bits in length. There is no limit to transmit bytes per transfer. The first byte that follows a Start condition should have the address field. When the I2C-bus is operating in master mode, master transmits the address field. An ACK bit follows each byte. The I2C controller sends first the MSB of the data and address byte to the SDA line.

Figure 29-3 illustrates the I2C-bus interface data format.

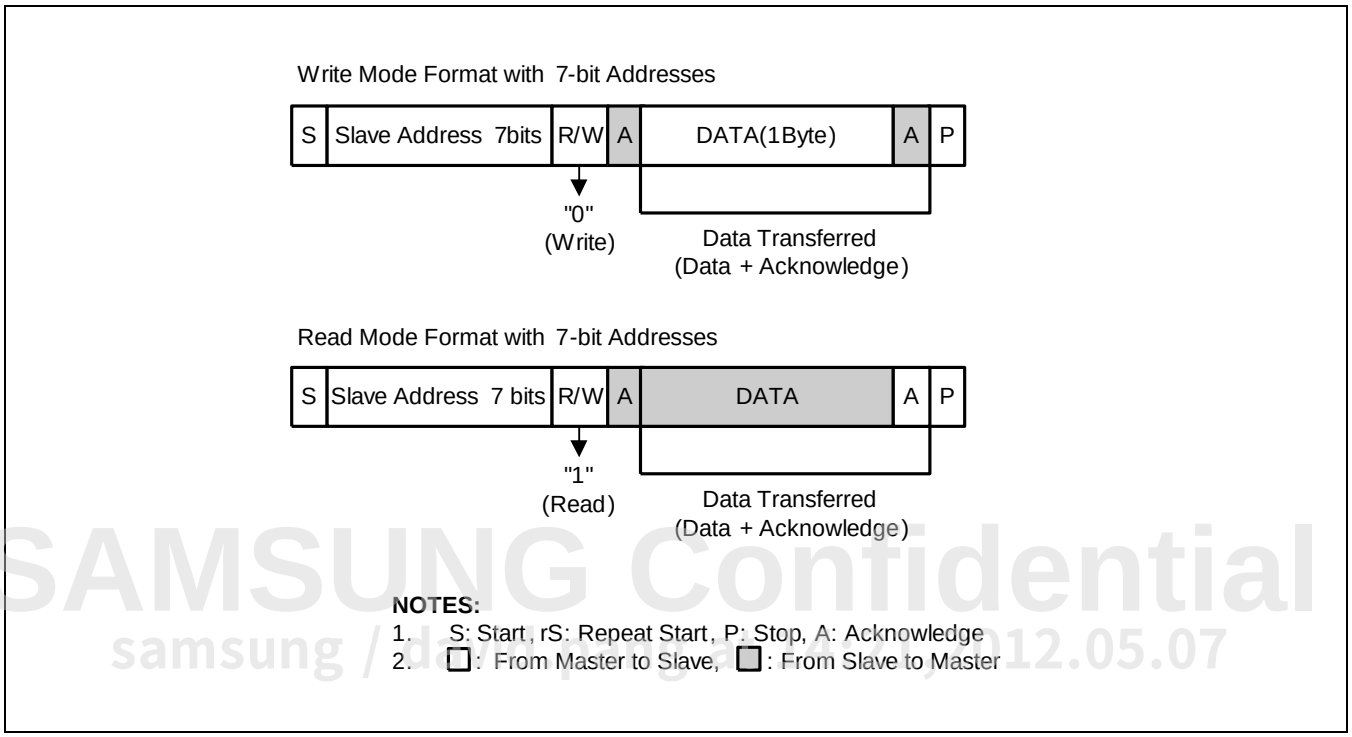


Figure 29-3 I2C-Bus Interface Data Format

Figure 29-4 illustrates the data transfer on the I2C-bus.

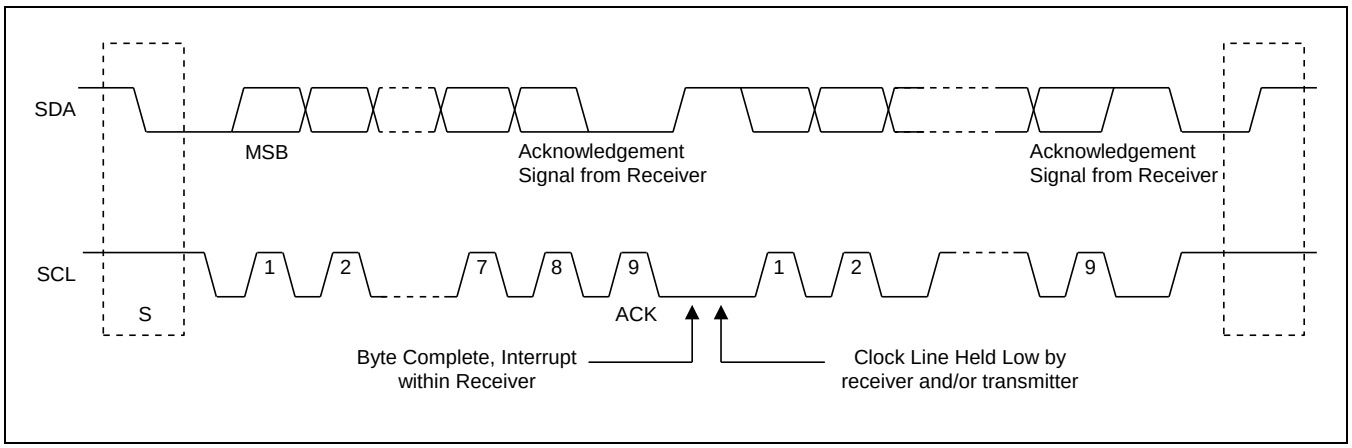


Figure 29-4 Data Transfer on the I2C-Bus

29.4.3 ACK Signal Transmission

To complete a 1-byte transfer operation, the receiver sends an ACK bit to the transmitter. The ACK pulse appears at the ninth clock of the SCL line. The I2C master device generates eight clock cycles to transmit or receive 1 byte data. The master generates clock pulse that is required to transmit the ACK bit.

When the transmitter receives ACK clock pulse, it sets the SDA line to High to release the SDA line. The receiver drives the SDA line Low during the ACK clock pulse to keep the SDA Low. This happens during the High period of the ninth SCL pulse. The software (I2CSTAT) enables or disables ACK bit transmit function. However, the ACK pulse on the ninth clock of SCL should complete the 1-byte data transfer operation.

Figure 29-5 illustrates the acknowledgement on the I2C-bus.

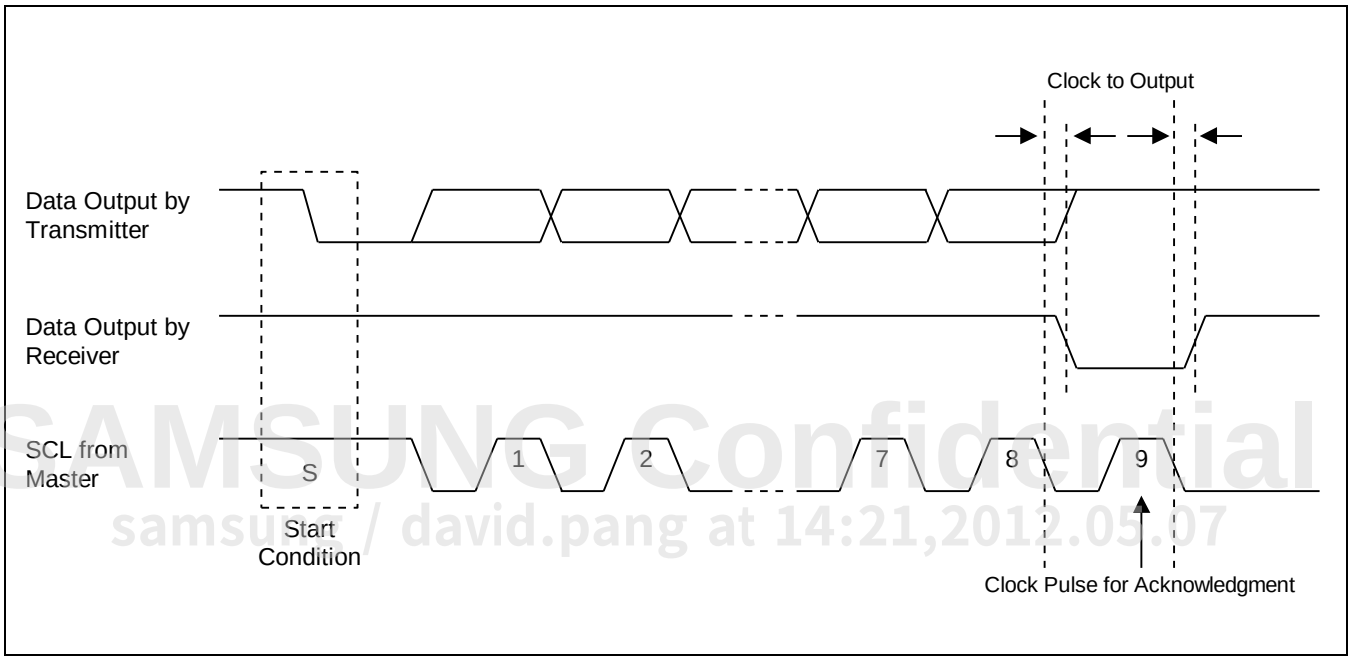


Figure 29-5 Acknowledgement on the I2C-Bus

29.4.4 Read-Write Operation

When the I2C controller transmits data in transmitter mode, the I2C-bus interface waits until I2C-bus Data Shift (I2CDS) register receives the new data. Before you write new data to the register, the SCL line is held Low. The I2C controller releases the SCL line after you write the data. Exynos 4412 SCP holds the interrupt to identify the completion of current data transfer. After the CPU receives the interrupt request, it writes new data to the I2CDS register again.

When the I2C controller receives data in receive mode, the I2C-bus interface waits until I2CDS register is Read. Before you read out the new data, the SCL line is held Low. The I2C controller releases the SCL line after you read the data. Exynos 4412 SCP holds the interrupt to identify the completion of new data reception. After the CPU receives the interrupt request, it reads the data from the I2CDS register.

29.4.5 Bus Arbitration Procedures

Arbitration occurs on the SDA line to prevent the conflict on the bus between two masters. If a master with a SDA High level detects other master with a SDA active Low level, it does not initiate a data transfer. This is because the current level on the bus is not corresponding to initiate a data transfer. The arbitration procedure extends until the SDA line turns High.

When two or more masters assert the SDA line Low simultaneously, each master evaluates whether it has the mastership or not. For the purpose of evaluation, each master detects the address bits. While each master generates the Slave address, it detects the address bit on the SDA line. This is because the SDA line becomes Low instead of High.

Let us assume that one master generates a Low as first address bit, while the other master is maintaining High. In such case, both masters detect Low on the bus. This is because the Low status is superior to the High status in power. When this happens, the Low (as the first bit of address) that generates master, gets the mastership while the High (as the first bit of address) that generates master, withdraws the mastership.

When both masters generate Low as the first bit of address, there is arbitration for the second address bit again. This arbitration continues till the end of last address bit.

29.4.6 Abort Conditions

When a Slave receiver cannot acknowledge the confirmation of the slave address, it holds the level of the SDA line High. In this case, the master generates a Stop condition and cancels the transfer.

When a master receiver is involved in the aborted transfer, it signals the end of Slave transmit operation by canceling the generation of an ACK. This happens after the Master receives the last data byte from the Slave. The Slave transmitter releases the SDA to enable a master to generate a Stop condition.

29.4.7 Configuring I2C-Bus

To control the frequency of SCL, you should write the 4-bit prescaler value in the I2CCON register. The I2C-bus interface address is stored in the I2C-bus address (I2CADD) register. By default, the I2C-bus interface address has an unknown value.

29.4.8 Flowcharts of Operations in Each Mode

Before you execute any I2C Tx/Rx operations:

1. If required, Write own Slave address on I2CADD register.
2. Set I2CCON register:
   a) Enable interrupt.
   b) Define SCL period.
3. Set I2CSTAT to enable Serial Output.

Figure 29-6 illustrates the operations for Master/Transmitter mode.

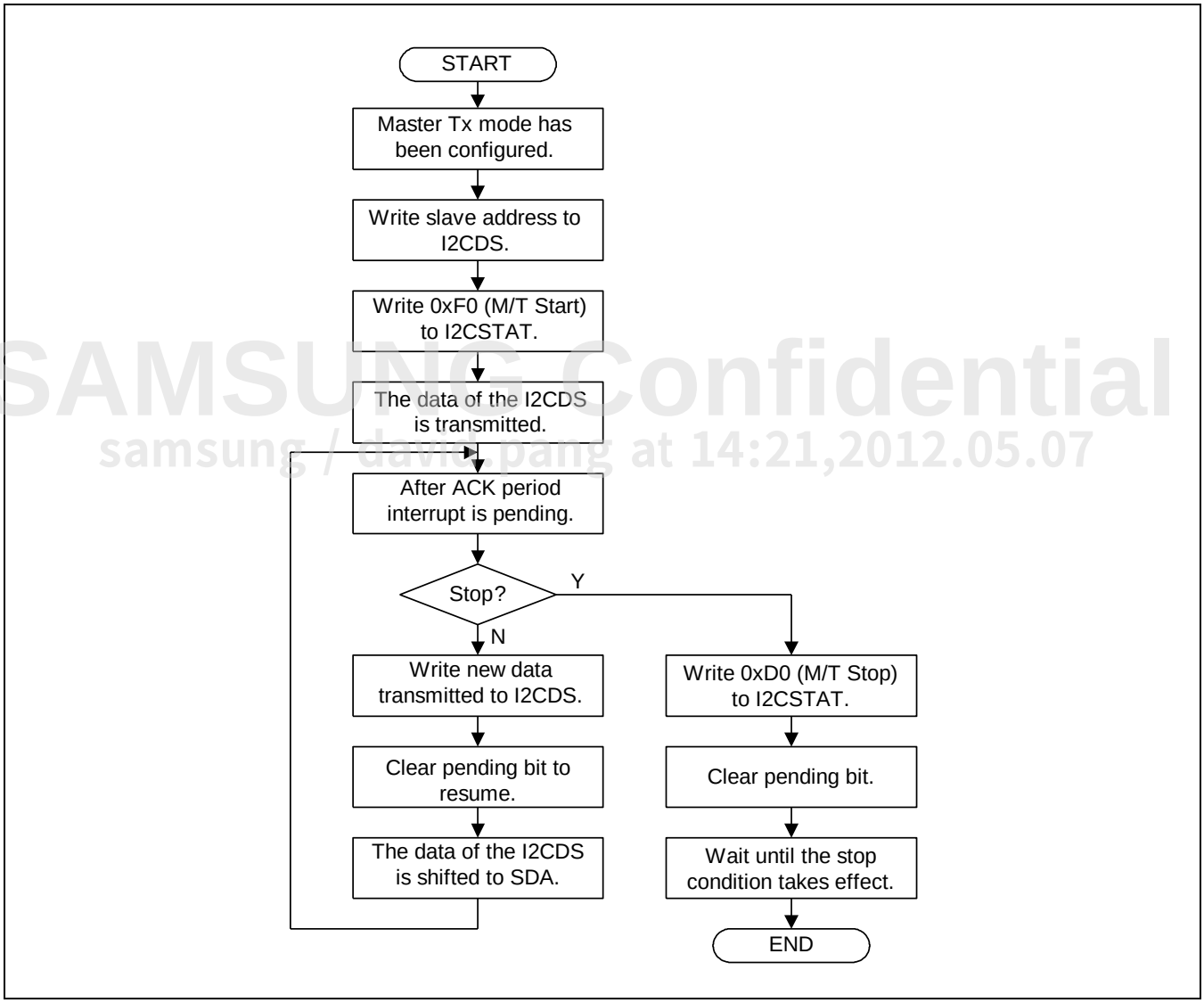


Figure 29-6 Operations for Master/Transmitter Mode

Figure 29-7 illustrates the operations for Master/Receiver Mode.

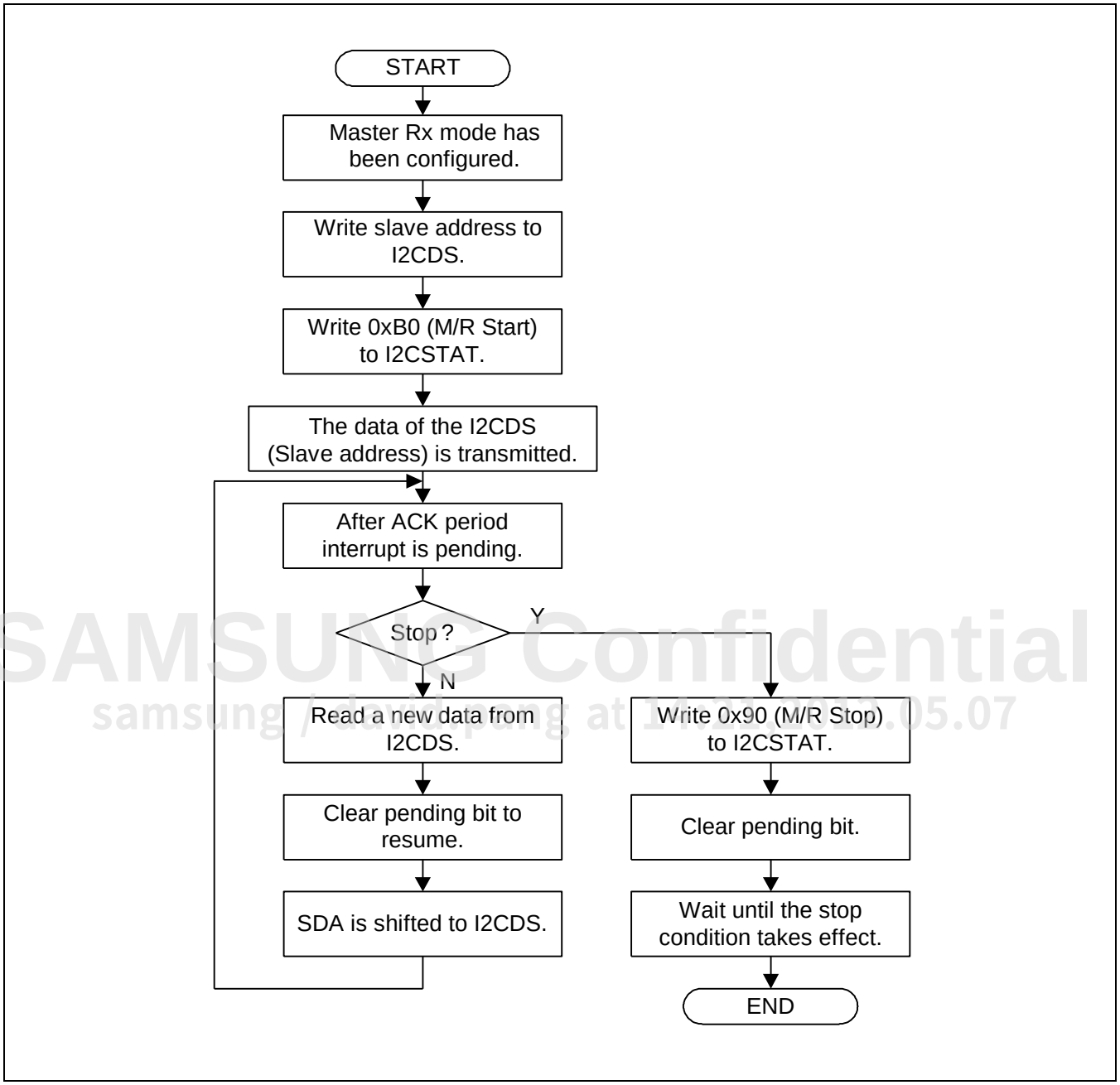


Figure 29-7 Operations for Master/Receiver Mode

Figure 29-8 illustrates the operations for Slave/Transmitter mode.

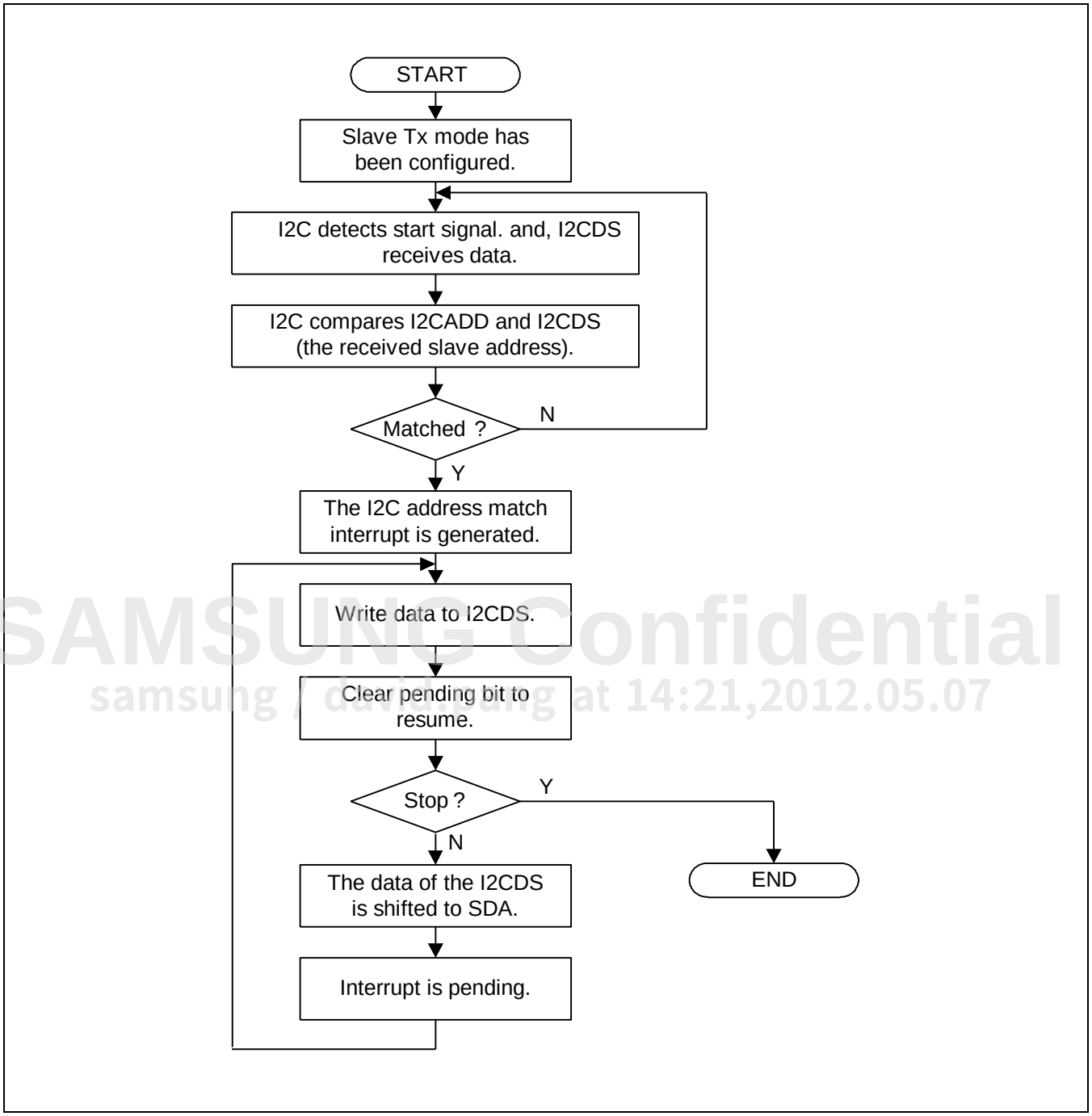


Figure 29-8 Operations for Slave/Transmitter Mode

Figure 29-9 illustrates the operations for Slave/Receiver Mode.

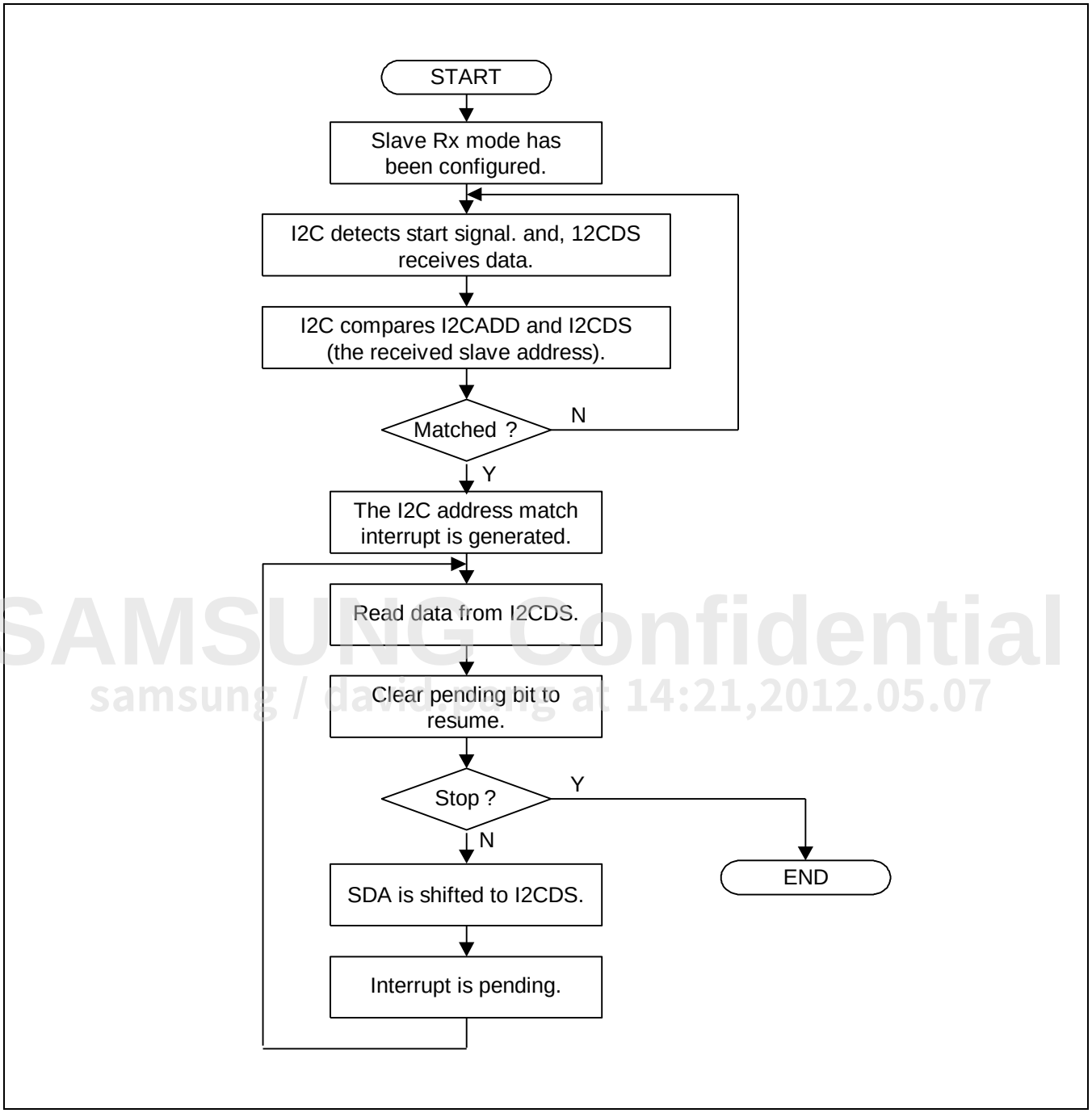


Figure 29-9 Operations for Slave/Receiver Mode

29.5 I/O Description

Signal	I/O	Description	Pad	Type
I2C0_SCL	Input/Output	I2C-bus interface0 serial clock line	XI2C0SCL	muxed
I2C0_SDA	Input/Output	I2C-bus interface0 serial data line	XI2C0SDA	muxed
I2C1_SCL	Input/Output	I2C-bus interface1 serial clock line	XI2C1SCL	muxed
I2C1_SDA	Input/Output	I2C-bus interface1 serial data line	XI2C1SDA	muxed
I2C2_SCL	Input/Output	I2C-bus interface2 serial clock line	XuRTSn_1	muxed
I2C2_SDA	Input/Output	I2C-bus interface2 serial data line	XuCTSn_1	muxed
I2C3_SCL	Input/Output	I2C-bus interface3 serial clock line	XuRTSn_2	muxed
I2C3_SDA	Input/Output	I2C-bus interface3 serial data line	XuCTSn_2	muxed
I2C4_SCL	Input/Output	I2C-bus interface4 serial clock line	XspiMOSI_0	muxed
I2C4_SDA	Input/Output	I2C-bus interface4 serial data line	XspiMISO_0	muxed
I2C5_SCL	Input/Output	I2C-bus interface5 serial clock line	XspiMOSI_1	muxed
I2C5_SDA	Input/Output	I2C-bus interface5 serial data line	XspiMISO_1	muxed
I2C6_SCL	Input/Output	I2C-bus interface6 serial clock line	Xi2s2SDO	muxed
I2C6_SDA	Input/Output	I2C-bus interface6 serial data line	Xi2s2SDI	muxed
I2C7_SCL	Input/Output	I2C-bus interface7 serial clock line	XpwmTOUT_3	muxed
I2C7_SDA	Input/Output	I2C-bus interface7 serial data line	XpwmTOUT_2	muxed

NOTE: The I2C bus interface for the HDMI has no external I/O.

29.6 Register Description

29.6.1 Register Map Summary

- Base Address: 0x1386_0000, 0x1387_0000, 0x1388_0000, 0x1389_0000, 0x138A_0000, 0x138B_0000, 0x138C_0000, 0x138D_0000, 0x138E_0000

Register	Offset	Description	Reset Value
I2CCONn	0x0000	Specifies the I2C-bus interface0 control register	0x0X
I2CSTATn	0x0004	Specifies the I2C-bus interface0 control/status register	0x00
I2CADDn	0x0008	Specifies the I2C-bus interface0 address register	0xXX
I2CDSn	0x000C	Specifies the I2C-bus interface0 transmit/receive data shift register	0xXX
I2CLCn	0x0010	Specifies the I2C-bus interface0 multi-master line control register	0x00

29.6.1.1 I2CCONn (n = 0 to 7)

- Base Address: 0x1386_0000, 0x1387_0000, 0x1388_0000, 0x1389_0000, 0x138A_0000, 0x138B_0000, 0x138C_0000, 0x138D_0000, 0x138E_0000
- Address = Base Address + 0x0000, Reset Value = 0x0X

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Acknowledge generation (1)	[7]	RW	I2C-bus acknowledge enable bit 0 = Disables 1 = Enables In Tx mode, the I2CSDA is idle in the ACK time. In Rx mode, the I2CSDA is low in the ACK time.	0
Tx clock source selection	[6]	RW	Source clock of I2C-bus transmit clock prescaler selection bit 0 = I2CCLK = fPCLK/16 1 = I2CCLK = fPCLK/512	0
Tx/Rx Interrupt (5)	[5]	RW	I2C-bus Tx/Rx interrupt enable/ disable bit 0 = Disables 1 = Enables	0
Interrupt pending flag (2), (3)	[4]	S	I2C-bus Tx/Rx interrupt pending flag You cannot write this bit to 1. If you read this bit as 1, the I2CSCL is tied to Low and the I2C is stopped. To resume the operation, write this bit as 0. 0 = 1) No interrupt is pending (If Read). 2) Clears pending condition and resumes the operation (If Write). 1 = 1) Interrupt is pending (If Read) 2) N/A (If Write)	0
Transmit clock value (4)	[3:0]	RW	I2C-bus transmit clock prescaler 4-bit prescaler value determines the I2C-bus transmit clock frequency according to the formula given here: Tx clock = I2CCLK/(I2CCON[3:0] + 1).	–

NOTE:

1. While interfacing with EEPROM, the ACK generation is disabled before Reading the last data to generate the STOP condition in Rx mode.
2. An I2C-bus interrupt occurs when:
   (a) 1-byte Transmit or Receive operation is complete. Alternatively, the ACK period is finished.
   (b) A general call or a Slave address match occurs.
   (c) Bus arbitration fails.
3. To adjust the setup time of SDA before SCL rising edge, ensure to Write I2CDS before clearing the I2C interrupt pending bit.
4. I2CCON[6] determines I2CCLK. Tx clock can vary by SCL transition time.
   When I2CCON[6] = 0, I2CCON[3:0] = 0x0 or 0x1 is not available.
5. When I2CCON[5] = 0, I2CCON[4] does not operate correctly.
   Therefore, set I2CCON[5] = 1 even if you do not use the I2C interrupt.

29.6.1.2 I2CSTATn (n = 0 to 7)

- Base Address: 0x1386_0000, 0x1387_0000, 0x1388_0000, 0x1389_0000, 0x138A_0000, 0x138B_0000, 0x138C_0000, 0x138D_0000, 0x138E_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Mode selection	[7:6]	RWX	I2C-bus Master/Slave Tx/Rx mode select bits 00 = Slave receive mode 01 = Slave transmit mode 10 = Master receive mode 11 = Master transmit mode	00
Busy signal status/START STOP condition	[5]	S	I2C-bus busy signal status bit 0 = (Read) Not busy (If Read) (write) STOP signal generation 1 = (Read) Busy (If Read) (write) START signal generation. Transfers the data in I2CDS automatically just after the start signal.	0
Serial output	[4]	S	I2C-bus data output enable/ disable bit 0 = Disables Rx/Tx 1 = Enables Rx/Tx	0
Arbitration status flag	[3]	RO	I2C-bus arbitration procedure status flag bit 0 = Bus arbitration successful 1 = Bus arbitration fails during serial I/O	0
Address-as-slave status flag	[2]	RO	I2C-bus address-as-slave status flag bit 0 = Clears when it detects START/STOP condition 1 = Receives slave address that matches the address value in the I2CADD	0
Address zero status flag	[1]	RO	I2C-bus address zero status flag bit 0 = Clears when it detects START/ STOP condition 1 = Received slave address is 00000000b	0
Last-received bit status flag	[0]	RO	I2C-bus last-received bit status flag bit 0 = Last-received bit is set to 0 (receives ACK). 1 = Last-received bit is set to 1 (does not receive ACK).	0

29.6.1.3 I2CADDn (n = 0 to 7)

- Base Address: 0x1386_0000, 0x1387_0000, 0x1388_0000, 0x1389_0000, 0x138A_0000, 0x138B_0000, 0x138C_0000, 0x138D_0000, 0x138E_0000
- Address = Base Address + 0x0008, Reset Value = 0xXX

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Slave address	[7:0]	RWX	7-bit slave address, latched from the I2C-bus. When serial output enable = 0 in the I2CSTAT, I2CADD is write-enabled. The I2CADD value is Read any time, regardless of the current serial output enable bit (I2CSTAT) setting. Slave address: [7:1] Not mapped: [0]	–

29.6.1.4 I2CDSn (n = 0 to 7)

- Base Address: 0x1386_0000, 0x1387_0000, 0x1388_0000, 0x1389_0000, 0x138A_0000, 0x138B_0000, 0x138C_0000, 0x138D_0000, 0x138E_0000
- Address = Base Address + 0x000C, Reset Value = 0xXX

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
Data shift	[7:0]	RWX	8-bit data shift register for I2C-bus Tx/Rx operation. When serial output enable = 1 in the I2CSTAT, I2CDS is write-enabled. The I2CDS value is Read any time, regardless of the current serial output enable bit (I2CSTAT) setting.	–

29.6.1.5 I2CLCn (n = 0 to 7)

- Base Address: 0x1386_0000, 0x1387_0000, 0x1388_0000, 0x1389_0000, 0x138A_0000, 0x138B_0000, 0x138C_0000, 0x138D_0000, 0x138E_0000
- Address = Base Address + 0x0010, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0
Filter enable	[2]	RW	I2C-bus filter enable bit When SDA port is operating as input, set this bit to High. This filter prevents error caused by glitch between two PCLK clocks. 0 = Disables Filter 1 = Enables Filter	0
SDA output delay	[1:0]	RW	I2C-bus SDA line delay length selection bits The I2C controller delays the SDA line by following clock cycle: 00 = 0 clock 01 = 5 clocks 10 = 10 clocks 11 = 15 clocks	00

30 Serial Peripheral Interface

30.1 Overview

Serial Peripheral Interface (SPI) in Exynos 4412 SCP transfers serial data by using various peripherals. SPI includes two 8, 16, and 32-bit shift registers to transmit and receive data. During an SPI transfer, it simultaneously transmits (shifts out serially) and receives (shifts in serially) data. SPI supports the protocols for National Semiconductor Microwire and Motorola Serial Peripheral Interface.

30.2 Features

The features of SPI include:

- Full duplex
- 8/16/32-bit shift register for Tx/Rx
- Supports 8-bit/16-bit/32-bit bus interface
- Supports the Motorola SPI protocol and National Semiconductor Microwire
- Two independent 32-bits wide transmit and receive FIFOs: depth 64 in port 0 and depth 16 in port 1 and 2
- Master-mode and Slave-mode
- Receive-without-transmit operation
- Tx/Rx maximum frequency at up to 50 MHz

30.2.1 Operation of SPI

The SPI transfers 1-bit serial data between Exynos 4412 SCP and external device. The SPI in Exynos 4412 SCP supports the CPU or DMA to transmit or receive FIFOs separately and to transfer data in both directions simultaneously. SPI has two channels, namely, Tx channel and Rx channel. Tx channel has the path from Tx FIFO to external device. Rx channel has the path from external device to Rx FIFO.

CPU (or DMA) must write data on the register SPI_TX_DATA, to write data in FIFO. Data on the register are automatically moved to Tx FIFOs. To read data from Rx FIFOs, CPU (or DMA) must access the register SPI_RX_DATA and data are automatically sent to the SPI_RX_DATA register.

CMU registers can control SPI operating frequency. Refer to "CMU" chapter for more information.

30.2.1.1 Operation Mode

SPI has two modes, namely, master and slave mode. In master mode, SPICLK is generated and transmitted to external device. XspiCS#, which is the signal to select slave, indicates that the data is valid when XspiCS# is set to low level. XspiCS# must be set low before packets are transmitted or received.

30.2.1.2 FIFO Access

The SPI supports CPU access and DMA access to FIFOs. Data size of CPU access and DMA access to FIFOs are selected either from 8-bit, 16-bit, or 32-bit data. When it selects 8-bit data size, then valid bits are from 0 to 7-bit. User can define the trigger threshold to raise interrupt to CPU. The trigger level of each FIFO in port 0 is set by 4 bytes step from 0 to 252 bytes, and that of each FIFO in port 1 is set by 1 byte step from 0 to 63 bytes. TxDMAOn or RxDMAOn bit of SPI_MODE_CFG register must be set to use DMA access. DMA access supports only single transfer and 4-burst transfer. In Tx FIFO, DMA request signal is high until Tx FIFO is full. In Rx FIFO, DMA request signal is high if FIFO is not empty.

30.2.1.3 Trailing Bytes in the Rx FIFO

When the number of samples in Rx FIFO is less than the threshold value in INT mode or DMA 4-burst mode and it does not receive any additional data, then the remaining bytes are called trailing bytes. To remove these bytes in Rx FIFO, it uses internal timer and interrupt signal. The value of internal timer is set up to 1024 clocks based on APB BUS clock. When timer value is zero, interrupt signal occurs and CPU can remove trailing bytes in FIFO.

30.2.1.4 Packet Number Control

SPI controls the number of packets to be received in master mode. Set SFR (PACKET_CNT_REG) to receive any number of packets. SPI stops generating SPICLK if the number of packets is similar to PACKET_CNT_REG. The size of one packet depends on channel width. (One packet is one byte when you configure channel width as byte, and one packet is four bytes when you configure channel width as word.) It is mandatory to follow software or hardware reset before reloading this function. (Software reset can clear all registers except special function registers, but hardware reset clears all registers.)

30.2.1.5 Chip Select Control

Chip select XspiCS# is active low signal. In other words, a chip is selected when XspiCS# input is 0.

You can control XspiCS# automatically or manually. No need to change. When you use manual control mode, you should clear AUTO_N_MANUAL (default value is 0). NSSOUT bit controls XspiCS# level.

When you use auto control mode, AUTO_N_MANUAL must be set as 1. XspiCS toggled between packet and packet automatically. NCS_TIME_COUNT controls inactive period of XspiCS. NSSOUT is not available at this time.

30.2.1.6 High Speed Operation as Slave

Exynos 4412 SCP SPI supports Tx/Rx operations up to 50 MHz, but there is a limitation. When Exynos 4412 SCP SPI works as a slave, it consumes large delay more than 15 ns in worst operating condition. Such a large delay can cause setup violation at SPI master device. To overcome the problem, Exynos 4412 SCP SPI provides fast slave Tx mode by setting 1 to HIGH_SPEED bit of CH_CFG register. In that mode, it reduces MISO output delay by half cycle, so that the SPI master device has more setup margin.
However, you can use the fast slave Tx mode only when CPHA = 0.

30.2.1.7 Feedback Clock Selection

Under SPI protocol specification, SPI master should capture the input data launched by slave (MISO) with its internal SPICLK. When SPI runs at high operating frequency such as 50 MHz, it is difficult to capture the MISO input because the required arrival time of MISO is half cycle period in Exynos 4412 SCP. It is shorter than the arrival time of MISO that consists of SPICLK output delay of SPI master, MISO output delay of SPI slave, and MISO input delay of SPI master. To overcome the problem, Exynos 4412 SCP SPI provides three feedback clocks that are phase-delayed clock of internal SPICLK.

A selection of feedback clock depends on MISO output delay of SPI slave. To capture MISO data correctly, it selects the feedback clock that satisfies the following constraint:

$t_{SPIMIS}(s) < t_{period}/2 - t_{SPISOD}$

* $t_{SPIMIS}(s)$: MISO input setup time of SPI master on a given feedback clock selection "s"
* t_{SPISOD}: MISO output delay of SPI slave
* t_{period}: SPICLK cycle period

If multiple feedback clocks meet the constraint, then it should select the feedback clock with smallest phase delay. Because of a feedback clock with large phase delay, it may capture data of next cycle.

For example of Exynos 4412 SCP, SPI CH1 with master configuration of 50 MHz operating frequency, 1.8 V external voltage and 15 pF load, if it assumes MISO output delay of SPI slave as 11 ns ($t_{SPIMIS}(s) < 10$ ns – 11 ns = – 1 ns), then it should use 270 degree phase-delayed feedback clock.

If the operating clock frequency is 33 MHz and other conditions are similar to the previous example, it is better to use 180 degree phase-delayed feedback clock ($t_{SPIMIS}(s) < 15$ ns – 11 ns = 4 ns).

30.2.1.8 SPI Transfer Format

The Exynos 4412 SCP supports four different formats for data transfer.

Figure 30-1 illustrates four waveforms for SPICLK.

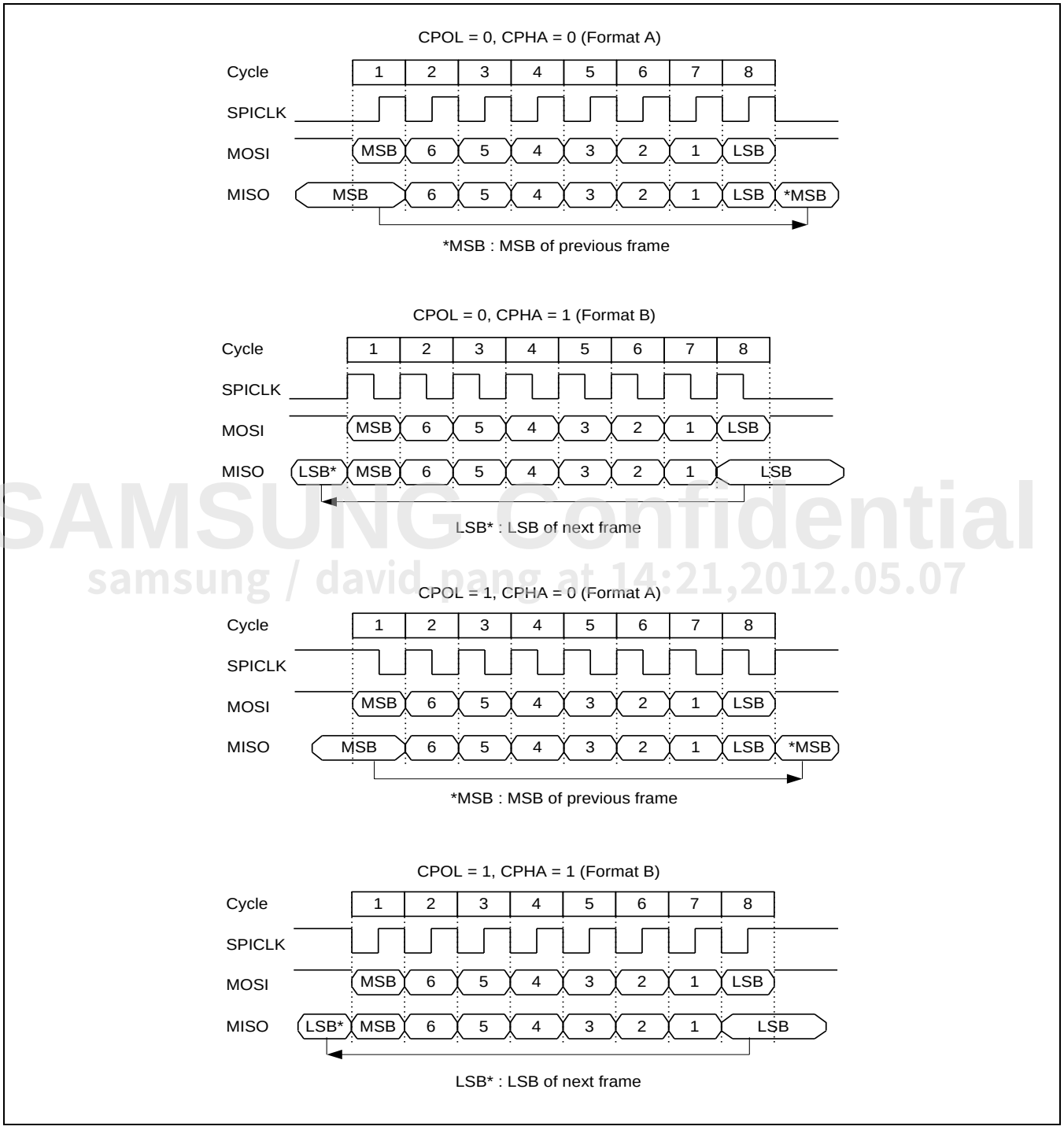


Figure 30-1 SPI Transfer Format

30.3 SPI Input Clock Description

Figure 1-2 illustrates the input clock diagram for SPI.

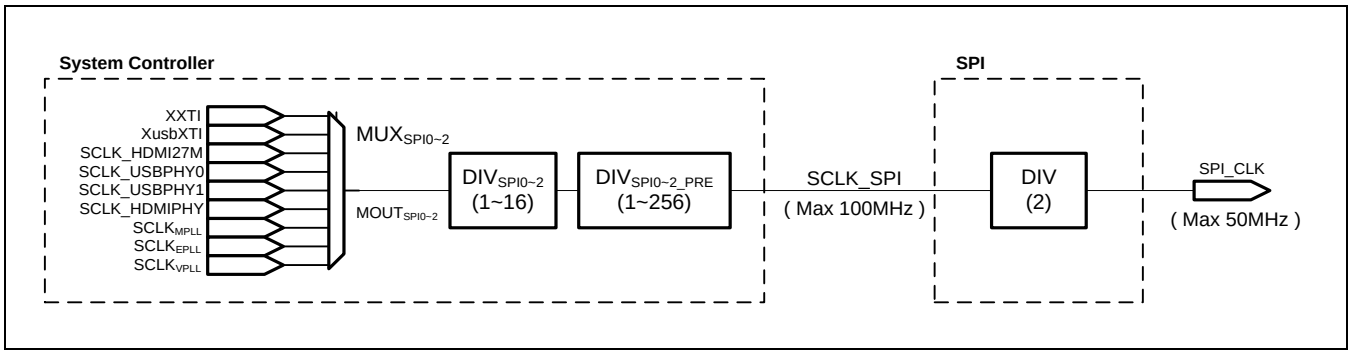


Figure 30-2 Input Clock Diagram for SPI

Exynos 4412 SCP provides SPI with a variety of clocks. As illustrated in the *Figure 28-8*, the SPI uses SCLK_SPI clock, which is from clock controller. You can also select SCLK_SPI from various clock sources. To select SCLK_SPI, refer to Chapter 7 Clock Controller for more information.

NOTE: SPI has an internal 2x clock divider. You should configure SCLK_SPI to have a double of the SPI operating clock frequency.

30.4 IO Description

The IO description table lists the external signals between the SPI and external device. The unused SPI ports are used as General Purpose I/O ports. Refer to "General Purpose I/O" chapter for more information.

Signal	I/O	Description	Pad	Type
SPI_0_CLK SPI_1_CLK SPI_2_CLK	In/Out	XspiCLK is the serial clock used to control time of data transfer. Out: when used as master In: when used as slave	XspiCLK_0 XspiCLK_1 Xi2s2CDCLK	muxed
SPI_0_nSS SPI_1_nSS SPI_2_nSS	In/Out	Slave selection signal. All data Tx/Rx sequences are executed if XspiCS is low. Out: when used as master In: when used as slave	XspiCSn_0 XspiCSn_1 Xi2s2LRCK	muxed
SPI_0_MISO SPI_1_MISO SPI_2_MISO	In/Out	This port is the input port in Master mode. You can use input mode to get data from slave output port. It transmits data to master through this port in slave mode. Out: when used as slave In: when used as master	XspiMISO_0 XspiMISO_1 Xi2s2SDI	muxed
SPI_0_MOSI SPI_1_MOSI SPI_2_MOSI	In/Out	This port is the output port in Master mode. It uses this port to transfer data from master output port. It receives data from master through this port in slave mode. Out: when used as master In: when used as slave	XspiMOSI_0 XspiMOSI_1 Xi2s2SDO	muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

30.5 Register Description

30.5.1 Register Map Summary

- Base Address: 0x1392_0000, 0x1393_0000, 0x1394_0000

Register	Offset	Description	Reset Value
CH_CFGn	0x0000	Specifies SPI configuration	0x0
MODE_CFGn	0x0008	Specifies FIFO control	0x0
CS_REGn	0x000C	Specifies slave selection control	0x1
SPI_INT_ENn	0x0010	Specifies interrupt enable	0x0
SPI_STATUSn	0x0014	Specifies SPI status	0x0
SPI_TX_DATAn	0x0018	Specifies Tx data	0x0
SPI_RX_DATAn	0x001C	Specifies Rx data	0x0
PACKET_CNT_REGn	0x0020	Specifies packet count	0x0
PENDING_CLR_REGn	0x0024	Specifies interrupt pending clear	0x0
SWAP_CFGn	0x0028	Specifies swap configuration	0x0
FB_CLK_SELn	0x002C	Specifies feedback clock selection	0x0

Setting Sequence of Special Function Register

Steps to set Special Function Register (nCS manual mode) are:

1. Set Transfer Type. (CPOL and CPHA set)
2. Set Feedback Clock Selection register.
3. Set SPI MODE_CFG register.
4. Set SPI INT_EN register.
5. Set PACKET_CNT_REG register if necessary.
6. Set Tx or Rx Channel on.
7. Set nSSout low to start Tx or Rx operation:
   a. Set nSSout Bit to low, then start Tx data writing.
   b. When auto chip selection bit is set, nSSout is controlled automatically.

30.5.1.1 CH_CFGn (n = 0 to 2)

- Base Address: 0x1392_0000, 0x1393_0000, 0x1394_0000
- Address = Base Address + 0x0000, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	–
HIGH_SPEED_EN	[6]	RW	Slave Tx output time control bit If this bit is enabled, slave Tx output time is reduced as much as half period of SPICLKout period. This bit is valid only in CPHA 0. 0 = Disables 1 = Enables	0
SW_RST	[5]	RW	Software Reset The following registers and bits are cleared by this bit. Rx/Tx FIFO data, SPI_STATUS register will be reset once in the initial time. And after that, if we want to reset the register again, we have to use SW_RST bit manually. 0 = Inactive 1 = Active	0
SLAVE	[4]	RW	Whether SPI Port is Master or Slave 0 = Master 1 = Slave	0
CPOL	[3]	RW	Determines whether active high or active low clock 0 = Active high 1 = Active low	0
CPHA	[2]	RW	Select one of the two fundamentally different transfer format 0 = Format A 1 = Format B	0
RX_CH_ON	[1]	RW	SPI Rx Channel On 0 = Channel off 1 = Channel on	0
TX_CH_ON	[0]	RW	SPI Tx Channel On 0 = Channel off 1 = Channel on	0

NOTE: SPI controller should reset when:

1. Reconfiguration of SPI registers is done.
2. Error interrupt has occurred.

30.5.1.2 MODE_CFGn (n = 0 to 2)

- Base Address: 0x1392_0000, 0x1393_0000, 0x1394_0000
- Address = Base Address + 0x0008, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	–
CH_WIDTH	[30:29]	RW	00 = Byte 01 = Halfword 10 = Word 11 = Reserved	0
TRAILING_CNT	[28:19]	RW	Count value from writing the last data in Rx FIFO to flush trailing bytes in FIFO	0
BUS_WIDTH	[18:17]	RW	00 = Byte 01 = Halfword 10 = Word 11 = Reserved	0
RX_RDY_LVL	[16:11]	RW	Rx FIFO trigger level in INT mode. Port 0: Trigger level (bytes) = 4 × N Port 1, 2: Trigger level (bytes) = N (N = value of RX_RDY_LVL field)	0
TX_RDY_LVL	[10:5]	RW	Tx FIFO trigger level in INT mode. Port 0: Trigger level (bytes) = 4 × N Port 1, 2: Trigger level (bytes) = N (N = value of TX_RDY_LVL field)	0
RSVD	[4:3]	–	Reserved	–
RX_DMA_SW	[2]	RW	Rx DMA mode enable/disable 0 = Disables DMA Mode 1 = Enables DMA Mode	0
TX_DMA_SW	[1]	RW	Tx DMA mode enable/disable 0 = Disables DMA Mode 1 = Enables DMA Mode	0
DMA_TYPE	[0]	RW	DMA transfer type, single or 4 bursts. 0 = Single 1 = 4 burst DMA transfer size must be set as the same size in SPI DMA.	0

NOTE:

1. CH_WIDTH is shift-register width.
2. BUS_WIDTH is SPI FIFO width, transfer data size should be aligned with BUS_WIDTH.
   For example, Tx/Rx data size must be aligned with 4 bytes if BUS_WIDTH is word.
3. CH_WIDTH must be smaller than BUS_WIDTH or similar to BUS_WIDTH.

30.5.1.3 CS_REGn (n = 0 to 2)

- Base Address: 0x1392_0000, 0x1393_0000, 0x1394_0000
- Address = Base Address + 0x000C, Reset Value = 0x1

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	–
NCS_TIME_COUNT	[9:4]	RW	NSSOUT inactive time = ((nCS_time_count + 3)/2) × SPICLKout	0
RSVD	[3:2]	–	Reserved	–
AUTO_N_MANUAL	[1]	RW	Chip select toggle manual or auto selection 0 = Manual 1 = Auto	0
NSSOUT	[0]	RW	Slave selection signal (manual only) 0 = Active 1 = Inactive	1

When AUTO_N_MANUAL is set, then SPI controller controls NSSOUT and does not perform data transfer continuously.

Unit data size depends on CH_WIDTH.

Figure 30-3 illustrates auto chip select mode waveform.

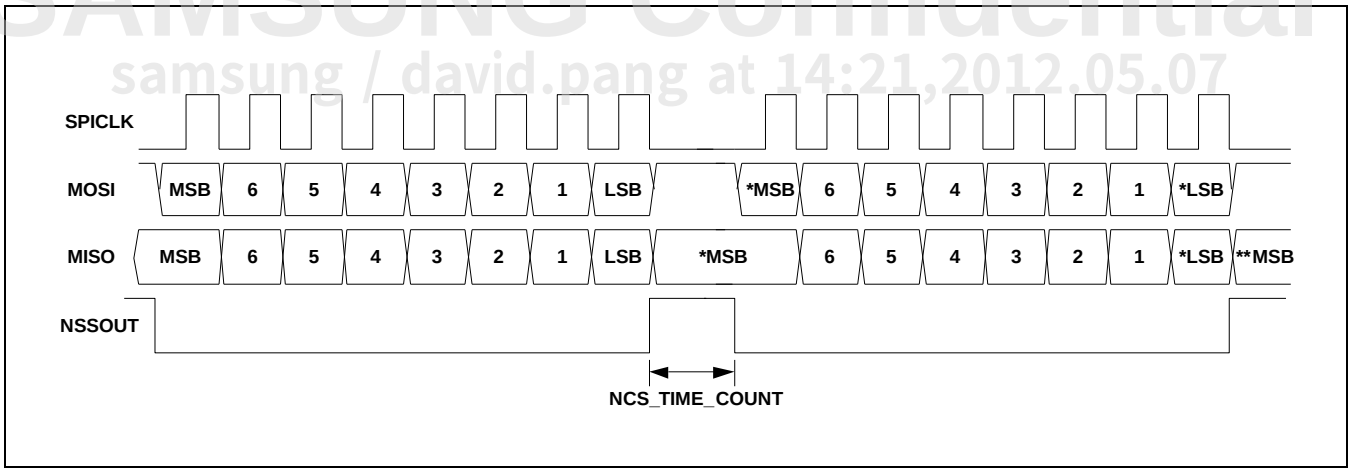


Figure 30-3 Auto Chip Select Mode Waveform (CPOL = 0, CPHA = 0, CH_WIDTH = Byte)

30.5.1.4 SPI_INT_ENn (n = 0 to 2)

- Base Address: 0x1392_0000, 0x1393_0000, 0x1394_0000
- Address = Base Address + 0x0010, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	–
INT_EN_TRAILING	[6]	RW	Interrupt Enable for trailing count to be 0 0 = Disables 1 = Enables	0
INT_EN_RX_ OVERRUN	[5]	RW	Interrupt Enable for RxOverrun 0 = Disables 1 = Enables	0
INT_EN_RX_ UNDERRUN	[4]	RW	Interrupt Enable for RxUnderrun 0 = Disables 1 = Enables	0
INT_EN_TX_ OVERRUN	[3]	RW	Interrupt Enable for TxOverrun 0 = Disables 1 = Enables	0
INT_EN_TX_ UNDERRUN	[2]	RW	Interrupt Enable for TxUnderrun. In slave mode, this bit must be clear first after turning on slave Tx path. 0 = Disables 1 = Enables	0
INT_EN_RX_FIFO_ RDY	[1]	RW	Interrupt Enable for RxFifoRdy (INT mode) 0 = Disables 1 = Enables	0
INT_EN_TX_FIFO_ RDY	[0]	RW	Interrupt Enable for TxFifoRdy (INT mode) 0 = Disables 1 = Enables	0

30.5.1.5 SPI_STATUSn (n = 0 to 2)

- Base Address: 0x1392_0000, 0x1393_0000, 0x1394_0000
- Address = Base Address + 0x0014, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	–
TX_DONE	[25]	R	Indication of transfer done in Shift register (master mode only) 0 = All case except below case 1 = If Tx FIFO and shift register are empty after transmission start	0
TRAILING_BYTE	[24]	R	Indication that trailing count is 0	0
RX_FIFO_LVL	[23:15]	R	Data level in Rx FIFO 0 to 256 bytes in port 0 0 to 64 bytes in port 1, 2	0
TX_FIFO_LVL	[14:6]	R	Data level in Tx FIFO 0 to 256 bytes in port 0 0 to 64 bytes in port 1, 2	0
RX_OVERRUN	[5]	R	Rx FIFO overrun error 0 = No error 1 = Overrun error	0
RX_UNDERRUN	[4]	R	Rx FIFO underrun error 0 = No error 1 = Underrun error	0
TX_OVERRUN	[3]	R	Tx FIFO overrun error 0 = No error 1 = Overrun error	0
TX_UNDERRUN	[2]	R	Tx FIFO underrun error 0 = No error 1 = Underrun error NOTE: Tx FIFO underrun error will occur if Tx FIFO is empty in slave mode.	0
RX_FIFO_RDY	[1]	R	0 = Data in FIFO less than trigger level 1 = Data in FIFO more than trigger level	0
TX_FIFO_RDY	[0]	R	0 = Data in FIFO more than trigger level 1 = Data in FIFO less than trigger level	0

30.5.1.6 SPI_TX_DATAn (n = 0 to 2)

- Base Address: 0x1392_0000, 0x1393_0000, 0x1394_0000
- Address = Base Address + 0x0018, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
TX_DATA	[31:0]	W	This field contains the data to be transmitted over the SPI channel.	0

30.5.1.7 SPI_RX_DATAn (n = 0 to 2)

- Base Address: 0x1392_0000, 0x1393_0000, 0x1394_0000
- Address = Base Address + 0x001C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RX_DATA	[31:0]	R	This field contains the data to be received over the SPI channel.	0

30.5.1.8 PACKET_CNT_REGn (n = 0 to 2)

- Base Address: 0x1392_0000, 0x1393_0000, 0x1394_0000
- Address = Base Address + 0x0020, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	–
PACKET_CNT_EN	[16]	RW	Enable bit for packet count 0 = Disables 1 = Enables	0
COUNT_VALUE	[15:0]	RW	Packet count value	0

30.5.1.9 PENDING_CLR_REGn (n = 0 to 2)

- Base Address: 0x1392_0000, 0x1393_0000, 0x1394_0000
- Address = Base Address + 0x0024, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	–
TX_UNDERRUN_CLR	[4]	RW	Tx underrun pending clear bit 0 = Non-Clear 1 = Clears	0
TX_OVERRUN_CLR	[3]	RW	Tx overrun pending clear bit 0 = Non-Clear 1 = Clears	0
RX_UNDERRUN_CLR	[2]	RW	Rx underrun pending clear bit 0 = Non-clear 1 = Clears	0
RX_OVERRUN_CLR	[1]	RW	Rx overrun pending clear bit 0 = Non-Clear 1 = Clears	0
TRAILING_CLR	[0]	RW	Trailing pending clear bit 0 = Non-Clear 1 = Clears	0

NOTE: After error interrupt pending clear, SPI controller should be reset.
Error interrupt list: Tx underrun, Tx overrun, Rx underrun, and Rx overrun.

30.5.1.10 SWAP_CFGn (n = 0 to 2)

- Base Address: 0x1392_0000, 0x1393_0000, 0x1394_0000
- Address = Base Address + 0x0028, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
RX_HWORD_SWAP	[7]	RW	0 = Off 1 = Swap	0
RX_BYTE_SWAP	[6]	RW	0 = Off 1 = Swap	0
RX_BIT_SWAP	[5]	RW	0 = Off 1 = Swap	0
RX_SWAP_EN	[4]	RW	Swap Enable 0 = Normal 1 = Swap	0
TX_HWORD_SWAP	[3]	RW	0 = Off 1 = Swap	0
TX_BYTE_SWAP	[2]	RW	0 = Off 1 = Swap	0
TX_BIT_SWAP	[1]	RW	0 = Off 1 = Swap	0
TX_SWAP_EN	[0]	RW	Swap Enable 0 = Normal 1 = Swap	0

NOTE: Data size must be larger than swap size.

30.5.1.11 FB_CLK_SELn (n = 0 to 2)

- Base Address: 0x1392_0000, 0x1393_0000, 0x1394_0000
- Address = Base Address + 0x002C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	–
FB_CLK_SEL	[1:0]	RW	In master mode, SPI uses a clock which is feedback from the SPICLK. The feedback clock is intended to capture safely the slave Tx signal. The slave Tx signal can lag if slave device is very far. There are four types of feedback clocks which experience different path delays. This register selects the feedback clock that you can use. Note that this register value is invalid when SPI operates in slave mode. 00 = SPICLK bypass (do not use feedback clock) 01 = A feedback clock with 90 degree phase lagging 10 = A feedback clock with 180 degree phase lagging 11 = A feedback clock with 270 degree phase lagging 90 degree phase lagging means 5 ns delay in 50 MHz operating frequency.	0x0

PAD Driving Strength

PAD driving strength of SPI is controlled by setting drive strength control register in GPIO. SPI related SFR is GPBDRV (for SPI port 0 and 1) and GPCDRV (for SPI port 2).

31 USB 2.0 Host Controller

31.1 Overview

Exynos 4412 SCP supports three ports USB host interface. The key features of this interface are:

- Complies with Enhanced HCI (EHCI) Revision 1.0a and Open HCI (OHCI) Revision 1.0 specifications (Both EHCI and OHCI compatible).
- Complies with USB Revision 2.0 specification.
- Complies with USB Revision 1.1 specification.
- Complies with either legacy UTMI, Revision 1.05, or UTMI + Level3, Revision 1.0
- Complies with High-Speed Inter-Chip (HSIC), Version 1.0
- Supports high-speed (480 Mbps transfer) peripherals.
- Supports power management features, such as:
  - Full Suspend / Resume functionality, including Remote Wakeup
  - Over-current protection on ports hooks for master clock suspension

31.2 Block Diagram

USB HOST 2.0 controller is composed of two independent blocks, namely, USB HOST 2.0 controller and USB PHY Controller. Each of these blocks has an AHB Slave interface that provides the microcontroller with Read and Write access to the Control and Status Registers (CSRs). The HOST Link has an AHB Master to enable the link to transfer data on the AHB. USB HOST has three Ports. You can use Port0 for Standard USB. You can use Port1 and Port2 for High-Speed Inter-Chip USB (HSIC).

Figure 31-1 illustrates the block diagram of USB system.

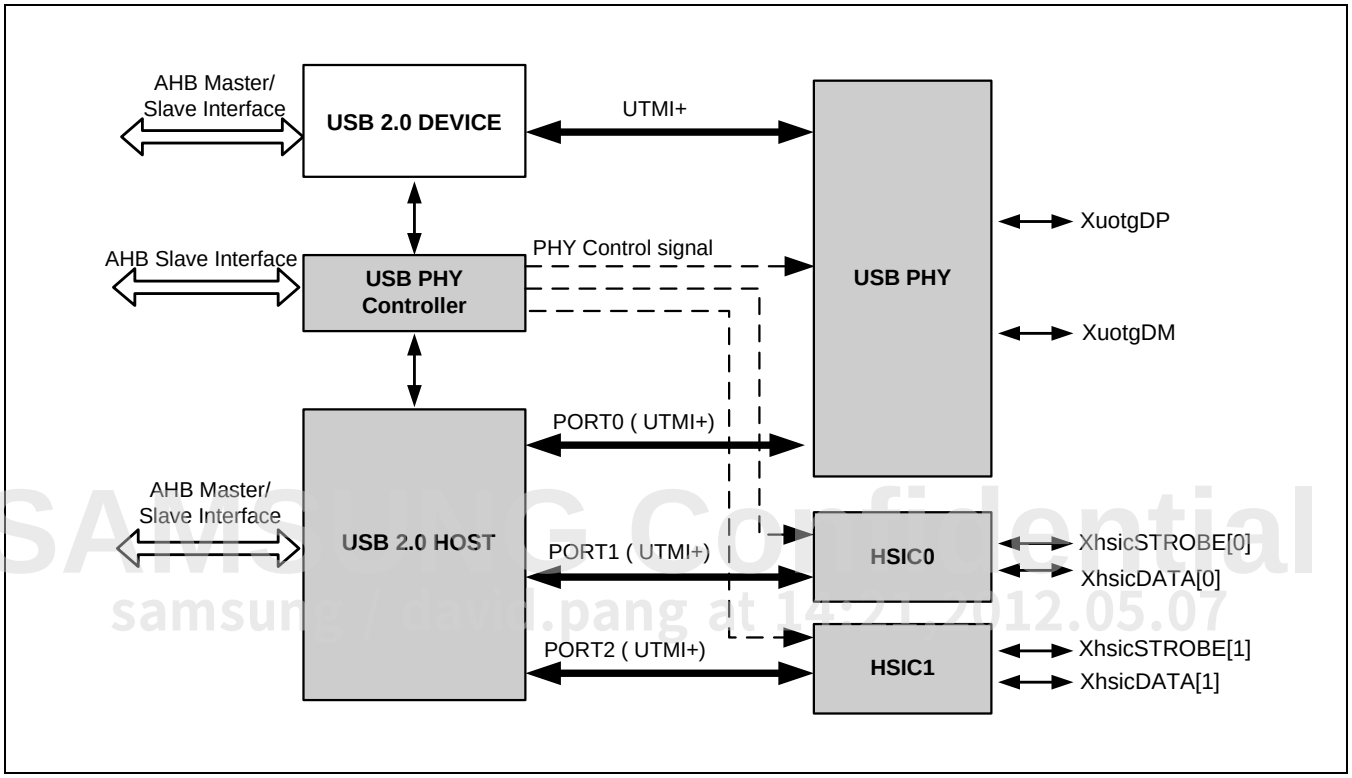


Figure 31-1 USB System Block Diagram

Figure 31-2 illustrates the block diagram of USB 2.0 Host Controller.

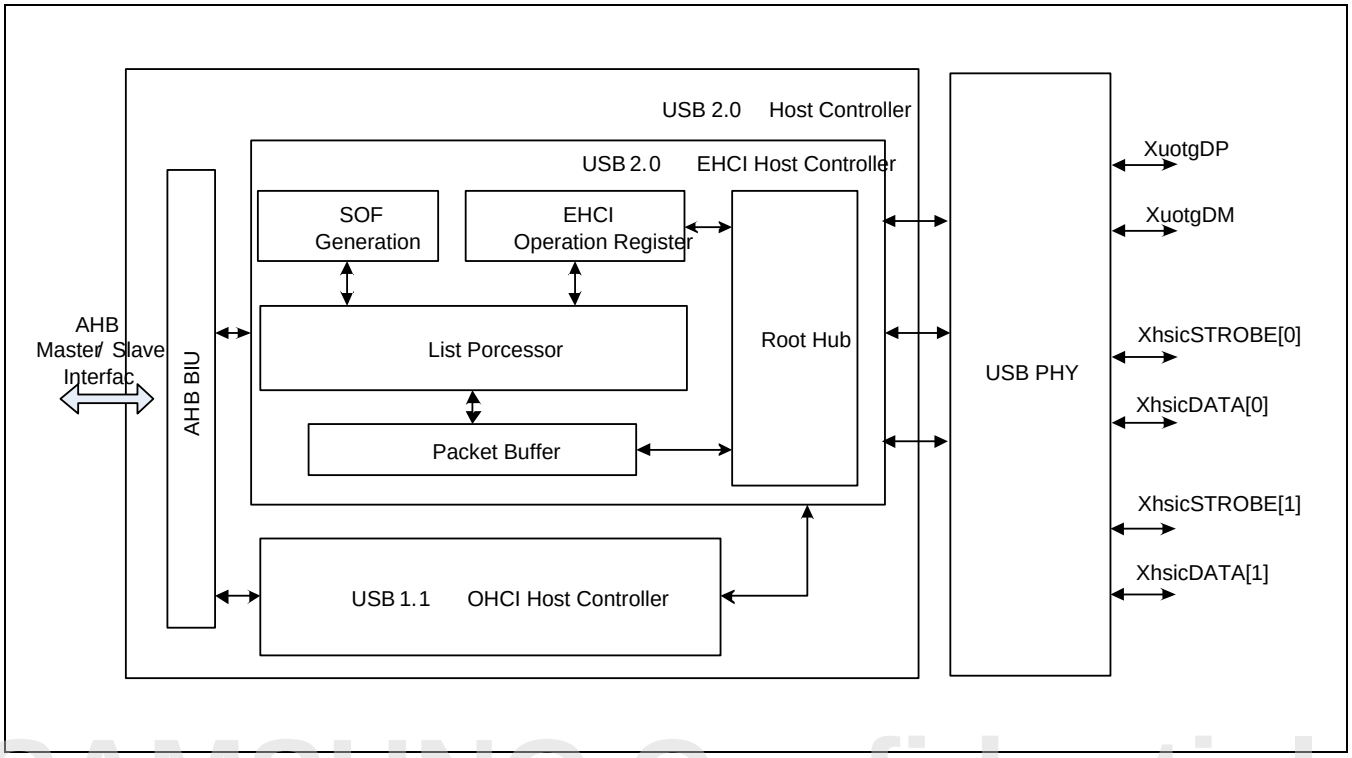


Figure 31-2 USB 2.0 Host Controller Block Diagram

31.3 I/O Description

Signal	I/O	Description	Pad	Type
XuotgDP	Input/Output	Data Plus Signal for Port0	XuotgDP	Dedicated
XuotgDM	Input/Output	Data Minus Signal for Port0	XuotgDM	Dedicated
XuotgREXT	Input/Output	Connection to the External 44.2 Ω (± 1 %) register	XuotgREXT	Dedicated
XhsicSTROBE[0]	Input/Output	Strobe signal for HSIC1	XhsicSTROBE[0]	Dedicated
XhsicDATA[0]	Input/Output	DDR data signal for HSIC1	XhsicDATA[0]	Dedicated
XhsicSTROBE[1]	Input/Output	Strobe signal for HSIC2	XhsicSTROBE[1]	Dedicated
XhsicDATA[1]	Input/Output	DDR data signal for HSIC2	XhsicDATA[1]	Dedicated
XuhostOVERCUR	Input	Port Overcurrent Indicator for Port0 When it asserts (active low), the corresponding port enters Disable state. This signal controls both EHCI and OHCI controller port state machines.	XuhostOVERCUR	Dedicated
XuhostPWREN	Output	Port Power enable for Port0 Use this control signal to switch the port power of each port. It implements the logic as described in Table 4-3 "Port Power Enable Control Rules" in the EHCI specification (Revision 1.0).	XuhostPWREN	Dedicated

31.4 HSIC Enable/Disable Setting

Following the HSIC specification (Version 1.0, Section 3.1.3), after a power-on reset, an HSIC IDLE condition is driven on the bus only if Port/Power bit in PORTSC register is set to 1.

31.5 Register Description

31.5.1 Register Map Summary

- Base Address: 0x1258_0000

Register	Offset	Description	Reset Value
Implemented Capability Registers for USB Host Controller			
HCCPBASE	0x0000	Specifies the capability and interface version number register.	0x0100_0010
HCSPARAMS	0x0004	Specifies the structural parameter.	0x0000_1313
HCCPARAMS	0x0008	Specifies the capability parameter.	0x0002_A016
Implemented Operational for USB Host Controller			
USBCMD	0x0010	Specifies the USB command.	0x0008_0B00
USBSTS	0x0014	Specifies the USB status.	0x0000_1000
USBINTR	0x0018	Enables the USB interrupt.	0x0000_0000
FRINDEX	0x001C	Specifies the USB frame index.	0x0000_0000
CTRLDSSEGMENT	0x0020	Specifies the 4G segment selector.	0x0000_0000
PERIODICLISTBASE	0x0024	Specifies the periodic frame list base address register.	0x0000_0000
ASYNCLISTADDR	0x0028	Specifies the asynchronous list address.	0x0000_0000
Implemented Auxiliary Registers for USB Host Controller			
CONFIGFLAG	0x0050	Specifies the configured flag register	0x0000_0000
Port 0 Status/Control	0x0054	Specifies the port status and control register	0x0000_2000
Port 1 Status/Control	0x0058	Specifies the port status and control register	0x0000_2000
Port 2 Status/Control	0x005C	Specifies the port status and control register	0x0000_2000
Implemented Specific Registers			
INSNREG00	0x0090	Specifies the programmable microframe base value.	0x0000_0000
INSNREG01	0x0094	Specifies the programmable packet buffer OUT/IN thresholds.	0x0040_0040
INSNREG02	0x0098	Specifies the programmable packet buffer depth.	0x0000_0100
INSNREG03	0x009C	Transfers break memory.	0x0000_0001
INSNREG04	0x00A0	INSNREG04	0x0000_0000
INSNREG05	0x00A4	INSNREG05	0x0000_1000
INSNREG06	0x00A8	AHB error status	0x0000_0000
INSNREG07	0x00AC	AHB master error address	0x0000_0000
OHCI Registers for USB Host Controller			
HcRevision	0x0000	Specifies the USB host controller revision register.	0x0000_0110
HcControl	0x0004	Specifies the USB host controller control register.	0x0000_0000
HcCommonStatus	0x0008	Specifies the USB host controller command status register.	0x0000_0000
HcInterruptStatus	0x000C	Specifies the USB host controller interrupt status	0x0000_0000

Register	Offset	Description	Reset Value
		register.	
HcInterruptEnable	0x0010	Specifies the USB host controller interrupt enable register.	0x0000_0000
HcInterruptDisable	0x0014	Specifies the USB host controller interrupt disable register.	0x0000_0000
HcHCCA	0x0018	Specifies the USB host controller HCCA register.	0x0000_0000
HcPeriodCuttentED	0x001C	Specifies the USB host controller period current ED register.	0x0000_0000
HcControlHeadED	0x0020	Specifies the USB host controller control head ED register.	0x0000_0000
HcControlCurrentED	0x0024	Specifies the USB host controller control current ED register.	0x0000_0000
HcBulkHeadED	0x0028	Specifies the USB host controller bulk head ED register.	0x0000_0000
HcBulkCurrentED	0x002C	Specifies the USB host controller bulk current ED register.	0x0000_0000
HcDoneHead	0x0030	Specifies the USB host controller done head register.	0x0000_0000
HcRmInterval	0x0034	Specifies the USB host controller fminterval register.	0x0000_2EDF
HcFmRemaining	0x0038	Specifies the USB host controller frame remaining register.	0x0000_0000
HcFmNumber	0x003C	Specifies the USB host controller frame number register.	0x0000_0000
HcPeriodicStart	0x0040	Specifies the USB host controller periodic start register.	0x0000_0000
HcLSThreshold	0x0044	Specifies the USB host controller low-speed threshold register.	0x0000_0628
HcRhDescriptorA	0x0048	Specifies the USB host controller root hub descriptor a register.	0x0200_0003
HcRhDescriptorB	0x004C	Specifies the USB host controller root hub descriptor b register.	0x0000_0000
HcRhStatus	0x0050	Specifies the USB host controller root hub status register.	0x0000_0000
HcRhPortStatus	0x0054	Specifies the USB host controller root hub port status register.	0x0000_0000

NOTE: The USB host controller in Exynos 4412 SCP complies with both EHCI Revision 1.1a and OHCI Revision 1.0 specification. Refer to EHCI Revision1.1a and OHCI Revision 1.0 specification for more information.

31.5.2 Implemented Specific Registers

Specific registers allow you to program configurable registers such as the Packet Buffer Depth, Break Memory Transfer, Frame Length, UTMI Control, and Status Registers Access.

31.5.2.1 INSNREG00

- Base Address: 0x1258_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	6'b0
Ena_incr16	[25]	RW	Enables the AHB master interface to utilize burst INCR16 when appropriate. 1'b0 = Do not use INCR16:use other enabled INCRX bursts or unspecified length burst INCR 1'b1 = Use INCR16 when appropriate	1'b0
Ena_incr8	[24]	RW	Enables the AHB master interface to utilize burst INCR8 when appropriate. 1'b0 = Do not use INCR8:use other enabled INCRX bursts or unspecified length burst INCR 1'b1 = Use INCR8 when appropriate	1'b0
Ena_incr4	[23]	RW	Enables the AHB master interface to utilize burst INCR4 when appropriate. 1'b0 = Do not use INCR4:use other enabled INCRX bursts or unspecified length burst INCR 1'b1 = Use INCR4 when appropriate	1'b0
Ena_incrx_align	[22]	RW	Forces AHB master to start INCR4/8/16 bursts only on burst boundaries. AHB requires that double word width burst to be addressed must be aligned only based on the double word boundary. 1'b0 = Normal AHB operation, start bursts on any double word boundary 1'b1 = Starts INCRX burst only on burst x-aligned addresses	1'b0
App_start_clk	[21]	RW	When the OHCI clocks are suspended, the system has to assert this signal to start the clocks (12 and 48 MHz). This should be disserted after the clocks are started and before the host is suspended again (Host is suspended means HCFS = SUSPEND or all the OHCI ports are suspended).	1'b0
Ohci_susp_lgcy	[20]	RW	1'b0 = If OHCI owns the USB port, the utmi_suspend_o_n asserts to zero in these cases. All of the OHCI ports are suspended or the OHCI is in global suspend state (HCFS = USBSUSOPEND). The utmi_susend_o_n deasserts to one in this case. Any of the OHCI ports are not suspended and OHCI is not in global suspend.	1'b0

Name	Bit	Type	Description	Reset Value
			1'b1 = If OHCI owns the USB port, the signal utmi_suspend_o_n reflects the status of the USB port(suspended or not suspended)	
RSVD	[19:14]	–	Reserved	6'b0
Microframe_base_ value	[13:1]	RW	Changes the microframe length value (default is microframe SOF = 125 us) to reduce the simulation time. This value is used as 1-microframe counter with word interface (16 bits).	13'b0
Ena_microframe_ length_value_field	[0]	RW	Writing 1’b1 enables the microframe length value field of this register.	1'b0

31.5.2.2 INSNREG01

- Base Address: 0x1258_0000
- Address = Base Address + 0x0094, Reset Value = 0x0040_0040

Name	Bit	Type	Description	Reset Value
Packet_buffer_ thresholds	[31:0]	RW	[31:16] OUT Threshold [15:0] IN Threshold Specifies the number of DWORDs (32-bit entries). You can use OUT threshold to start the USB transfer as soon as it fetches the OUT threshold data from system memory. You can also use this to disconnect the data fetch if the threshold amount of space is not available in the Packet Buffer. You can use IN threshold to start the memory transfer as soon as the IN threshold data is available in the Packet Buffer. You can also use this to disconnect the data write if the threshold amount of data is not available in the Packet BufferFor INCRX configurations, the minimum threshold amount that you can program is the highest possible INCRX burst value. The minimum OUT and IN threshold amount that you can program through INSN registers is 16 bytes. OUT and IN threshold values can be equal to the packet buffer depth. The default value of thresholds is 256 bytes IN and OUT thresholds.	0x0040_0040

31.5.2.3 INSNREG02

- Base Address: 0x1258_0000
- Address = Base Address + 0x0098, Reset Value = 0x0000_0100

Name	Bit	Type	Description	Reset Value
Packet_buffer_depth	[31:0]	RW	Programmable Packet Buffer Depth. The value specified here is the number of DWORDs (32-bit interface)	0x00000100

31.5.2.4 INSNREG03

- Base Address: 0x1258_0000
- Address = Base Address + 0x009C, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	19'b0
Tx_Tx_turnaround_ delay_add	[12:10]	RW	Tx-Tx Turnaround Delay Add on This field specifies the extra delays in phy_clks to be added to the "Transmit to Transmit turnaround delay" value maintained in the core. The default value of this register field is 0.	3'b0
Periodic_frame_list_ fetch	[9]	RW	Periodic Frame List Fetch Setting this bit will force the host controller to fetch the periodic frame list in every microframe of a frame.	1'b0
Time_available_offset	[8:1]	RW	Time-Available Offset. This value indicates the additional number of bytes to be accommodated for the time-available calculation.	8'b0
Break_memory_ transfer	[0]	RW	Specifies Break Memory Transfer You can use this in conjunction with INSNREG01 to enable/disable breaking memory transactions into chunks after it reaches the OUT/IN threshold value. 1’b0 = Disables this function 1’b1 = Enables this function	1'b1

31.5.2.5 INSNREG04

- Base Address: 0x1258_0000
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:0]	RW	Reserved	0x00000000

31.5.2.6 INSNREG05

- Base Address: 0x1258_0000
- Address = Base Address + 0x00A4, Reset Value = 0x0000_1000

Name	Bit	Type	Description	Reset Value
RSVD	[31:0]	RW	Reserved	0x00001000

31.5.2.7 INSNREG06

- Base Address: 0x1258_0000
- Address = Base Address + 0x00A8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
AHB_error_captured	[31]	RW	AHB Error Captured Indicates that an AHB error was encountered and values were captured. To clear this field the application must write a 0 to it.	1'b0
RSVD	[30:12]	–	Reserved	19'b0
HBURST_value_of_control_phase	[11:9]	RW	HBURST value of the control phase at which the AHB error occurred.	3'b0
Number_of_beats_expected	[8:4]	RW	Expects number of beats in the burst at which the AHB error occurs. Valid values are 0 to 6	5'b0
Number_of_successfully_completed_beats	[3:0]	RW	Number of successfully-completed beats in the current burst before the AHB error occurred.	4'b0

31.5.2.8 INSNREG07

- Base Address: 0x1258_0000
- Address = Base Address + 0x00AC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
AHB_master_error_address	[31:0]	RW	AHB Master Error Address AHB address of the control phase at which the AHB error occurred.	0x00000000

32 USB2.0 Device

32.1 Overview of USB2.0 Device

Samsung USB 2.0 Controller is designed to aid the rapid implementation of the USB 2.0 peripheral device. It supports both high-speed (HS, 480 Mbps) and full-speed (FS, 12 Mbps) transfers. Using the standard UTMI interface and AHB interface, the USB 2.0 Controller can support up to 16 Endpoints (including Endpoint 0) with programmable Interrupt, Bulk and Isochronous transfer mode.

32.2 Key Features of USB2.0 Device

The USB2.0 device features include:

- Complies to USB 2.0 Specification (Revision 1.0a)
- Operates in High-speed (480 Mbps) and Full-speed (12 Mbps)
- Supports UTMI + Level 3 interface (Revision 1.0)
- Supports only 32-bit data on the AHB
- 1 Control Endpoint 0 for control transfer
- 15 Device Mode programmable Endpoints
  - Programmable endpoint type: Bulk, Isochronous, or Interrupt
  - Programmable IN/OUT direction
- Supports packet-based, dynamic FIFO memory allocation of 7936 depths (35-bit width)

32.3 Block Diagram of USB2.0 Device

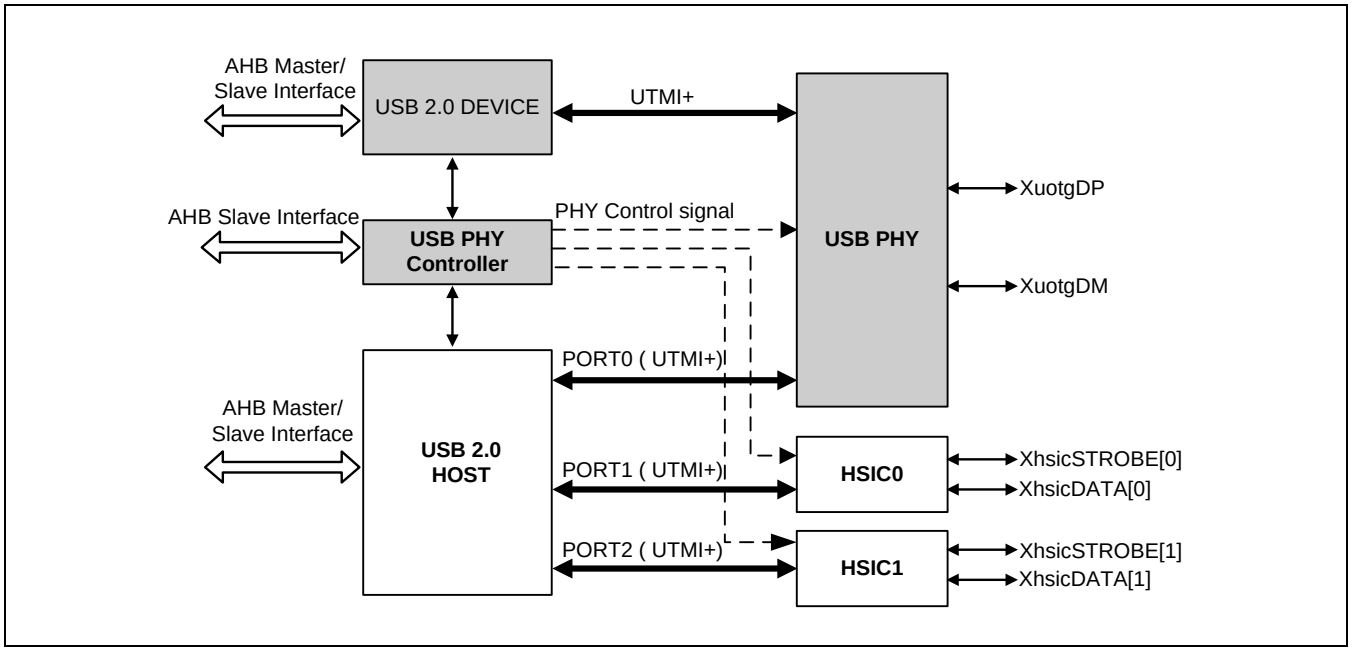


Figure 32-1 System Level Block Diagram

USB 2.0 device controller is composed of two independent blocks, namely, USB 2.0 device and USB PHY Controller. Each of these blocks has an AHB Slave interface that provides the microcontroller with read and write access to the Control and Status Registers (CSRs). The device Link has an AHB Master to enable the link to transfer data on the AHB.

32.4 Modes of Operation

The end user application operates the Link in either DMA or Slave mode. It cannot operate the USB device controller using DMA and Slave modes simultaneously.

32.4.1 DMA Mode

USB device uses the AHB Master interface to transmit packet data to be fetched (AHB to USB) and receive data update (USB to AHB). The AHB master uses programmed DMA address (DIEPDMAn/DOEPDMAn register) to access the data buffers.

32.4.2 Slave Mode

USB device can operate in either transaction-level or pipelined transaction-level. The application handles one data packet at a time per channel/endpoint in transaction-level operations. In pipelined transaction-level operation, the application programs the device to perform multiple transactions. The advantage of pipelined operation is that the application is not interrupted based on packet.

32.5 Power Management Unit Setting

A register in Power Management Unit has to be set for USB to work appropriately. For more information, refer to Power Management Unit.

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	31'b0
ENABLE	[0]	RW	Enables USB OTG PHY (USB PHY 0) 0 = Disables 1 = Enables This bit must be set before USB OTG PHY is used.	1'b0

The USBOTG_PHY_CONTROL register (Based on the address 0x1002_0704) is guided to be set differently depending on the following system operation mode:

- **Normal Mode**
  - Reset Value of ENABLE is 1'b0. To start USB transaction set this value to 1'b1.
  - In Normal Mode, power to USB PHY is switched off if USB device function is not used.

- **Stop/Deep Stop/Sleep Mode**
  - In Stop/Deep Stop/Sleep Mode operation modes, USB PHY power is switched off.
  - Therefore, to prevent unwanted leakage current, ENABLE must be set to 1'b0 before entering these modes.

32.6 Register Map

32.6.1 Overview of Register Map

To control and observe the device PHY, access the USB PHY control registers based on the address 125B_0000h.

The device link Core registers is based on the address 1248_0000h, which is classified as follows:

- Core Global Registers
- Device Mode Registers
  - Device Global Registers
  - Device Endpoint-Specific Registers

32.6.2 Device Link CSR Memory Map

Figure 32-2 shows the device link CSR address map. All registers are implemented in the AHB Clock domain.

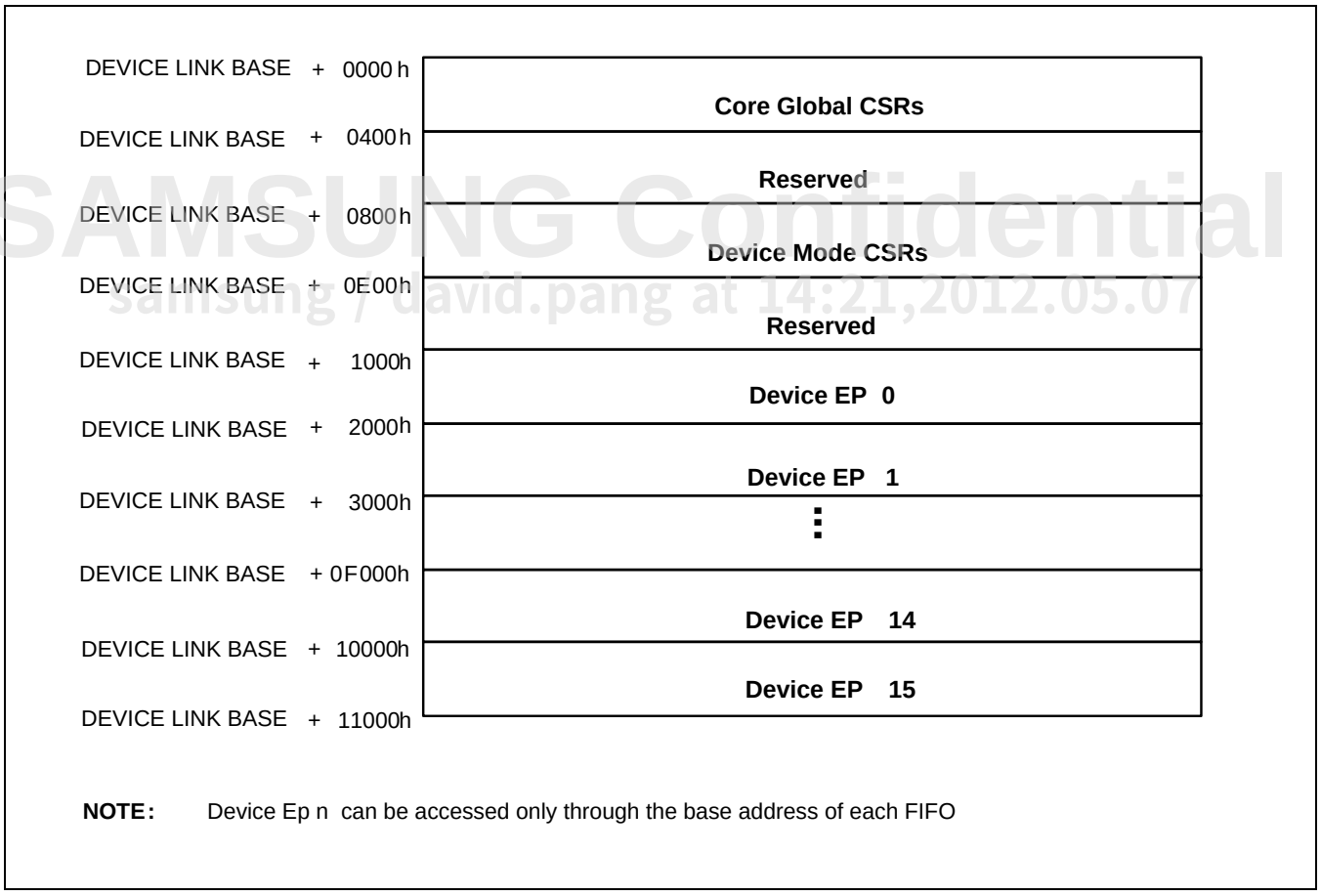


Figure 32-2 Device Link CSR Memory Map

32.6.3 Device FIFO Address Mapping

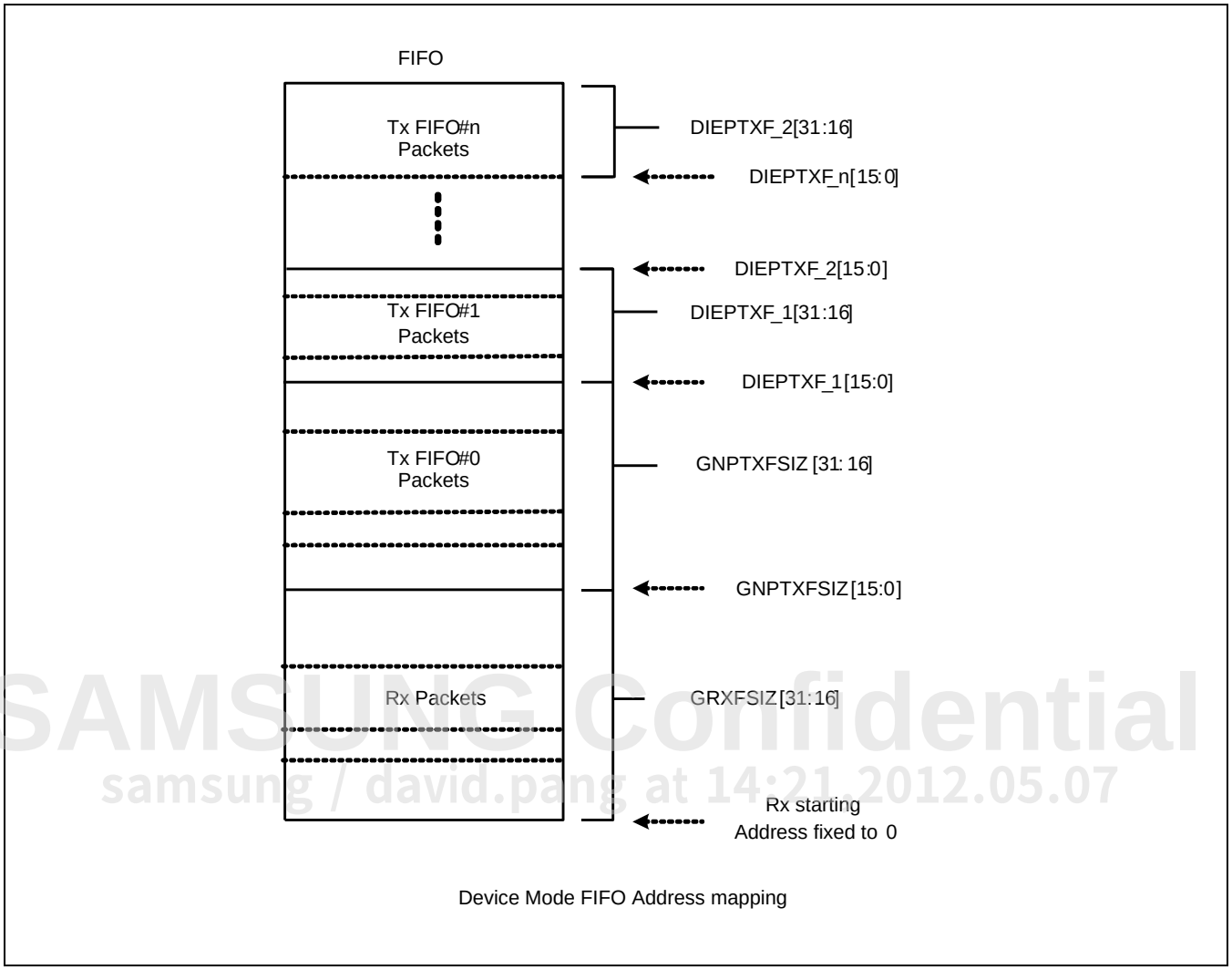


Figure 32-3 Device FIFO Mapping

Figure 32-3 shows the device FIFO Address Mapping. The following registers must be programmed as follows;

In device Mode

- RXFSIZ[31:16]: OTG_RX _DFIFO_ DEPTH
- NPTXFSIZ[15:0]: OTG_RX_ DFIFO_ DEPTH
- NPTXFSIZ[31:16]: OTG_TX_DINEP_DFIFO_DEPTH_0
- DIEPTXF_1[15:0]: OTG_RX_DFIFO_DEPTH + OTG_TX_DINEP_DFIFP_DEPTH_0
- DIEPTXF_1[31:16]: OTG_TX_DINEP_DFIFO_DEPTH_1
- DEIPTXF_2[15:0]: DIEPTXF_1[15:0] + OTG_TX_DINEP_DFIFO_DEPTH_1
- DIEPTXF_2[31:16]: OTG_TX_DINEP_DFIFO_DEPTH_2

NOTE: When the device is operating in non Scatter Gather Internal DMA mode, the last locations of the SPRAM are used to store the DMAADDR values of each Endpoint (1 Location per Endpoint). When the device is operating in Scatter Gather, then the last locations of the SPRAM store the Base Descriptor address, Current Descriptor address, Current Buffer address, and status word (DWORD-32 bits information for each endpoint direction (4 locations per Endpoint .If an endpoint is bidirectional then 4 will be used for IN and another 4 for OUT).

32.6.4 Application Access to the CSRs

The Access column of each register description that follows specifies how the application and the core access the register fields of the CSRs. The following conventions are used.

Read Only	R	The application has permission to read the Register field. Writes to read-only fields have no effect.
Write Only	W	The application has permission to write in the Register field.
READ and Write	RW	The application has permission to read and writes in the Register field. The application sets this field by writing 1'b1 and clears it by writing 1'b0.
Read, Write, and Self Clear	RW_SC	Register field is read and written by the application (Read and Write), and is cleared to 1'b0 by the core (Self Clear). The conditions under which the core clears this field are explained in detail in the field's description.
Read, Write, Self Set, and Self Clear	RW_SS_SC	Register field is read and written by the application (Read and Write), set to 1'b1 by the core on certain USB events (Self Set), and cleared to 1'b0 by the core (Self Clear).
Read, Self set, and Write Clear	R/SS_WC	Register field is read by the application (Read), set to 1'b1 by the core on certain internal or USB or AHB event (Self Set), and cleared to 1'b0 by the application with a register write of 1'b1 (Write Clear). A register write of 1'b0 has no effect on this field.
Read, Write Set, and Self Clear	RWS_SC	Register field is read by the application (Read), set to 1'b1 by the application with a register write of 1'b1 (Write Set), and is cleared to 1'b0 by the core (Self Clear). The application cannot clear this type of field, and a register write of 1'b0 to this bit has no effect on this field.
Read, Self set, and Self Clear or Write Clear	R/SS_SC_WC	Register field is read by the application (Read), set to 1'b1 by the core on certain internal or USB or AHB events (Self Set), and cleared to 1'b0 either by the core itself (Self Clear) or by the application with a register write of 1'b1 (Write Clear). A register write of 1'b0 to this bit has no effect on this field.

32.7 I/O Description

Signal	I/O	Description	Pad.	Type.
XuotgDP	Input/Output	Data Plus Signal from the USB Cable	XuotgDP	Dedicated
XuotgDM	Input/Output	Data Minus Signal from the USB Cable	XuotgDM	Dedicated
XusbXTI	Input	Crystal Oscillator XI	XusbXTI	Dedicated
XusbXTO	Output	Crystal Oscillator XO	XusbXTO	Dedicated
XuotgREXT	Input/Output	Connection to the External 44.2 Ω (± 1 %) register	XuotgREXT	Dedicated
XuotgVBUS	Input/Output	USB mini-Receptacle VBUS This is the USB power supply pin. High: not connected to USB HOST Low: connected to USB HOST	XuotgVBUS	Dedicated
XuotgID	Input	USB mini-Receptacle Identifier Leave this pin floating (> 100 kΩ)	XuotgID	Dedicated
XuotgDRVVBUS	Output	Off-Chip Charge Pump Enable Not Available.	XuotgDRVVBUS	Dedicated

32.8 Register Description

32.8.1 Register Map Summary

- Base Address: 0x125B_0000

Register	Offset	Description	Reset Value
USB PHY Control Register			
UPHYPWR	0x0000	Specifies the USB PHY power control register	0x0000_7FF9
UPHYCLK	0x0004	Specifies the USB PHY clock control register	0x0000_9005
URSTCON	0x0008	Specifies the USB reset control register	0x0000_0079
UPHYTUNE1	0x0020	Specifies the USB HOST PHY tune register	0x0028_19B3
UPHYTUNE0	0x0024	Specifies the USB DEVICE PHY tune register	0x0028_19B3

- Base Address: 0x1248_0000

Register	Offset	Description	Reset Value
Device LINK Core Register			
GOTGCTL	0x0000	Specifies the OTG control and status register	0x0001_0000
GOTGINT	0x0004	Specifies the OTG interrupt register	0x0000_0000
GAHBCFG	0x0008	Specifies the core AHB configuration register	0x0000_0000
GUSBCFG	0x000C	Specifies the core USB configuration register	0x0000_1408
GRSTCTL	0x0010	Specifies the core reset register	0x8000_0000
GINTSTS	0x0014	Specifies the core interrupt register	0x0400_0020
GINTMSK	0x0018	Specifies the core interrupt mask register	0x0000_0000
GRXSTSR	0x001C	Specifies the receive status debug read register	Undefined
GRXSTSP	0x0020	Specifies the receive status read/pop register	Undefined
GRXFSIZ	0x0024	Specifies the receive FIFO size register	0x0000_1F00
GNPTXFSIZ	0x0028	Specifies the non-periodic transmit FIFO size register	0x1F00_1F00
GNPTXSTS	0x002C	Specifies the non-periodic transmit FIFO/Queue status register	0x0008_1F00
GLPMCFG	0x0054	Specifies the core LPM configuration register	0x0000_0000
HPTXFSIZ	0x0100	Specifies the host periodic transmit FIFO size register	0x0300_5A00
DIEPTXF1	0x0104	Specifies the device in endpoint transmit FIFO-1 size register	0x0300_2200
DIEPTXF2	0x0108	Specifies the device in endpoint transmit FIFO-2 size register	0x0300_2500
DIEPTXF3	0x010C	Specifies the device in endpoint transmit FIFO-3 size register	0x0300_2800
DIEPTXF4	0x0110	Specifies the device in endpoint transmit FIFO-4 size register	0x0300_2B00
DIEPTXF5	0x0114	Specifies the device in endpoint transmit FIFO-5 Size Register	0x0300_2E00
DIEPTXF6	0x0118	Specifies the device in endpoint transmit FIFO-6 size register	0x0300_3100
DIEPTXF7	0x011C	Specifies the device in endpoint transmit FIFO-7 size register	0x0300_3400
DIEPTXF8	0x0120	Specifies the device in endpoint transmit FIFO-8 size register	0x0300_3700

Register	Offset	Description	Reset Value
DIEPTXF9	0x0124	Specifies the device in endpoint transmit FIFO-9 size register	0x0300_3A00
DIEPTXF10	0x0128	Specifies the device in endpoint transmit FIFO-10 size register	0x0300_3D00
DIEPTXF11	0x012C	Specifies the device in endpoint transmit FIFO-11 size register	0x0300_4000
DIEPTXF12	0x0130	Specifies the device in endpoint transmit FIFO-12 size register	0x0300_4300
DIEPTXF13	0x0134	Specifies the device in endpoint transmit FIFO-13 size register	0x0300_4600
DIEPTXF14	0x0138	Specifies the device in endpoint transmit FIFO-14 size register	0x0300_4900
DIEPTXF15	0x013C	Specifies the device in endpoint transmit FIFO-15 size register	0x0300_4C00
Device Mode Register (Device Global Register)			
DCFG	0x0800	Specifies the device configuration register	0x0820_0000
DCTL	0x0804	Specifies the device control register	0x0000_0000
DSTS	0x0808	Specifies the device status register	0x0000_0002
DIEPMSK	0x0810	Specifies the device in endpoint common interrupt mask register	0x0000_0000
DOEPMSK	0x0814	Specifies the device out endpoint common interrupt mask register	0x0000_0000
DAINT	0x0818	Specifies the device all endpoints interrupt register	0x0000_0000
DAINTMSK	0x081C	Specifies the device all endpoints interrupt mask register	0x0000_0000
DVBUSDIS	0x0828	Specifies the device VBUS discharge time register	0x0000_0B8F
DVBUSPULSE	0x082C	Specifies the device VBUS pulsing time register	0x0000_02C6
DTHRCTL	0x0830	Specifies the device threshold control register	0x0C10_0020
DIEPEMPMSK	0x0834	Specifies the device in endpoint FIFO EMPTY interrupt mask register	0x0000_0000
Device Mode Register (Device Logical IN Endpoint-Specific Register)			
DIEPCTL0	0x0900	Specifies the device control in endpoint 0 control register	0x0000_8000
DIEPINT0	0x0908	Specifies the device in endpoint 0 interrupt register	0x0000_0000
DIEPTSIZ0	0x0910	Specifies the device in endpoint 0 transfer size register	0x0000_0000
DIEPDMA0	0x0914	Specifies the device in endpoint 0 DMA address register	0x0000_0000
DTXFSTS0	0x0918	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB0	0x091C	Specifies the device in endpoint 0 DMA buffer address register	0x0000_0000
DIEPCTL1	0x0920	Specifies the device control in endpoint 1 control register	0x0000_0000
DIEPINT1	0x0928	Specifies the device in endpoint 1 interrupt register	0x0000_0080
DIEPTSIZ1	0x0930	Specifies the device in endpoint 1 transfer size register	0x0000_0000
DIEPDMA1	0x0934	Specifies the device in endpoint 1 DMA address register	0x0000_0000
DTXFSTS1	0x0938	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB1	0x093C	Specifies the device in endpoint 1 DMA buffer address register	0x0000_0000
DIEPCTL2	0x0940	Specifies the device control in endpoint 2 control register	0x0000_0000
DIEPINT2	0x0948	Specifies the device in endpoint 2 interrupt register	0x0000_0080

Register	Offset	Description	Reset Value
DIEPTSIZ2	0x0950	Specifies the device in endpoint 2 transfer size register	0x0000_0000
DIEPDMA2	0x0954	Specifies the device in endpoint 2 DMA address register	0x0000_0000
DTXFSTS2	0x0958	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB2	0x095C	Specifies the device in endpoint 2 DMA buffer address register	0x0000_0000
DIEPCTL3	0x0960	Specifies the device control in endpoint 3 control register	0x0000_0000
DIEPINT3	0x0968	Specifies the device in endpoint 3 interrupt register	0x0000_0080
DIEPTSIZ3	0x0970	Specifies the device in endpoint 3 transfer size register	0x0000_0000
DIEPDMA3	0x0974	Specifies the device in endpoint 3 DMA address register	0x0000_0000
DTXFSTS3	0x0978	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB3	0x097C	Specifies the device in endpoint 3 DMA buffer address register	0x0000_0000
DIEPCTL4	0x0980	Specifies the device control in endpoint 4 control register	0x0000_0000
DIEPINT4	0x0988	Specifies the device in endpoint 4 interrupt register	0x0000_0080
DIEPTSIZ4	0x0990	Specifies the device in endpoint 4 transfer size register	0x0000_0000
DIEPDMA4	0x0994	Specifies the device in endpoint 4 DMA address register	0x0000_0000
DTXFSTS4	0x0998	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB4	0x099C	Specifies the device in endpoint 4 DMA buffer address register	0x0000_0000
DIEPCTL5	0x09A0	Specifies the device control in endpoint 5 control register	0x0000_0000
DIEPINT5	0x09A8	Specifies the device in endpoint 5 interrupt register	0x0000_0080
DIEPTSIZ5	0x09B0	Specifies the device in endpoint 5 transfer size register	0x0000_0000
DIEPDMA5	0x09B4	specifies the device in endpoint 5 DMA address register	0x0000_0000
DTXFSTS5	0x09B8	specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB5	0x09BC	Specifies the device in endpoint 5 DMA buffer address register	0x0000_0000
DIEPCTL6	0x09C0	Specifies the device control in endpoint 6 control register	0x0000_0000
DIEPINT6	0x09C8	Specifies the device in endpoint 6 interrupt register	0x0000_0080
DIEPTSIZ6	0x09D0	Specifies the device in endpoint 6 transfer size register	0x0000_0000
DIEPDMA6	0x09D4	Specifies the device in endpoint 6 DMA address register	0x0000_0000
DTXFSTS6	0x09D8	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB6	0x09DC	Specifies the device in endpoint 6 DMA buffer address register	0x0000_0000
DIEPCTL7	0x09E0	Specifies the device control in endpoint 7 control register	0x0000_0000
DIEPINT7	0x09E8	Specifies the device in endpoint 7 interrupt register	0x0000_0080
DIEPTSIZ7	0x09F0	Specifies the device in endpoint 7 transfer size register	0x0000_0000
DIEPDMA7	0x09F4	Specifies the device in endpoint 7 dma address register	0x0000_0000
DTXFSTS7	0x09F8	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB7	0x09FC	Specifies the device in endpoint 7 DMA buffer address register	0x0000_0000
DIEPCTL8	0x0A00	Specifies the device control in endpoint 8 control register	0x0000_0000
DIEPINT8	0x0A08	Specifies the device in endpoint 8 interrupt register	0x0000_0080
DIEPTSIZ8	0x0A10	Specifies the device in endpoint 8 transfer size register	0x0000_0000

Register	Offset	Description	Reset Value
DIEPDMA8	0x0A14	Specifies the device in endpoint 8 DMA address register	0x0000_0000
DTXFSTS8	0x0A18	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB8	0x0A1C	Specifies the device in endpoint 8 DMA buffer address register	0x0000_0000
DIEPCTL9	0x0A20	Specifies the device control in endpoint 9 control register	0x0000_0000
DIEPINT9	0x0A28	Specifies the device in endpoint 9 interrupt register	0x0000_0080
DIEPTSIZ9	0x0A30	Specifies the device in endpoint 9 transfer size register	0x0000_0000
DIEPDMA9	0x0A34	Specifies the device in endpoint 9 DMA address register	0x0000_0000
DTXFSTS9	0x0A38	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB9	0x0A3C	Specifies the device in endpoint 9 DMA buffer address register	0x0000_0000
DIEPCTL10	0x0A40	Specifies the device control in endpoint 10 control register	0x0000_0000
DIEPINT10	0x0A48	Specifies the device in endpoint 10 interrupt register	0x0000_0080
DIEPTSIZ10	0x0A50	Specifies the device in endpoint 10 transfer size register	0x0000_0000
DIEPDMA10	0x0A54	Specifies the device in endpoint 10 DMA address register	0x0000_0000
DTXFSTS10	0x0A58	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB10	0x0A5C	Specifies the device in endpoint 10 DMA buffer address register	0x0000_0000
DIEPCTL11	0x0A60	Specifies the device control in endpoint 11 control register	0x0000_0000
DIEPINT11	0x0A68	Specifies the device in endpoint 11 interrupt register	0x0000_0080
DIEPTSIZ11	0x0A70	Specifies the device in endpoint 11 transfer size register	0x0000_0000
DIEPDMA11	0x0A74	Specifies the device in endpoint 11 DMA address register	0x0000_0000
DTXFSTS11	0x0A78	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB11	0x0A7C	Specifies the device in endpoint 11 DMA buffer address register	0x0000_0000
DIEPCTL12	0x0A80	Specifies the device control in endpoint 12 control register	0x0000_0000
DIEPINT12	0x0A88	Specifies the device in endpoint 12 interrupt register	0x0000_0080
DIEPTSIZ12	0x0A90	Specifies the device in endpoint 12 transfer size register	0x0000_0000
DIEPDMA12	0x0A94	Specifies the device in endpoint 12 DMA address register	0x0000_0000
DTXFSTS12	0x0A98	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB12	0x0A9C	Specifies the device in endpoint 12 DMA buffer address register	0x0000_0000
DIEPCTL13	0x0AA0	Specifies the device control in endpoint 13 control register	0x0000_0000
DIEPINT13	0x0AA8	Specifies the device in endpoint 13 interrupt register	0x0000_0080
DIEPTSIZ13	0x0AB0	Specifies the device in endpoint 13 transfer size register	0x0000_0000
DIEPDMA13	0x0AB4	Specifies the device in endpoint 13 DMA address register	0x0000_0000
DTXFSTS13	0x0AB8	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB13	0x0ABC	Specifies the device in endpoint 13 DMA buffer address register	0x0000_0000
DIEPCTL14	0x0AC0	Specifies the device control in endpoint 14 control register	0x0000_0000

Register	Offset	Description	Reset Value
DIEPINT14	0x0AC8	Specifies the device in endpoint 14 interrupt register	0x0000_0080
DIEPTSIZ14	0x0AD0	Specifies the device in endpoint 14 transfer size register	0x0000_0000
DIEPDMA14	0x0AD4	Specifies the device in endpoint 14 DMA address register	0x0000_0000
DTXFSTS14	0x0AD8	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB14	0x0ADC	Specifies the device in endpoint 14 DMA buffer address register	0x0000_0000
DIEPCTL15	0x0AE0	Specifies the device control in endpoint 15 control register	0x0000_0000
DIEPINT15	0x0AE8	Specifies the device in endpoint 15 interrupt register	0x0000_0080
DIEPTSIZ15	0x0AF0	Specifies the device in endpoint 15 transfer size register	0x0000_0000
DIEPDMA15	0x0AF4	Specifies the device in endpoint 15 DMA address register	0x0000_0000
DTXFSTS15	0x0AF8	Specifies the device in endpoint transmit FIFO status register	0x0000_0100
DIEPDMAB15	0x0AFC	Specifies the device in endpoint 15 DMA buffer address register	0x0000_0000
Device Mode Register (Device Logical OUT Endpoint-Specific Register)			
DOEPCTL0	0x0B00	Specifies the device control out endpoint 0 control register	0x0000_8000
DOEPINT0	0x0B08	Specifies the device out endpoint 0 interrupt register	0x0000_0000
DOEPTSIZ0	0x0B10	Specifies the device out endpoint 0 transfer size register	0x0000_0000
DOEPDMA0	0x0B14	Specifies the device out endpoint 0 DMA address register	0x0000_0000
DOEPDMAB0	0x0B1C	Specifies the device out endpoint 0 DMA buffer address register	0x0000_0000
DOEPCTL1	0x0B20	Specifies the device control out endpoint 1 control register	0x0000_0000
DOEPINT1	0x0B28	Specifies the device out endpoint 1 interrupt register	0x0000_0000
DOEPTSIZ1	0x0B30	Specifies the device out endpoint 1 transfer size register	0x0000_0000
DOEPDMA1	0x0B34	Specifies the device out endpoint 1 DMA address register	0x0000_0000
DOEPDMAB1	0x0B3C	Specifies the device out endpoint 1 DMA buffer address register	0x0000_0000
DOEPCTL2	0x0B40	Specifies the device control out endpoint 2 control register	0x0000_0000
DOEPINT2	0x0B48	Specifies the device out endpoint 2 interrupt register	0x0000_0000
DOEPTSIZ2	0x0B50	Specifies the device out endpoint 2 transfer size register	0x0000_0000
DOEPDMA2	0x0B54	Specifies the device out endpoint 2 DMA address register	0x0000_0000
DOEPDMAB2	0x0B5C	Specifies the device out endpoint 2 DMA buffer address register	0x0000_0000
DOEPCTL3	0x0B60	Specifies the device control out endpoint 3 control register	0x0000_0000
DOEPINT3	0x0B68	Specifies the device out endpoint 3 interrupt register	0x0000_0000
DOEPTSIZ3	0x0B70	Specifies the device out endpoint 3 transfer size register	0x0000_0000
DOEPDMA3	0x0B74	Specifies the device out endpoint 3 DMA address register	0x0000_0000
DOEPDMAB3	0x0B7C	Specifies the device out endpoint 3 DMA buffer address register	0x0000_0000

Register	Offset	Description	Reset Value
DOEPCTL4	0x0B80	Specifies the device control out endpoint 4 control register	0x0000_0000
DOEPINT4	0x0B88	Specifies the device out endpoint 4 interrupt register	0x0000_0000
DOEPTSIZ4	0x0B90	Specifies the device out endpoint 4 transfer size register	0x0000_0000
DOEPDMA4	0x0B94	Specifies the device out endpoint 4 DMA address register	0x0000_0000
DOEPDMAB4	0x0B9C	Specifies the device out endpoint 4 DMA buffer address register	0x0000_0000
DOEPCTL5	0x0BA0	Specifies the device control out endpoint 5 control register	0x0000_0000
DOEPINT5	0x0BA8	Specifies the device out endpoint 5 interrupt register	0x0000_0000
DOEPTSIZ5	0x0BB0	Specifies the device out endpoint 5 transfer size register	0x0000_0000
DOEPDMA5	0x0BB4	Specifies the device out endpoint 5 DMA address register	0x0000_0000
DOEPDMAB5	0x0BBC	Specifies the device out endpoint 5 DMA buffer address register	0x0000_0000
DOEPCTL6	0x0BC0	Specifies the device control out endpoint 6 control register	0x0000_0000
DOEPINT6	0x0BC8	Specifies the device out endpoint 6 interrupt register	0x0000_0000
DOEPTSIZ6	0x0BD0	Specifies the device out endpoint 6 transfer size register	0x0000_0000
DOEPDMA6	0x0BD4	Specifies the device out endpoint 6 DMA address register	0x0000_0000
DOEPDMAB6	0x0BDC	Specifies the device out endpoint 6 DMA buffer address register	0x0000_0000
DOEPCTL7	0x0BE0	Specifies the device control out endpoint 7 control register	0x0000_0000
DOEPINT7	0x0BE8	Specifies the device out endpoint 7 interrupt register	0x0000_0000
DOEPTSIZ7	0x0BF0	Specifies the device out endpoint 7 transfer size register	0x0000_0000
DOEPDMA7	0x0BF4	Specifies the device out endpoint 7 DMA address register	0x0000_0000
DOEPDMAB7	0x0BFC	Specifies the device out endpoint 7 DMA buffer address register	0x0000_0000
DOEPCTL8	0x0C00	Specifies the device control out endpoint 8 control register	0x0000_0000
DOEPINT8	0x0C08	Specifies the device out endpoint 8 interrupt register	0x0000_0000
DOEPTSIZ8	0x0C10	Specifies the device out endpoint 8 transfer size register	0x0000_0000
DOEPDMA8	0x0C14	Specifies the device out endpoint 8 DMA address register	0x0000_0000
DOEPDMAB8	0x0C1C	Specifies the device out endpoint 8 DMA buffer address register	0x0000_0000
DOEPCTL9	0x0C20	Specifies the device control out endpoint 9 control register	0x0000_0000
DOEPINT9	0x0C28	Specifies the device out endpoint 9 interrupt register	0x0000_0000
DOEPTSIZ9	0x0C30	Specifies the device out endpoint 9 transfer size register	0x0000_0000
DOEPDMA9	0x0C34	Specifies the device out endpoint 9 DMA address register	0x0000_0000
DOEPDMAB9	0x0C3C	Specifies the device out endpoint 9 DMA buffer address register	0x0000_0000
DOEPCTL10	0x0C40	Specifies the device control out endpoint 10 control register	0x0000_0000
DOEPINT10	0x0C48	Specifies the device out endpoint 10 interrupt register	0x0000_0000

Register	Offset	Description	Reset Value
DOEPTSIZ10	0x0C50	Specifies the device out endpoint 10 transfer size register	0x0000_0000
DOEPDMA10	0x0C54	Specifies the device out endpoint 10 DMA address register	0x0000_0000
DOEPDMAB10	0x0C5C	Specifies the device out endpoint 10 DMA buffer address register	0x0000_0000
DOEPCTL11	0x0C60	Specifies the device control out endpoint 11 control register	0x0000_0000
DOEPINT11	0x0C68	Specifies the device out endpoint 11 interrupt register	0x0000_0000
DOEPTSIZ11	0x0C70	Specifies the device out endpoint 11 transfer size register	0x0000_0000
DOEPDMA11	0x0C74	Specifies the device out endpoint 11 DMA address register	0x0000_0000
DOEPDMAB11	0x0C7C	Specifies the device out endpoint 11 DMA buffer address register	0x0000_0000
DOEPCTL12	0x0C80	Specifies the device control out endpoint 12 control register	0x0000_0000
DOEPINT12	0x0C88	Specifies the device out endpoint 12 interrupt register	0x0000_0000
DOEPTSIZ12	0x0C90	Specifies the device out endpoint 12 transfer size register	0x0000_0000
DOEPDMA12	0x0C94	Specifies the device out endpoint 12 DMA address register	0x0000_0000
DOEPDMAB12	0x0C9C	Specifies the device out endpoint 12 DMA buffer address register	0x0000_0000
DOEPCTL13	0x0CA0	Specifies the device control out endpoint 13 control register	0x0000_0000
DOEPINT13	0x0CA8	Specifies the device out endpoint 13 interrupt register	0x0000_0000
DOEPTSIZ13	0x0CB0	Specifies the device out endpoint 13 transfer size register	0x0000_0000
DOEPDMA13	0x0CB4	Specifies the device out endpoint 13 DMA address register	0x0000_0000
DOEPDMAB13	0x0CBC	Specifies the device out endpoint 13 DMA buffer address register	0x0000_0000
DOEPCTL14	0x0CC0	Specifies the device control out endpoint 14 control register	0x0000_0000
DOEPINT14	0x0CC8	Specifies the device out endpoint 14 interrupt register	0x0000_0000
DOEPTSIZ14	0x0CD0	Specifies the device out endpoint 14 transfer size register	0x0000_0000
DOEPDMA14	0x0CD4	Specifies the device out endpoint 14 DMA address register	0x0000_0000
DOEPDMAB14	0x0CDC	Specifies the device out endpoint 14 DMA buffer address register	0x0000_0000
DOEPCTL15	0x0CE0	Specifies the device control out endpoint 15 control register	0x0000_0000
DOEPINT15	0x0CE8	Specifies the device out endpoint 15 interrupt register	0x0000_0000
DOEPTSIZ15	0x0CF0	Specifies the device out endpoint 15 transfer size register	0x0000_0000
DOEPDMA15	0x0CF4	Specifies the device out endpoint 15 DMA address register	0x0000_0000
DOEPDMAB15	0x0CFC	Specifies the device out endpoint 15 DMA buffer address register	0x0000_0000

32.8.2 USB PHY Control Register

32.8.2.1 UPHYPWR

- Base Address: 0x125B_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_7FF9

Name	Bit	Type	Description	Reset Value
RSVD	[31:15]	–	Reserved	17'b0
hsic_1_sleep	[14]	RW	HSIC1 PHY, Sleep mode 1'b0 = Normal 1'b1 = Sleep	1'b1
hsic_1_analog_ powerdown	[13]	RW	Analog block power down in HSIC1 PHY 1'b0 = Analog block power up (Normal Operation) 1'b1 = Analog block power down	1'b1
hsic_1_force _suspend	[12]	RW	HSIC1 PHY, Apply suspend signal to save power 1'b0 = Disables (Normal Operation) 1'b1 = Enables	1'b1
hsic_0_sleep	[11]	RW	HSIC0 PHY, sleep mode 1'b0 = Normal 1'b1 = Sleep	1'b1
hsic_0_analog_ powerdown	[10]	RW	Analog block power down in HSIC0 PHY 1'b0 = Analog block power up (Normal Operation) 1'b1 = Analog block power down	1'b1
hsic_0_force _suspend	[9]	RW	HSIC0 PHY, Apply suspend signal to save power 1'b0 = Disables (Normal Operation) 1'b1 = Enables	1'b1
phy_1_sleep	[8]	RW	Standard PHY for USB HOST, sleep mode 1'b0 = Normal 1'b1 = Sleep	1'b1
phy_1_analog _powerdown	[7]	RW	Standard USB PHY for USB HOST, analog block power down 1'b0 = Analog block power up (Normal Operation) 1'b1 = Analog block power down	1'b1
phy_1_force _suspend	[6]	RW	Standard PHY for USB HOST, apply suspend signal to save power 1'b0 = Disables (Normal Operation) 1'b1 = Enables	1'b1
phy_0_sleep	[5]	RW	Standard PHY for USB DE device VICE, sleep mode 1'b0 = Normal 1'b1 = Sleep	1'b1
phy_0_otg _disable	[4]	RW	Standard PHY for USB device, OTG block power down 1'b0 = OTG Block power up 1'b1 = OTG Block power down If the application does not use OTG functionality, you can set this input high to save power.	1'b1

Name	Bit	Type	Description	Reset Value
phy_0_analog _powerdown	[3]	RW	Standard PHY for USB device, Analog block power down 1'b0 = Analog block power up (Normal Operation) 1'b1 = Analog block power down	1'b1
RSVD	[2:1]	–	Reserved	2b'0
phy_0_force _suspend	[0]	RW	Standard PHY for USB device, apply suspend signal for to save power save 1'b0 = Disables (Normal Operation) 1'b1 = Enables	1'b1

32.8.2.2 UPHYCLK

- Base Address: 0x125B_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_9005

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	15'h0
hsic_refclkdiv	[16:10]	RW	Reference clock frequency select HSIC PHY. This bus selects the HSIC transceiver PHY reference clock frequency 6'b0100100 = 12 MHz 6'b0011100 = 15 MHz 6'b0011010 = 16 MHz 6'b0010101 = 19.2 MHz 6'b0010100 = 20 MHz	6'b0100100
RSVD	[9:8]	–	Reserved	?
phy_1_common_on_n	[7]	RW	Standard PHY for USB HOST, Force XO, Bias, Bandgap, and PLL to Remain Powered During a Suspend. This bit controls the power-down signals of sub-blocks in the Common block if the Standard PHY is suspended. 1'b0 = 48 MHz Clock on clk48m_ohci is available at all times, except in Suspend mode. 1'b1 = 48 MHz Clock on clk48m_ohci is available at all times, even in Suspend mode.	1'b0
RSVD	[6:5]	–	Reserved	?
phy_0_common_on_n	[4]	RW	Standard PHY for USB device, Force XO, Bias, Bandgap, and PLL to Remain Powered During a Suspend. This bit controls the power-down signals of sub-blocks in the Common block if the Standard PHY is suspended. 1'b0 = 48 MHz Clock on clk48m_ohci is available at all times, except in Suspend mode. 1'b1 = 48 MHz Clock on clk48m_ohci is available at all times, even in Suspend mode.	1'b0
phy_0_id_pullup0	[3]	RW	Standard PHY for USB device, Analog ID Input Sample Enable 1'b0 = Id_dig disable. 1'b1 = Id_dig enable. (The id_dig output is valid, and within 20 ms, Id_dig must indicate the type of plug connected.)	1'b0
phy_fsel	[2:0]	RW	Reference Clock Frequency Select for PLL in Standard PHY 3'b000 = 9.6 MHz 3'b001 = 10 MHz 3'b010 = 12 MHz 3'b011 = 19.2 MHz 3'b100 = 20 MHz 3'b101 = 24 MHz	3'b101

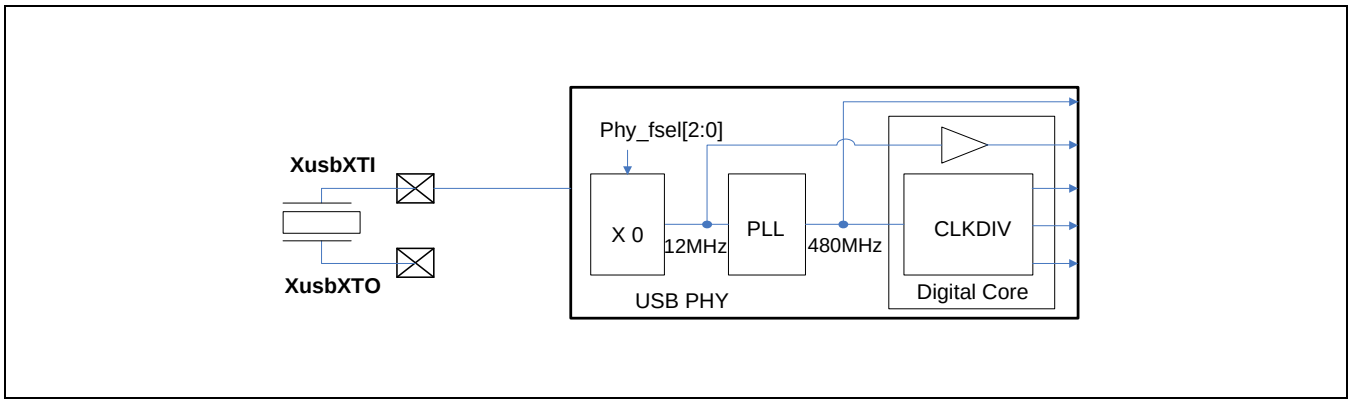


Figure 32-4 USB PHY Clock Path

32.8.2.3 URSTCON

- Base Address: 0x125B_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0079

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved	22'h0
host_link _port_2_rst	[10]	RW	USB HOST link Port2 S/W reset (to/from HSIC1)	1'b0
host_link _port_1_rst	[9]	RW	USB HOST link Port1 S/W reset (to/from HSIC0)	1'b0
host_link _port_0_rst	[8]	RW	USB HOST link Port0 S/W reset (to/from Standard USB PHY)	1'b0
host_link_all _sw_rst	[7]	RW	USB Host link all S/W reset	1'b0
hsic_1_sw_rst	[6]	RW	HSIC1 PHY Power-On Reset. This signal resets all test registers and state machines for the HSIC0 PHY and it must be asserted for a minimum of 10 µs	1'b1
hsic_0_sw_rst	[5]	RW	HSIC0 PHY Power-On Reset This signal resets all test registers and state machines for the HSIC0 PHY and it must be asserted for a minimum of 10 µs	1'b1
phy_1_sw_rst	[4]	RW	Standard USB HOST PHY Power-On Reset This signal resets all test registers and state machines for the Standard USB HOST PHY and it must be asserted for a minimum of 10 µs	1'b1
host_phy_sw_rst	[3]	RW	All USB PHY Power-On Reset. This signal resets all test registers and state machines for 2 HSIC PHYs and 1 Standard PHY. and it must be asserted for a minimum of 10 µs	1'b1
otg_phylink _sw_rst	[2]	RW	OTG Link Core phy_clock domain S/W Reset	1'b0
otg_hlink _sw_rst	[1]	RW	OTG Link Core hclk domain S/W Reset	1'b0
phy_0_sw_rst	[0]	RW	Standard USB device PHY Power-On Reset This signal resets all test registers and state machines for the Standard USB device PHY and it must be asserted for at least 10 µs	1'b1

32.8.2.4 UPHYTUNE1

- Base Address: 0x125B_0000
- Address = Base Address + 0x0020, Reset Value = 0x0028_19B3
- UPHYTUNE1 register is designed for USB HOST. Therefore you should check whether HOST_MODE bit is 1'b0 or not in USB control register of SYSREG.
- Below values must be static prior to setting POR to 1'b0. The default values for these bits are defined based on simulation results, and the USB 2.0 PHY is designed so that you do not have to change these bits from their default setting to center the design. However, in some cases for example, where extreme board parasitics cause the eye shape to degrade you can use these override bits to improve signal quality.

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	5'h0
txresetune	[26:25]	RW	USB Source Impedance Adjustment In some applications, there can be significant series resistance on the D+ and D– paths between the transceiver and cable. These bits adjust the driver source impedance to compensate for added series resistance on the USB. Note: Any setting other than the default can result in source impedance variation across process, voltage, and temperature conditions that does not meet USB 2.0 specification limits. 2'b11: Source impedance is reduced by approximately 4 Ω. 2'b10: Source impedance is reduced by approximately 2 Ω. 2'b01: Design default 2'b00: Source impedance is increased by approximately 1.5 Ω.	2'b01
txrisetune	[24:23]	RW	HS Transmitter Rise/Fall Time Adjustment These bits adjust the rise/fall times of the high-speed waveform. 2'b11: – 20% 2'b10: – 15% 2'b01: Design default 2'b00: + 10%	2'b01
txpreempamptune	[22:21]	RW	HS Transmitter Pre-Emphasis Current Control These bits control the amount of current sourced to DP0 and DM0 after a J-to-K or K-to-J transition. The HS Transmitter pre-emphasis current is defined in terms of unit amounts. One unit amount is approximately 600 μA and is defined as 1X pre-emphasis current. 2'b11: HS Transmitter pre-emphasis circuit sources 3X pre-emphasis current.	2'b01

			2'b10: HS Transmitter pre-emphasis circuit sources 2X pre-emphasis current. 2'b01 (design default): HS Transmitter pre-emphasis circuit sources 1X pre-emphasis current. 2'b00: HS Transmitter pre-emphasis is disabled.	
txpreemppulsetune	[20]	RW	HS Transmitter Pre-Emphasis Duration Control These bit controls the duration for which the HS pre-emphasis current is sourced onto DP0 or DM0. The HS Transmitter pre-emphasis duration is defined in terms of unit amounts. One unit of pre-emphasis duration is approximately 580 ps and is defined as 1X pre-emphasis duration. This bit is valid only if either TXPREEMPAMPTUNE0[1] or TXPREEMPAMPTUNE0[0] is set to 1'b1. 1'b1: 1X, short pre-emphasis current duration 1'b0: (design default) 2X, long pre-emphasis current duration	1'b0
compdistune	[19:17]	RW	Disconnect Threshold Adjustment These bits adjust the voltage level for the threshold used to detect a disconnect event at the host. 3'b111: + 4.5% 3'b110: + 3% 3'b101: + 1.5% 3'b100: Design default 3'b011: – 1.5% 3'b010: – 3% 3'b001: – 4.5% 3'b000: – 6%	3'b100
otgtune	[16:14]	RW	VBUS Valid Threshold Adjustment These bits adjust the voltage level for the VBUS Valid threshold. 3'b111: + 9% 3'b110: + 6% 3'b101: + 3% 3'b100: 0% 3'b011: – 3% 3'b010: – 6% 3'b001: – 9% 3'b000: – 12% (design default)	3'b000
sqrxtune	[13:11]	RW	Squelch Threshold Adjustment These bits adjust the voltage level for the threshold used to detect valid high-speed data. 3'b111: – 20% 3'b110: – 15% 3'b101: – 10% 3'b100: – 5%	3'b011

			3'b011: Design default 3'b010: + 5% 3'b001: + 10% 3'b000: + 15%	
txfslstune	[10:7]	RW	FS/LS Source Impedance Adjustment These bits adjust the low- and full-speed single-ended source impedance while driving high. The following adjustment values are based on nominal process, voltage, and temperature. 4'b1111: – 5% 4'b0111: – 2.5% 4'b0011: Design default 4'b0001: + 2.5% 4'b0000: + 5%	4'b0011
RSVD	[6]	–	Reserved	1'h0
txhsxvtune	[5:4]	RW	Transmitter High-Speed Crossover Adjustment These bits adjust the voltage at which the DP0 and DM0 signals cross while transmitting in HS mode. 2'b11: Default setting 2'b10: + 15 mV 2'b01: – 15 mV 2'b00: Reserved	2'b11
txvreftune	[3:0]	RW	HS DC Voltage Level Adjustment These bits adjust the high-speed DC level voltage. 4'b1111: + 24% 4'b1110: + 22% 4'b1101: + 20% 4'b1100: + 18% 4'b1011: + 16% 4'b1010: + 14% 4'b1001: + 12% 4'b1000: + 10% 4'b0111: + 8% 4'b0110: + 6% 4'b0101: + 4% 4'b0100: + 2% 4'b0011: Design default 4'b0010: – 2% 4'b0001: – 4% 4'b0000: – 6%	4'b0011

32.8.2.5 UPHYTUNE0

- Base Address: 0x125B_0000
- Address = Base Address + 0x0024, Reset Value = 0x0028_19B3
- UPHYTUNE0 register is designed for USB Device. Therefore you should check whether HOST_MODE bit is 1'b0 or not in USB control register of SYSREG.
- Below values must be static prior to setting POR to 1'b0. The default values for these bits are defined based on simulation results, and the USB 2.0 PHY is designed so that you do not have to change these bits from their default setting to center the design. However, in some cases for example, where extreme board parasitics cause the eye shape to degrade you can use these override bits to improve signal quality.

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	5'h0
txresetune	[26:25]	RW	USB Source Impedance Adjustment In some applications, there can be significant series resistance on the D+ and D– paths between the transceiver and cable. These bits adjust the driver source impedance to compensate for added series resistance on the USB. Note: Any setting other than the default can result in source impedance variation across process, voltage, and temperature conditions that does not meet USB 2.0 specification limits. 2'b11: Source impedance is reduced by approximately 4 Ω. 2'b10: Source impedance is reduced by approximately 2 Ω. 2'b01: Design default 2'b00: Source impedance is increased by approximately 1.5 Ω.	2'b01
txrisetune	[24:23]	RW	HS Transmitter Rise/Fall Time Adjustment These bits adjust the rise/fall times of the high-speed waveform. 2'b11: – 20% 2'b10: – 15% 2'b01: Design default 2'b00: + 10%	2'b01
txpreempamptune	[22:21]	RW	HS Transmitter Pre-Emphasis Current Control These bits control the amount of current sourced to DP0 and DM0 after a J-to-K or K-to-J transition. The HS Transmitter pre-emphasis current is defined in terms of unit amounts. One unit amount is approximately 600 μA and is defined as 1X pre-emphasis current. 2'b11: HS Transmitter pre-emphasis circuit sources 3X pre-emphasis current.	2'b01

			2'b10: HS Transmitter pre-emphasis circuit sources 2X pre-emphasis current. 2'b01 (design default): HS Transmitter pre-emphasis circuit sources 1X pre-emphasis current. 2'b00: HS Transmitter pre-emphasis is disabled.	
txpreemppulsetune	[20]	RW	HS Transmitter Pre-Emphasis Duration Control These bit controls the duration for which the HS pre-emphasis current is sourced onto DP0 or DM0. The HS Transmitter pre-emphasis duration is defined in terms of unit amounts. One unit of pre-emphasis duration is approximately 580 ps and is defined as 1X pre-emphasis duration. This bit is valid only if either TXPREEMPAMPTUNE0[1] or TXPREEMPAMPTUNE0[0] is set to 1'b1. 1'b1: 1X, short pre-emphasis current duration 1'b0: (design default) 2X, long pre-emphasis current duration	1'b0
compdistune	[19:17]	RW	Disconnect Threshold Adjustment These bits adjust the voltage level for the threshold used to detect a disconnect event at the host. 3'b111: + 4.5% 3'b110: + 3% 3'b101: + 1.5% 3'b100: Design default 3'b011: – 1.5% 3'b010: – 3% 3'b001: – 4.5% 3'b000: – 6%	3'b100
otgtune	[16:14]	RW	VBUS Valid Threshold Adjustment These bits adjust the voltage level for the VBUS Valid threshold. 3'b111: + 9% 3'b110: + 6% 3'b101: + 3% 3'b100: 0% 3'b011: – 3% 3'b010: – 6% 3'b001: – 9% 3'b000: – 12% (design default)	3'b000
sqrxtune	[13:11]	RW	Squelch Threshold Adjustment These bits adjust the voltage level for the threshold used to detect valid high-speed data. 3'b111: – 20% 3'b110: – 15% 3'b101: – 10% 3'b100: – 5%	3'b011

			3'b011: Design default 3'b010: + 5% 3'b001: + 10% 3'b000: + 15%	
txfslstune	[10:7]	RW	FS/LS Source Impedance Adjustment These bits adjust the low- and full-speed single-ended source impedance while driving high. The following adjustment values are based on nominal process, voltage, and temperature. 4'b1111: – 5% 4'b0111: – 2.5% 4'b0011: Design default 4'b0001: + 2.5% 4'b0000: + 5%	4'b0011
RSVD	[6]	–	Reserved	1'h0
txhsxvtune	[5:4]	RW	Transmitter High-Speed Crossover Adjustment These bits adjust the voltage at which the DP0 and DM0 signals cross while transmitting in HS mode. 2'b11: Default setting 2'b10: + 15 mV 2'b01: – 15 mV 2'b00: Reserved	2'b11
txvreftune	[3:0]	RW	HS DC Voltage Level Adjustment These bits adjust the high-speed DC level voltage. 4'b1111: + 24% 4'b1110: + 22% 4'b1101: + 20% 4'b1100: + 18% 4'b1011: + 16% 4'b1010: + 14% 4'b1001: + 12% 4'b1000: + 10% 4'b0111: + 8% 4'b0110: + 6% 4'b0101: + 4% 4'b0100: + 2% 4'b0011: Design default 4'b0010: – 2% 4'b0001: – 4% 4'b0000: – 6%	4'b0011

32.8.2.5.1 Parameter override Bits Usage

To enable adjusting various USB 2.0 specification-related characteristics, the USB 2.0 PHY provides top-level parameter override bits. The USB 2.0 PHY is designed to a default setting of these bits, and you are not expected to have to change these bits from their default setting. These override bits are not intended for per-part centering or dynamic centering. However, there might be circumstances that require bit settings that are different than the default. If a change is required, statically set these bits to the same value for all product parts. Statistical analysis of the USB 2.0 PHY's silicon characterization will determine whether any of these bits require a different setting other than the default. The following sections provide an example usage of the override bits and describe the function of each override bit.

Note) The following sections include example eye diagrams that are provided for illustrative purposes only and do not necessarily reflect the effect of the actual bit settings in the USB 2.0 PHY design

32.8.2.5.1.1 compdistune

COMPDISTUNE overrides the disconnect threshold for host applications. The default setting creates a disconnect threshold of approximately 575 mV. If test-chip screening shows that the threshold must be changed, these override bits can be used to re-center the threshold. COMPDISTUNE affects only the disconnect threshold; COMPDISTUNE does not affect any other USB 2.0 PHY characteristic.

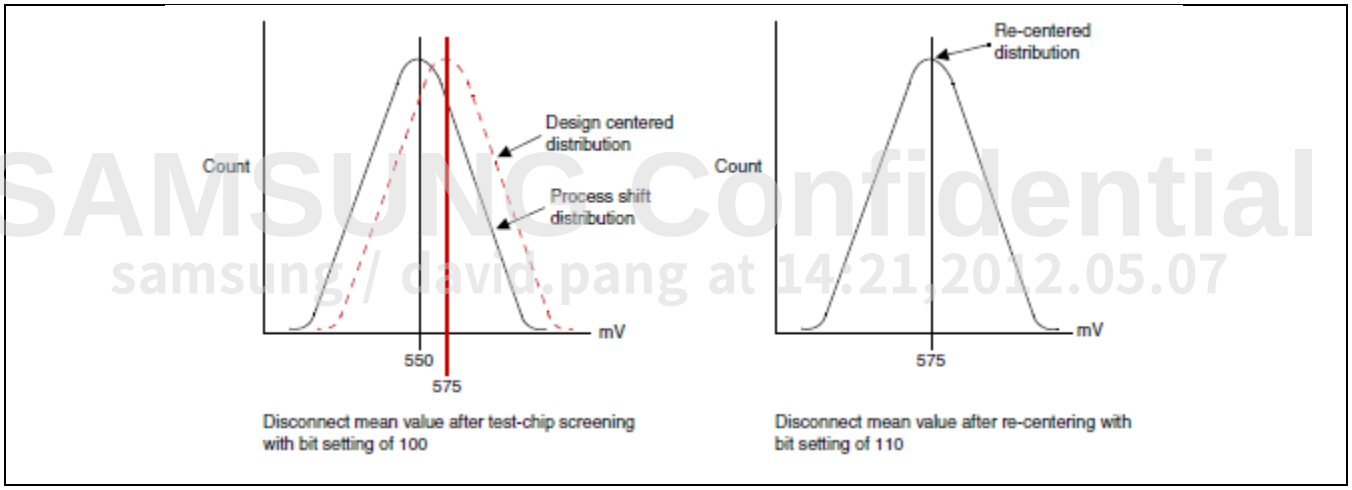


Figure 32-5 COMPDISTUNE

32.8.2.5.1.2 otgtune

OTGTUNE overrides the VBUS Valid threshold for OTG applications. The default setting creates a VBUS Valid threshold of approximately 4.65 V. If test-chip screening shows that the threshold must be changed, these override bits can be used to re-center the threshold. OTGTUNE affects only the disconnect threshold. OTGTUNE does not affect any other USB 2.0 PHY characteristic.

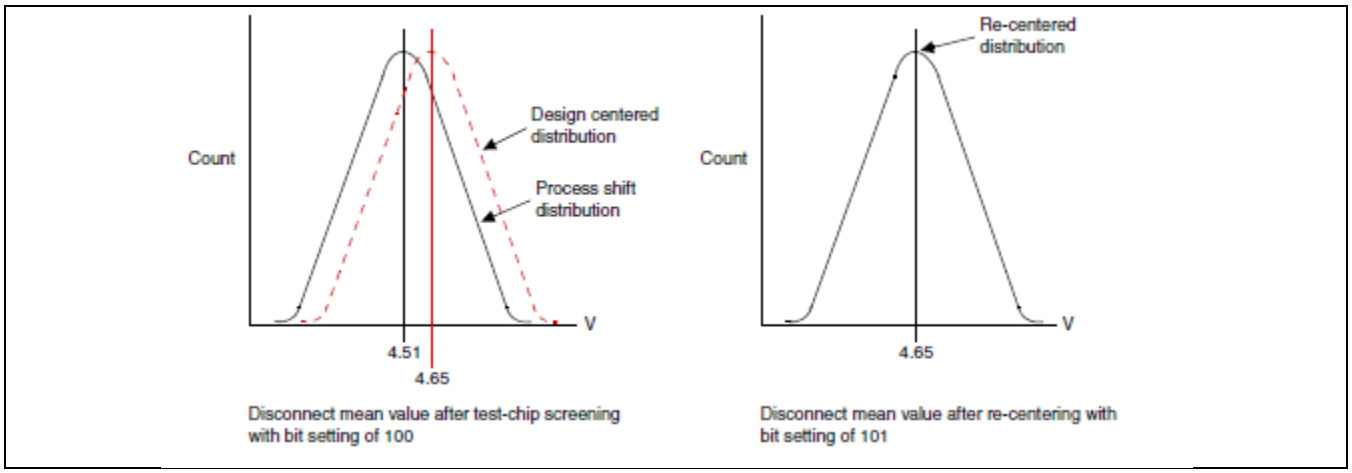


Figure 32-6 OTGTUNE

32.8.2.5.1.3 sqrxtune

SQRXTUNE overrides the squelch threshold for HS data detection. The default setting creates a squelch threshold of approximately 125 mV. If test-chip screening determines that the threshold must be changed, these override bits can be used to re-center the threshold. SQRXTUNE affects only the squelch threshold. SQRXTUNE does not affect any other USB 2.0 PHY characteristic.

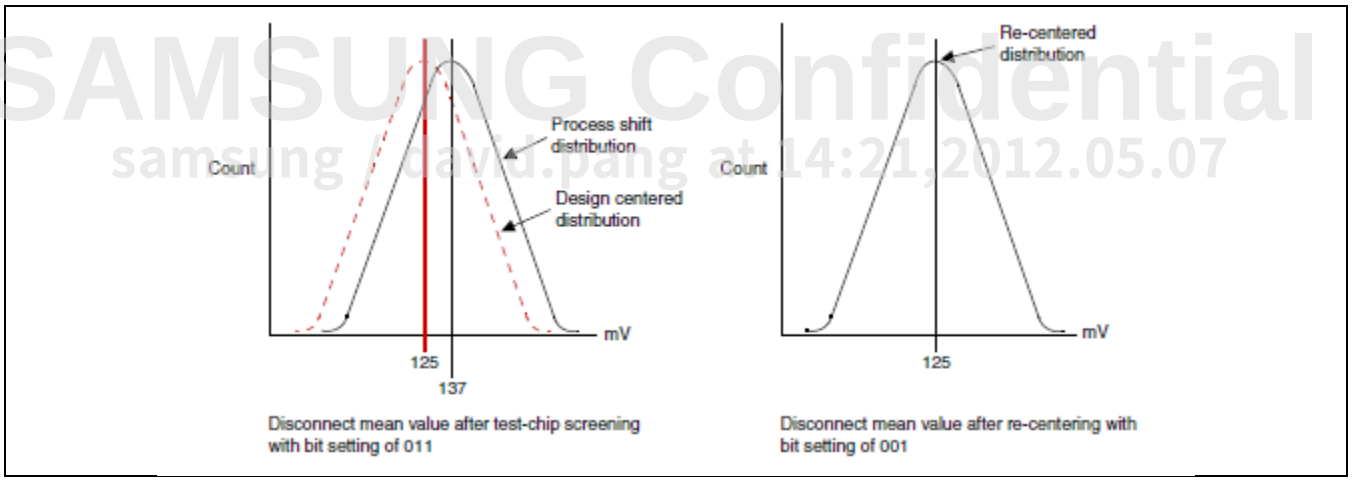


Figure 32-7 SQRXTUNE

32.8.2.5.1.4 txfslstune

TXFSLSTUNE overrides the single-ended FS source impedance when driving high. The automatic tuning circuit of the USB 2.0 PHY tunes the HS input impedance. The single-ended FS/LS source impedance while driving low is a replica of the HS input impedance while driving low; therefore, the FS/LS source impedance while driving low is centered automatically. By design, the FS/LS source impedance while driving high is matched to the drive-low impedance. Due to a shift in process, the FS/LS drive-high source impedance might be different than the drive-low impedance. If test-chip screening shows that the threshold must be changed, these override bits can be used to re-center the threshold. TXFSLSTUNE affects only the FS/LS source impedance of both the DP and DM drivers. TXFSLSTUNE does not affect any other USB 2.0 PHY characteristic.

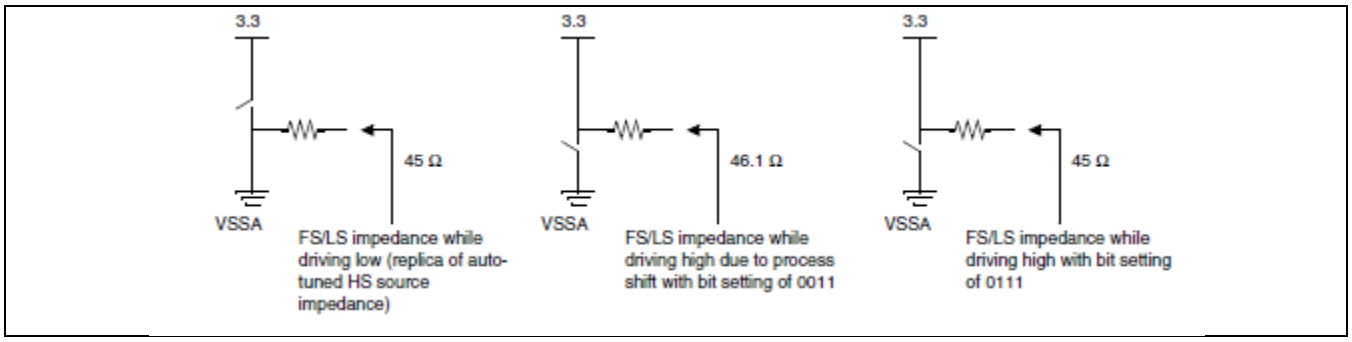


Figure 32-8 TXFSLSTUNE

32.8.2.5.1.5 txpreempamptune

Top-level control bits TXPREEMPAMPTUNE [1:0] add pre-emphasis to the rising edge of each HS transmit data bit. The HS eye might become distorted due to board, package, and on-chip parasitic capacitance. These top-level bits are designed to add extra current on each rising edge of the HS TX data bit on DP or DM. The default setting disables pre-emphasis. If test-chip screening shows that only the rising edge of HS transmit data are slowed down, these override bits can be used to improve the HS eye characteristics. As shown in Figure 4-19, adding pre-emphasis affects the rising edge of HS TX data. Furthermore, the addition of pre-emphasis can cause overshoot (as shown in Figure 4-19) in the HS eye depending on how much the rising edge was initially slowed.

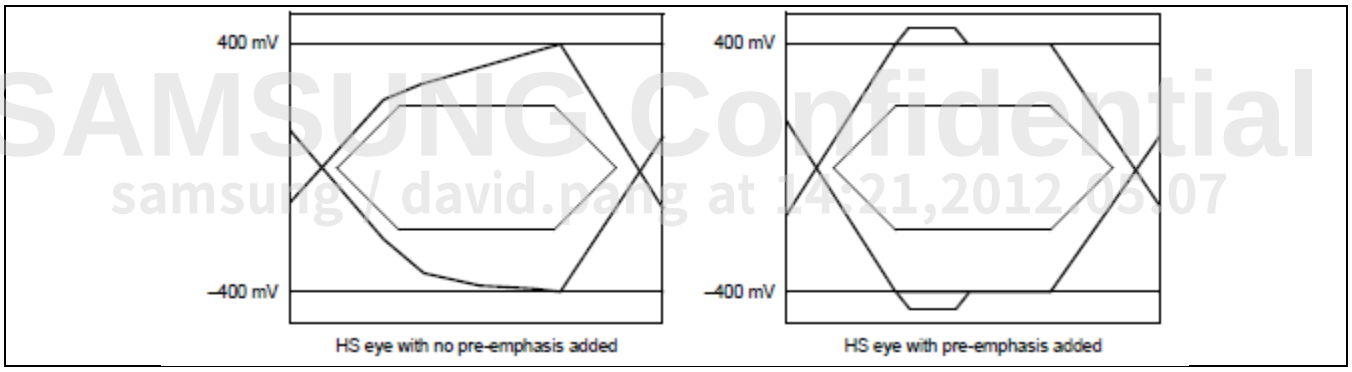


Figure 32-9 TXPREEMPAMPTUNE

32.8.2.5.1.6 txpreemppulsetune

In some applications, board trace lengths or parasitic resistance and capacitance values can make it necessary to change the duration for which pre-emphasis is applied. TXPREEMPPULSETUNE controls the duration of the pulse pre-emphasis pulse width. The default setting selects the longer pre-emphasis duration. Test-chip screening will show if the setting of TXPREEMPPULSETUNE is correct. Adjust this bit as needed to change the HS eye characteristics. As shown in Figure 32-10 TXPREEMPAMPTUNE affects the rising edge of HS TX data. Figure 32-10 shows a conceptual change in the duration of HS pre-emphasis.

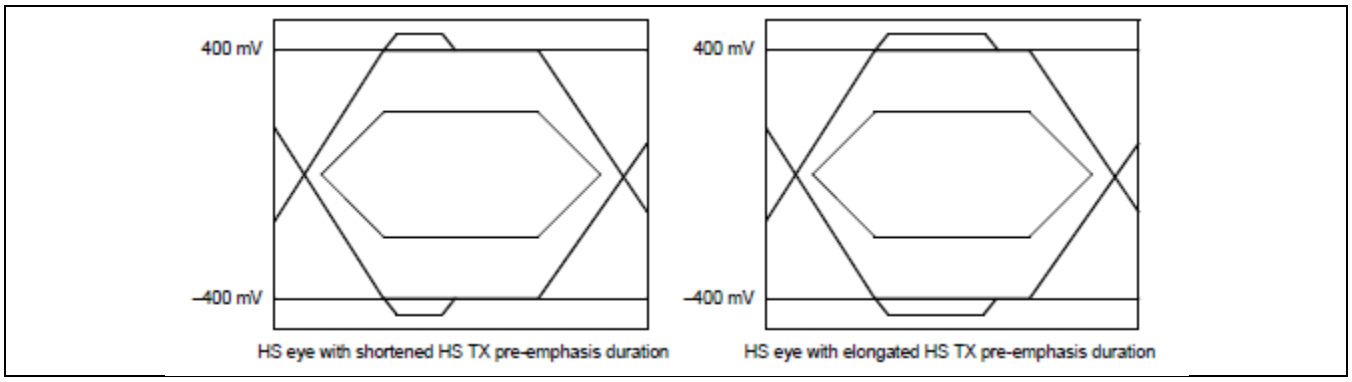


Figure 32-10 TXPREEMPPULSETUNE

32.8.2.5.1.7 txrisetune

TXRISETUNE overrides the rise and fall times of HS transmit data bits. Due to board, package, and on-chip parasitic capacitance, the HS eye rise/fall times might become too slow or too fast according to the USB 2.0 specification. If test-chip screening shows that the HS transmit data rise and fall times are either too fast or too slow, these override bits can be used to re-center the rise and fall times. TXRISETUNE can create or remove slight overshoot and ringing due to faster or slower HS edge rates, respectively.

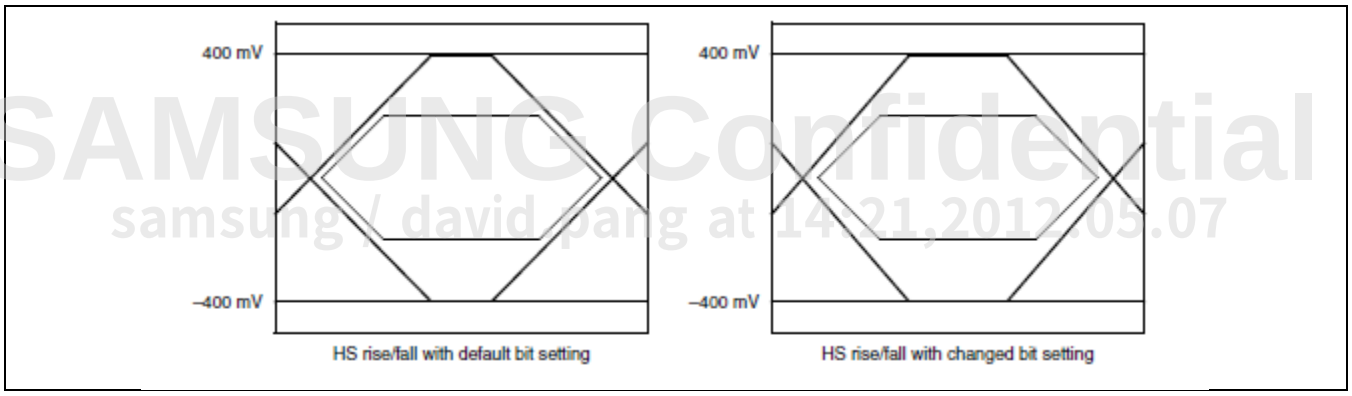


Figure 32-11 TXRISETUNE

32.8.2.5.1.8 txvreftune

TXVREFTUNE overrides the HS transmit DC level. The default bit setting is intended to create an HS transmit DC level of approximately 400 mV. If test-chip screening shows that the level must be changed, these override bits can be used to re-center the level. TXVREFTUNE might affect HS rise and fall times, because the HS transmit DC level is changed while the HS transmit slew rate remains constant.

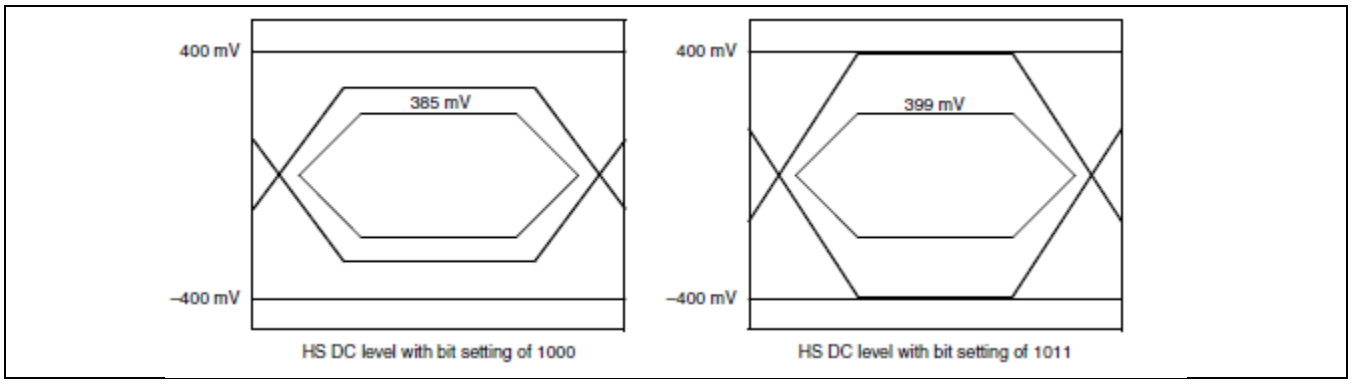


Figure 32-12 TXVREFTUNE

32.8.2.5.1.9 txhsxvtune

TXHSXVTUNE overrides the HS eye crossover voltage. The default bit setting is intended to set the crossover voltage as close as possible to 0 V. If test-chip screening shows that the crossover voltage must be changed, these override bits can be used to re-center the crossover voltage. TXHSXVTUNE affects HS rise and fall times.

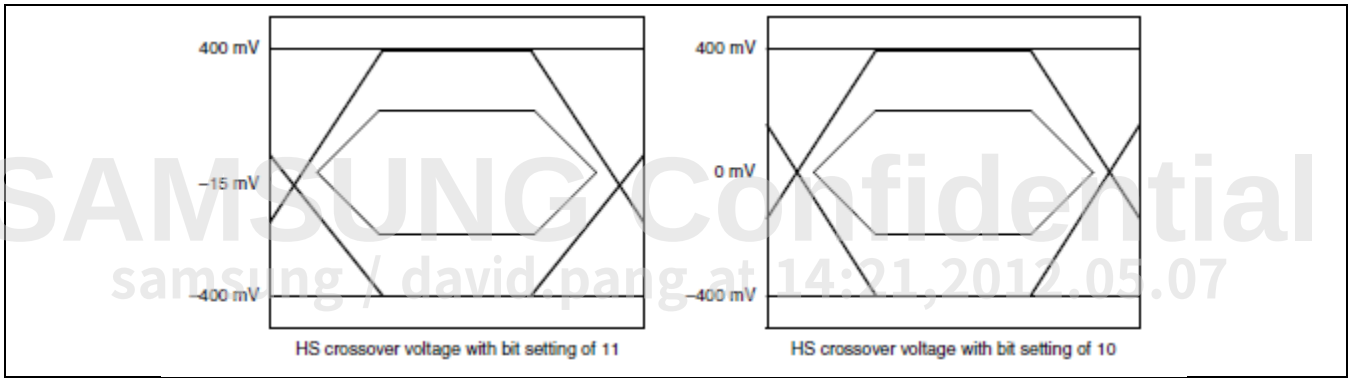


Figure 32-13 TXHSXVTUNE

32.8.3 OTG LINK Core Register (OTG Global Register)

These registers are available in device modes, and not required to be reprogrammed to switch between these modes.

32.8.3.1 GOTGCTL

- Base Address: 0x1248_0000
- Address = Base Address + 0x0000, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	12'h0
BSesVld	[19]	R	B-Session Valid Indicates the device mode transceiver status. 1'b0 = B-session is not valid 1'b1 = B-session is valid	1'b0
ASesVld	[18]	R	A-Session Valid Indicates the Host mode transceiver status. 1'b0 = A-session is not valid 1'b1 = A-session is valid	1'b0
DbncTime	[17]	R	Long/Short DEBOUNCE Time Indicates the DEBOUNCE time of a detected connection. 1'b0 = Long DEBOUNCE time, used for physical connections 1'b1 = Short DEBOUNCE time, used for soft connections	1'b0
ConIDSts	[16]	R	Connector ID Status Indicates the connector ID status. 1'b0 = The OTG core is in A-device mode 1'b1 = The OTG core is in B-device mode	1'b1
RSVD	[15:12]	–	Reserved	4'h0
DevHNPEn	[11]	RW	Device HNP Enable The application sets the bit if it successfully receives a Set Feature. 1'b0 = HNP is not enabled in the application 1'b1 = HNP is enabled in the application	1'b0
HstSet HNPEn	[10]	RW	Host Set HNP Enable The application sets this bit if it has successfully enabled HNP on the connected device. 1'b0 = Host Set HNP is not enabled 1'b1 = Host Set HNP is enabled	1'b0
HNPReq	[9]	RW	HNP Request The application sets this bit to initiate an HNP request to the connected USB host. The core clears this bit if the HstNegSucStsChng bit is cleared. 1'b0 = No HNP request 1'b1 = HNP request	1'b0
HstNegScs	[8]	R	Host Negotiation Success	1'b0

Name	Bit	Type	Description	Reset Value
			The core sets this bit if host negotiation is successful. The core clears this bit if the HNP Request (HNPReq) bit in this register is set. 1'b0 = Host negotiation failure 1'b1 = Host negotiation success	
RSVD	[7:2]	–	Reserved	6'h0
SesReq	[1]	RW	Session Request The application sets this bit to initiate a session request on the USB. The core clears this bit if the HstNegSucStsChng bit is cleared. 1'b0 = No session request 1'b1 = Session request	1'b0
SesReqScs	[0]	R	Session Request Success The core sets this bit if a session request initiation is successful. 1'b0 = Session request failure 1'b1 = Session request success	1'b0

The OTG Control and Status register controls the behavior and reflects the status of the core's OTG function.

32.8.3.2 GOTGINT

- Base Address: 0x1248_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	12'h0
DbnceDone	[19]	R_SS_WC	DEBOUNCE Done The core sets this bit if the DEBOUNCE is complete after the device connects. This bit is valid if the HNP Capable or SRP Capable bit is set in the Core USB Configuration register.	1'b0
ADev TOUTChg	[18]	R_SS_WC	A-Device Timeout Change The core sets this bit to indicate that the A-device has timed out while waiting for the B-device to connect.	1'b0
HstNegDet	[17]	R_SS_WC	Host Negotiation Detected. The core sets this bit if it detects a host negotiation request on the USB.	1'b0
RSVD	[16:10]	–	Reserved	7'h0
HstnegSuc StsChng	[9]	R_SS_WC	Host Negotiation Success Status Change The core sets this bit on the success or failure of a USB host negotiation request.	1'b0
SesReq SucStsChng	[8]	R_SS_WC	Session Request Success Status Change The core sets this bit on the success or failure of a session request.	1'b0
RSVD	[7:3]	–	Reserved	5'h0
SesEndDet	[2]	R_SS_WC	Session End Detected The core sets this bit if the b_valid signal is DEASSERTED.	1'b0
RSVD	[1:0]	–	Reserved	2'h0

The application reads this register at the time of OTG interrupt. To clear the OTG interrupt, application clears the bits in this register.

32.8.3.3 GAHBCFG

- Base Address: 0x1248_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	23'h0
PTxFEmpLvl	[8]	RW	Periodic TxFIFO Empty Level Indicates if the periodic TxFIFO empty Interrupt bit in the Core Interrupt registers (GINTSTS.PTxFEmp) is triggered. This bit is used only in Slave mode. 1'b0 = GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is half empty 1'b1 = GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is completely empty	1'b0
NPTxFEmpLvl	[7]	RW	Non-Periodic TxFIFO Empty Level This bit is used only in slave mode. This bit indicates when IN endpoint Transmit FIFO empty interrupt (DIEPINTn.TxFEmp) is triggered. 1'b0 = DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is half empty 1'b1 = DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is completely empty	1'b0
RSVD	[6]	–	Reserved	1'b0
DMAEn	[5]	RW	DMA Enable 1'b0 = Core operates in slave mode 1'b1 = Core operates in a DMA mode	1'b0
HBstLen	[4:1]	RW	Burst Length/Type Internal DMA Mode – AHB master burst type: 4'b0000 = Single 4'b0001 = INCR 4'b0011 = INCR4 4'b0101 = INCR8 4'b0111 = INCR16 Others = Reserved	4'b0
GlblIntrMsk	[0]	RW	Global Interrupt Mask The application uses this bit to mask or unmask the interrupt line assertion. Irrespective of this bit's setting, the interrupt status registers are updated by the core 1'b0 = Mask the interrupt assertion to the application 1'b1 = Unmask the interrupt assertion to the application	1'b0

This register configures the core after power-on or a change in mode of operation. This register mainly contains AHB system-related configuration parameters. Do not change this register after the initial programming. The application must program this register before starting any transactions on either the AHB or the USB.

32.8.3.4 GUSBCFG

- Base Address: 0x1248_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_1408

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	1'h0
ForceDevMode	[30]	RW	Force Device Mode Writing a 1 to this bit forces the core to device mode irrespective of utmiotg_iddig input pin. 1'b0 = Normal mode 1'b1 = Force device mode After setting the force bit, the application must wait at least 25 ms before the change to take effect.	1'b0
ForceHstMode	[29]	RW	Force Host Mode Writing a 1 to this bit forces the core to host mode irrespective of utmiotg_iddig input pin. 1'b0 = Normal mode 1'b1 = Force Host mode After setting the force bit, the application must wait at least 25 ms before the change to take effect.	1'b0
RSVD	[28:14]	–	Reserved	15'h0
USBTrdTim	[13:10]	RW	USB Turnaround Time Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (SPRAM). This must be programmed to 4'h5 = When the MAC interface is 16-bit UTMI+. 4'h9 = When the MAC interface is 8-bit UTMI+. NOTE: The values above are calculated for the minimum AHB frequency of 30 MHz. USB turnaround time is critical for certification where long cables and 5-Hubs are used, so if you need the AHB to run at less than 30 MHz, and if USB turnaround time is not critical, these bits can be programmed to a larger value.	4'h5
HNPCap	[9]	RW	HNP-Capable The application uses this bit to control the OTG controller HNP capabilities. 1'b0 = HNP capability is not enabled 1'b1 = HNP capability is enabled	1'b0
SRPCap	[8]	RW	SRP-Capable The application uses this bit to control the OTG core's SRP capabilities. 1'b0 = SRP capability is not enabled 1'b1 = SRP capability is enabled	1'b0
RSVD	[7:4]	–	Reserved	4'h0
PHYIf	[3]	RW	PHY Interface	1'b1

Name	Bit	Type	Description	Reset Value
			The application uses this bit to configure the core to support a UTMI+ PHY with an 8 - or 16-bit interface. Only 16-bit interface is supported. This bit must be set to 1. 1'b0 = 8 bits 1'b1 = 16 bits	
TOutCal	[2:0]	RW	HS/FS Timeout Calibration Set this bit to 3'h7.	3'h0

This register configures the core after power-on or a changing to Host mode or device mode. It contains USB and USB-PHY related configuration parameters. The application must program this register before starting any transactions on either the AHB or the USB. Do not make changes to this register after the initial programming.

32.8.3.5 GRSTCTL

- Base Address: 0x1248_0000
- Address = Base Address + 0x0010, Reset Value = 0x8000_0000

Name	Bit	Type	Description	Reset Value
AHBIdle	[31]	R	AHB Master Idle Indicates that the AHB Master State Machine is in the IDLE condition.	1'b1
DMAReq	[30]	R	DMA Request Signal Indicates that the DMA request is in progress. Used for debug.	1'b0
RSVD	[29:11]	–	Reserved	19'h0
TxFNum	[10:6]	RW	TxFIFO Number This is the FIFO number. Use TxFIFO Flush bit to flush FIFO number. This field must not be changed until the core clears the TxFIFO Flush bit. 5'h0 = Non-periodic TxFIFO flush in Host mode. Tx FIFO 0 flush in device mode. 5'h1 = Periodic TxFIFO flush in Host mode. TxFIFO 1 Flush in device. 5'h2 = TxFIFO 2 flush in device mode … 5'hF = Periodic TxFIFO 15 flush in device mode 5'h10 = Flush all the transmit FIFOs in device or host mode	5'h0
TxFFlsh	[5]	R_WS_SC	TxFIFO Flush This bit selectively flushes a single or all transmit FIFOs, but cannot flush if the core is in the middle of a transaction. The application must only write this bit after checking that the core is neither writing to the TxFIFO nor reading from the TxFIFO. The application must wait until the core clears this bit before performing any operations. This bit takes 8 clocks to clear.	1'b0
RxFFlsh	[4]	R_WS_SC	RxFIFO Flush The application flushes the entire RxFIFO using this bit, but must first ensure that the core is not in the middle of a transaction. The application must only write to this bit after checking that the core is neither reading from the RxFIFO nor writing to the RxFIFO. The application must wait until the bit is cleared before performing any other operations. This bit takes 8 clocks to clear.	1'b0
INTknQFlsh	[3]	R_WS_SC	IN Token Sequence Learning Queue Flush The application writes this bit to flush the IN token sequence learning queue.	1'b0
FrmCntrRst	[2]	R_WS_SC	Host Frame Counter Reset The application writes this bit to reset the (micro) frame number counter inside the core. If the (micro) frame counter is reset, the subsequent SOF sent out by the core will have a (micro) frame number of 0.	1'b0
HSftRst	[1]	R_WS	HClk Soft Reset	1'b0

Name	Bit	Type	Description	Reset Value
		_SC	The application uses this bit to flush the control logic in the AHB clock domain. Only AHB clock domain pipelines are reset. FIFOs are not flushed with this bit. All state machines in the AHB clock domain are reset to the Idle state after terminating the transactions on the AHB, following the protocol. CSR control bits used by the AHB clock domain state machines are cleared. To clear this interrupt, status mask bits that control the interrupt status and are generated by the AHB clock domain state machine are cleared. Because interrupt status bits are not cleared, the application can get the status of any core events that occurred after it set this bit. This is a self-clearing bit that the core clears after all necessary logic is reset in the core. This can take several clocks, depending on the core's current state	
CSftRst	[0]	R_WS_SC	Core Soft Reset Resets the hclk and phy_clock domains as follows = Clears the interrupts and all the CSR registers except the following register bits = HCFG.FSLSPclkSel DCFG.DevSpd All module state machines (except the AHB Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed. Any transactions on the AHB master are terminated as soon as possible, after gracefully completing the last data phase of an AHB transfer. Any transactions on the USB are terminated immediately. The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which may take several clocks, depending on the current state of the core. Once this bit is cleared software must wait at least 3 PHY clocks before accessing the PHY domain. Software must also check that bit 31 of this register is 1 (AHB Master is IDLE) before starting any operation. Typically software reset is used during software development and if you dynamically change the PHY selection bits in the USB configuration registers listed above. If you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain has to be reset for proper operation.	1'b0

The application uses this register to reset various hardware features inside the core.

32.8.3.6 GINTSTS

- Base Address: 0x1248_0000
- Address = Base Address + 0x0014, Reset Value = 0x0400_0020

Name	Bit	Type	Description	Reset Value
WkUpInt	[31]	R_SS_WC	Resume/Remote Wakeup Detected Interrupt In device mode, this interrupt is asserted if a resume is detected on the USB. In Host mode, this interrupt is asserted if a remote wakeup is detected on the USB.	1'b0
SessReqInt	[30]	R_SS_WC	Session Request/New Session Detected Interrupt In Host mode, this interrupt is asserted if a session request is detected from the device. In device mode, this interrupt is asserted if the b_valid signal goes high.	1'b0
DisconnInt	[29]	R_SS_WC	Disconnect Detected Interrupt Asserted when a device disconnect is detected.	1'b0
ConIDStsChng	[28]	R_SS_WC	Connector ID Status Change The core sets this bit if there is a change in connector ID status.	1'b0
LPM_Int	[27]	R_SS_WC	LPM Transaction Received Interrupt The core asserts this interrupt the device receives an LPM transaction with a non-ERRORed response. The interrupt is asserted in Host mode when the device responds to an LPM token with a non-ERRORed response. Or when the host core has completed LPM transactions for the programmed number of times (GLPMCFG.RetryCnt). This field is valid only if OTG_ENABLE_LPM is set to 1 and the Global Core LPM Configuration register's LPM-Capable (LPMCap) field is set to 1.	1'b0
PTxFEmp	[26]	R	Periodic TxFIFO Empty Asserted if the Periodic Transmit FIFO is either half or completely empty and there is space for at least one entry to be written in the Periodic Request Queue. The half or completely empty status is determined by the Periodic TxFIFO Empty Level bit in the Core AHB Configuration register.(GAHBCFG.PTFEmpLvl)	1'b1
HChInt	[25]	R	Host Channels Interrupt The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred, and then read the corresponding Host Channel-n Interrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this bit.	1'b0
PrtInt	[24]	R	Host Port Interrupt	1'b0

Name	Bit	Type	Description	Reset Value
			The core sets this bit to indicate a change in port status of one of the OTG core ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host Port Control and Status register to clear this bit.	
ResetDet	[23]	RW	Reset Detected Interrupt The core asserts this interrupt in device mode when it detects a reset on the USB in Partial Power-Down mode when the device is in Suspend. This interrupt is not asserted in Host mode.	1'b0
FetSusp	[22]	R_SS_WC	Data Fetch Suspended. This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data for IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application for an endpoint mismatch algorithm For example, after detecting an endpoint mismatch, the application: • Sets a global non-periodic IN NAK handshake • Disables In endpoints • Flushes the FIFO • Determines the token sequence from the IN Token sequence learning queue • Re-enables the endpoints • Clears the global non-periodic IN NAK handshake If the global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token received: the core generates an "IN token received when FIFO empty" interrupt. The OTG then sends the host a NAK response. To avoid this scenario, the application checks the GINSTS. FetSusp interrupt, which ensures that the FIFO is full before clearing a global NAK handshake. Alternatively, the application masks the "IN token received when FIFO empty" interrupt if clearing a global IN NAK handshake.	1'b0
IncompIP	[21]	R_SS_WC	Incomplete Periodic Transfer (IncompIP) In Host mode, the core sets this interrupt bit if there are incomplete periodic transactions still pending which are scheduled for the current microframe.	1'b0
IncomplSOOUT			Incomplete Isochronous Out Transfer (IncompISOOUT) The device mode, the core sets this interrupt to indicate that there is at least one isochronous out endpoint on which the transfer is not complete in the current microframe. This interrupt is asserted along with the End	–

Name	Bit	Type	Description	Reset Value
			of periodic frame interrupt (EOPF) bit in this register.	
IncomplSOIN	[20]	R_SS_WC	Incomplete Isochronous IN Transfer (IncompISOIN) The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register. NOTE: This interrupt is not asserted in Scatter/Gather DMA mode.	1'b0
OEPInt	[19]	R	OUT Endpoints Interrupt (OEPInt) The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in device mode). The application must read the device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding device OUT Endpoint-n Interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit.	1'b0
IEPInt	[18]	R	IN Endpoints Interrupt (IEPInt) The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in device mode). The application must read the device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on which the interrupt occurred, and then read the corresponding device IN Endpoint-n Interrupt (DIEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINTn register to clear this bit.	1'b0
RSVD	[17:16]	–	Reserved	2'b0
EOPF	[15]	R_SS_WC	End of Periodic Frame Interrupt Indicates that the period specified in the Periodic Frame Interval field of the device configuration register (DCFG.PerFrInt) has been reached in the current microframe.	1'b0
ISOutDrop	[14]	R_SS_WC	Isochronous OUT Packet Dropped Interrupt The core sets this bit if it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO does not have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint.	1'b0
EnumDone	[13]	R_SS_WC	Enumeration Done The core sets this bit to indicate that speed enumeration is complete. The application must read the device Status (DSTS) register to obtain the enumerated speed.	1'b0
USBRst	[12]	R_SS	USB Reset	1'b0

Name	Bit	Type	Description	Reset Value
		_WC	The core sets this bit to indicate that a reset is detected on the USB.	
USBSusp	[11]	R_SS_WC	USB Suspend The core sets this bit to indicate that a suspend state was detected on the USB. The core enters the Suspended state if there is no activity on the line_state signal for an extended period of time.	1'b0
ErlySusp	[10]	R_SS_WC	Early Suspend The core sets this bit to indicate that an Idle state has been detected on the USB for 3 ms.	1'b0
RSVD	[9:8]	–	Reserved	2'b0
GOUTNakEff	[7]	R	Global OUT NAK Effective Indicates that the Set Global OUT NAK bit in the device Control register (DCTL.SGOUTNak), set by the application, has taken effect in the core. This bit can be cleared by writing the Clear Global OUT NAK bit in the device Control register (DCTL.CGOUTNak).	1'b0
GINNakEff	[6]	R	Global IN Non-periodic NAK Effective Indicates that the Set Global Non-periodic IN NAK bit in the device Control register (DCTL.SGNPInNak), set by the application, has taken effect in the core. That is, the core has sampled the Global IN NAK bit set by the application. This bit can be cleared by clearing the Clear Global Non-periodic IN NAK bit in the device Control register (DCTL.CGNPInNak). This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit.	1'b0
NPTxFEmp	[5]	R	Non-periodic TxFIFO Empty This interrupt is valid only when OTG_EN_DED_TX_FIFO = 0. This interrupt is asserted when the Non-periodic TxFIFO is either half or completely empty, and there is space for at least one entry to be written to the Non-periodic Transmit Request Queue. The half or completely empty status is determined by the Non-periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl).	1'b1
RxFLvl	[4]	R	RxFIFO Non-Empty Indicates that there is at least one packet pending to be read from the RxFIFO.	1'b0
Sof	[3]	R_SS_WC	Start of (micro) Frame In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF (HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt. In device mode, in the core sets this bit to indicate that	1'b0

Name	Bit	Type	Description	Reset Value
			an SOF token has been received on the USB. The application can read the device Status register to get the current (micro) frame number. This interrupt is seen only when the core is operating at either HS or FS.	
OTGInt	[2]	R	OTG Interrupt The core sets this bit to indicate an OTG protocol event. The application must read the OTG Interrupt Status (GOTGINT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the GOTGINT register to clear this bit.	1'b0
ModeMis	[1]	R_SS_WC	Mode Mismatch Interrupt The core sets this bit if the application is trying to access: • A Host mode register, if the core is operating in device mode • A device mode register, if the core is operating in Host mode	1'b0
CurMod	[0]	R	Current Mode Of Operation Indicates the current mode of operation. 1'b0 = Device mode 1'b1 = Host mode	1'b0

This register interrupts the application for system-level events in the current mode of operation (device mode or Host mode).

32.8.3.7 GINTMSK

- Base Address: 0x1248_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
WkUpIntMsk	[31]	RW	Resume/remote wakeup detected interrupt mask	1'b0
SessReqIntMsk	[30]	RW	Session request/new session detected interrupt mask	1'b0
DisconnIntMsk	[29]	RW	Disconnect detected interrupt mask	1'b0
ConIDStsChngMsk	[28]	RW	Connector id status change mask	1'b0
LPM_IntMsk	[27]	RW	LPM Transaction received interrupt mask	1'b0
PTxFEmpMsk	[26]	RW	Periodic TxFIFO empty mask	1'b0
HChIntMsk	[25]	RW	Host channels interrupt mask	1'b0
PrtIntMsk	[24]	RW	Host port interrupt mask	1'b0
ResetDetMsk	[23]	RW	Reset detected interrupt mask	1'b0
FetSuspMsk	[22]	RW	Data fetch suspended mask	1'b0
incomplPMsk incompISOOUTMsk	[21]	RW	Incomplete periodic transfer mask	1'b0
		–	Incomplete isochronous out transfer mask	
incompISOINMsk	[20]	RW	Incomplete isochronous in transfer mask	1'b0
OEPIntMsk	[19]	RW	OUT endpoints interrupt mask	1'b0
INEPIntMsk	[18]	RW	IN endpoints interrupt mask	1'b0
RSVD	[17]	–	Reserved	1'b0
RSVD	[16]	–	Reserved	1'b0
EOPFMsk	[15]	RW	End of periodic frame interrupt mask	1'b0
ISOOutDropMsk	[14]	RW	Isochronous out packet dropped interrupt mask	1'b0
EnumDoneMsk	[13]	RW	Enumeration done mask	1'b0
USBRstMsk	[12]	RW	USB reset mask	1'b0
USBSuspMsk	[11]	RW	USB suspend mask	1'b0
ErlySuspMsk	[10]	RW	Early suspend mask	1'b0
RSVD	[9]	–	Reserved	1'b0
RSVD	[8]	–	Reserved	1'b0
GOUTNakEffMsk	[7]	RW	Global out nak effective mask	1'b0
GINNakEffMsk	[6]	RW	Global Non-Periodic in nak effective mask	1'b0
NPTxFEmpMsk	[5]	RW	Non-Periodic TxFIFO empty mask	1'b0
RxFLvlMsk	[4]	RW	Receive FIFO non-empty mask	1'b0
SofMsk	[3]	RW	Start of (micro) frame mask	1'b0
OTGIntMsk	[2]	RW	OTG Interrupt mask	1'b0
ModeMisMsk	[1]	RW	Mode mismatch interrupt mask	1'b0
RSVD	[0]	–	Reserved	1'b0

This register works with the Core Interrupt register to interrupt the application. If an interrupt bit is masked, the interrupt associated with that bit will not be generated. However, the Core Interrupt (GINTSTS) register bit corresponding to that interrupt will still be set.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

32.8.3.8 GRXSTSR/GRXSTSP (Host Mode)

- Base Address: 0x1248_0000
- Address = Base Address + 0x001C, + 0x0020 Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:21]	–	Reserved	–
PktSts	[20:17]	R	Packet Status Indicates the status of the received packet. 4'b0010 = IN data packet received 4'b0011 = IN transfer completed (triggers an interrupt) 4'b0101 = Data toggle error (triggers an interrupt) 4'b0111 = Channel halted (triggers an interrupt) Others = Reserved	–
DPID	[16:15]	R	Data PID Indicates the Data PID of the received packet. 2'b00 = DATA0 2'b10 = DATA1 2'b01 = DATA2 2'b11 = MDATA	–
BCnt	[14:4]	R	Byte Count Indicates the byte count of the received IN data packet.	–
ChNum	[3:0]	R	Channel number Indicates the channel number to which the current received packet belongs.	–

A read to the Receive Status Debug Read register returns the contents of the top of the Receive FIFO. A read to the Receive Status Read and Pop register additionally pops the top data entry out of the RxFIFO.

The receive status contents must be interpreted differently in Host and device modes. The core ignores the receive status pop/read if the receive FIFO is empty and returns a value of 32'h0000_0000. The application must only pop the Receive Status FIFO if the Receive FIFO Non-Empty bit of the Core Interrupt register (GINTSTS.RxFLvl) is asserted.

32.8.3.9 GRXSTSR/GRXSTSP (Device Mode)

- Base Address: 0x1248_0000
- Address = Base Address + 0x001C, + 0x0020, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	7'h3F
FN	[24:21]	R	Frame Number This is the least significant 4 bits of the (micro) frame number in which the packet is received on the USB. This field is supported if isochronous OUT endpoints are supported.	4'hF
PktSts	[20:17]	R	Packet Status Indicates the status of the received packet. 4'b0001 = Global OUT NAK (triggers an interrupt) 4'b0010 = OUT data packet received 4'b0011 = OUT transfer completed (triggers an interrupt) 4'b0100 = SETUP transaction completed (triggers an interrupt) 4'b0110 = SETUP data packet received Others = Reserved	4'b1111
DPID	[16:15]	R	Data PID Indicates the Data PID of the received OUT data packet. 2'b00 = DATA0 2'b10 = DATA1 2'b01 = DATA2 2'b11 = MDATA	2'b11
BCnt	[14:4]	R	Byte Count Indicates the byte count of the received data packet.	11'h3FF
EPNum	[3:0]	R	Endpoint number Indicates the endpoint number to which the current received packet belongs.	4'hF

32.8.3.10 GRXFSIZ

- Base Address: 0x1248_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_1F00

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	16'h0
RxFDep	[15:0]	RW	RxFIFO Depth This value is in terms of 32-bit words. • Minimum value is 16 • Maximum value is 7936 The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth. A new value must be written to this field. Programmed values must not exceed the power-on value set.	16'h1F00

The application programs the RAM size that must be allocated to the RxFIFO.

32.8.3.11 GNPTXFSIZ

- Base Address: 0x1248_0000
- Address = Base Address + 0x0028, Reset Value = 0x1F00_1F00

Name	Bit	Type	Description	Reset Value
NPTxFDep	[31:16]	RW	Non-Periodic TxFIFO Depth (For host mode) This value is in terms of 32-bit words. • Minimum value is 16 • Maximum value is 7936 The power-on reset value of this register is specified as the Largest Non-Periodic Tx Data FIFO Depth (7936). A new value must be written to this field. Programmed values must not exceed the power-on value set.	16'h1F00
INEPTxF0Dep		–	IN Endpoint TxFIFO 0 Depth (For device mode) This value is in terms of 32-bit words. • Minimum value is 16 • Maximum value is 7936	
NPTxFStAddr	[15:0]	RW	Non-Periodic Transmit Start Address (For host mode) This field contains the memory start address for Non-Periodic Transmit FIFO RAM. The power-on reset value of this register is specified as the largest Rx Data FIFO Depth (7936). A new value must be written to this field. programmed values must not exceed the power-on value set.	16'h1F00
INEPTxF0StAddr		–	IN Endpoint FIFO0 Transmit RAM Start Address (For device mode) This field contains the memory start address for in endpoint transmit FIFO# 0	

The application programs the RAM size and the memory start address for the Non-Periodic TxFIFO.

32.8.3.12 GNPTXSTS

- Base Address: 0x1248_0000
- Address = Base Address + 0x002C, Reset Value = 0x0008_1F00

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	1'b0
NPTxQTop	[30:24]	R	Top of the Non-Periodic Transmit Request Queue. Entry in the Non-Periodic Tx Request Queue that is currently being processed by the MAC. • Bits[30:27]: Channel/endpoint number • Bits[26:25]: 2'b00 = IN/OUT token 2'b01 = Zero-length transmit packet (device IN/host OUT) 2'b10 = PING/CSPLIT token 2'b11 = Channel halt command Bit[24] = Terminate (last entry for selected channel/endpoint)	7'h0
NPTxQSpcAvail	[23:16]	R	Non-Periodic Transmit Request Queue Space Available. Indicates the amount of free space available in the Non-Periodic transmit request queue. This queue holds both IN and out requests in Host mode. Device mode has only IN requests. 8'h0 = Non-Periodic transmit request queue is full 8'h1 = 1 Location available 8'h2 = 2 Locations available n = n locations available(0 ≤ n ≤ 8) Others = Reserved	8'h08
NPTxFSpcAvail	[15:0]	R	Non-Periodic TxFIFO Space Available Indicates the amount of free space available in the Non-Periodic TxFIFO. Values are in terms of 32-bit words. 16'h0 = Non-Periodic TxFIFO is full 16'h1 = 1 Word available 16'h2 = 2 Words available 16'hn = n Words available (where 0 ≤ n ≤ 32768) 16'h8000 = 32768 Words available Others = Reserved	16'h1F00

This read-only register contains the free space information for the Non-Periodic TxFIFO and the Non-Periodic Transmit Request Queue.

32.8.3.13 GLPMCFG

- Base Address: 0x1248_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	4'b0
LPM_RetryCnt_Sts	[27:25]	R	Number of LPM host retries remaining to be transmitted for the current LPM sequence.	3'b0
SndLPM	[24]	R_WS_SC	When the application software sets this bit, an LPM transaction containing two tokens, EXT and LPM, is sent. The hardware clears this bit once a valid response (STALL, NYET, or ACK) is received from the device or the core has finished transmitting the programmed number of LPM retires. NOTE: This bit must only be set when the host is connected to a local port.	1'b0
LPM_Retry_Cnt	[23:21]	RW	When the device gives an ERROR response, this is the number of additional LPM retries that the host performs until a valid device response (STALL, NYET, or ACK) is received.	3'b0
LPM_Chnl_Indx	[20:17]	RW	The channel number on which the LPM transaction must be applied while sending an LPM transaction to the local device. Based on the LPM channel index, the core automatically inserts the device address and endpoint number programmed in the corresponding channel into the LPM transaction.	4'b0
L1ResumeOK	[16]	R	Indicates that the application or host can start a resume from the Sleep state. This bit is valid in the LPM Sleep (L1) state. It is set in Sleep mode after a delay of 50µs (TL1Residency). The bit is reset when SlpSts is 0 1'b1 = The application/core can start resume from the Sleep state 1'b0 = The application/core cannot start resume from the Sleep state	1'b0
SlpSts	[15]	R	Host Mode: The host transitions to the Sleep (L1) state as a side-effect of a successful LPM transaction by the core to the local port with an ACK response from the device. The read value of this bit reflects the port's current sleep status. The core clears this bit after: • The core detects a remote L1 Wakeup signal; • The application sets the Port Reset bit or the Port L1Resume bit in the HPRT register; or • The application sets the L1Resume/Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.L1WkUpInt or GINTSTS.DisconnInt, respectively).	1'b0
		–	Device Mode: This bit is set as long as a Sleep condition is present on the USB bus. The core enters the Sleep	

Name	Bit	Type	Description	Reset Value
			state when an ACK response is sent to an LPM transaction and the timer TL1TokenRetry. Has expired. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the PHY's Suspend input pin. The application must rely on SlpSts and not ACK in CoreL1Res to confirm transition into sleep. The core comes out of sleep: • When there is any activity on the USB line_state • When the application writes to the Remote Wakeup Signaling bit in the device Control register (DCTL.RmtWkUpSig) or when the application resets or soft-disconnects the device.	
CoreL1Res	[14:13]	R	Host Mode: The handshake response received from the local device for LPM transaction 00 = ERROR (No handshake response) 01 = STALL 10 = NYET 11 = ACK	2'b0
		–	Device Mode: The core's response to the received LPM transaction is reflected in these two bits.	
HIRD_Thres	[12:8]	RW	Host Mode: The core asserts L1SuspendM to put the PHY into Deep Low-Power mode in L1 when HIRD_Thres[4] is set to 1'b1. HIRD_Thres[3:0] specifies the time for which resume signaling is to be reflected by the host (TL1HubDrvResume2) on the USB when it detects device-initiated resume. HIRD_Thres must not be programmed with a value greater than 4'b1100 in Host mode, because this exceeds maximum TL1HubDrvResume2. Host mode resume Sl. No HIRD_Thres[3:0] signaling time (μs) 1 = 4'b0000 60 2 = 4'b0001 135 3 = 4'b0010 210 4 = 4'b0011 285 5 = 4'b0100 360 6 = 4'b0101 435 7 = 4'b0110 510 8 = 4'b0111 585 9 = 4'b1000 660 10 = 4'b1001 735 11 = 4'b1010 810 12 = 4'b1011 885 13 = 4'b1100 960 14 = 4'b1101 invalid 15 = 4'b1110 invalid 16 = 4'b1111 invalid	5'b0
		R	Device Mode: The core asserts L1SuspendM to put the PHY into Deep Low-Power mode in L1 when the HIRD value is greater than or equal to the value defined in this	

Name	Bit	Type	Description	Reset Value
			field (GLPMCFG.HIRD_Thres[3:0]), and HIRD_Thres[4] is set to 1'b1.	
EnblSlpM	[7]	RW	For UTMI+ interface: The application uses this bit to control utmi_sleep_n assertion to the PHY in the L1 state. For the host, this bit is valid only in Local device mode. 1'b0 = utmi_sleep_n assertion from the core is not transferred to the external PHY. 1'b1 = utmi_sleep_n assertion from the core is transferred to the external PHY when utmi_l1_suspend_n cannot be asserted.	1'b0
bRemoteWake	[6]	RW	Host Mode: The remote wakeup value to be sent in the LPM transaction's wIndex field.	1'b0
		R	Device Mode: This field is updated with the received bRemoteWake LPM token's bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction.	1'b0
HIRD	[5:2]	RW	Host Mode: The value of HIRD to be sent in an LPM transaction. This value is also used to initiate resume for a duration TL1HubDrvResume1 for host initiated resume	4'b0
HIRD	[5:2]	R	Device Mode: This field is updated with the Received LPM Token HIRD bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction Sl. No HIRD[3:0] THIRD (µs) 1 = 4'b0000 50 2 = 4'b0001 1253 4'b0010 200 4 = 4'b0011 275 5 = 4'b0100 350 6 = 4'b0101 425 7 = 4'b0110 500 8 = 4'b0111 575 9 = 4'b1000 650 10 = 4'b1001 725 11 = 4'b1010 800 12 = 4'b1011 875 13 = 4'b1100 950 14 = 4'b1101 1025 15 = 4'b1110 1100 16 = 4'b1111 1175	4'b0
AppL1Res	[1]	R	Handshake response to LPM token pre-programmed by device application software. The response depends on GLPMCFG.LPMCap. If GLPMCFG.LPMCap is 1'b0, the core always responds with a NYET. If GLPMCFG.LPMCap is 1'b1, the core responds as follows: 0 = NYET The pre-programmed software bit is overridden for response to LPM token when: - The	1'b0

Name	Bit	Type	Description	Reset Value
			received bLinkState is not L1 (STALL response) - An error is detected in either of the LPM token packets due to corruption (ERROR response). 1 = ACK Even though an ACK is pre-programmed, the core responds with an ACK only on a successful LPM transaction. The LPM transaction is successful if: - There are no PID/CRC5 errors in both the EXT token and the LPM token (else ERROR) - A valid bLinkState = 0001B (L1) is received in the LPM transaction (else STALL) - No data is pending in the Transmit queue (else NYET)	
LPMCap	[0]	RW	LPM-Capable (LPMCap) The application uses this bit to control the DWC_otg core LPM capabilities. If the core operates as a non-LPM-capable host, it cannot request the connected device/hub to activate LPM mode. If the core operates as a non-LPM-capable device, it cannot respond to any LPM transactions. 1'b0 = LPM capability is not enabled. 1'b1 = LPM capability is enabled. Otherwise, reads return 0.	1'b0

This register controls the operation of the core's LPM and HSIC capabilities. It also contains status bits pertaining to these features.

32.8.3.14 HPTXFSIZ

- Base Address: 0x1248_0000
- Address = Base Address + 0x0100, Reset Value = 0x0300_5A00

Name	Bit	Type	Description	Reset Value
PTxFSize	[31:16]	RW	Host Periodic TxFIFO Depth This value is in terms of 32-bit words • Minimum value is 16 • Maximum value is 7936 A new value must be written to this field. Programmed values must not exceed the Maximum value.	16'h0300
PTxFStAddr	[15:0]	RW	Host Periodic TxFIFO Start Address. The power-on reset value of this register is the sum of the Largest Rx Data FIFO Depth and Largest Non-periodic Tx Data FIFO Depth If you have programmed new values for the RxFIFO or Non-Periodic TxFIFO, write their sum in this field. Programmed values must not exceed the power-on value.	16'h5A00

This register holds the size and the memory start address of the Periodic TxFIFO.

32.8.3.15 DIEPTXFn (n = 1 to 15)

- Base Address: 0x1248_0000
- Address = Base Address + 0x0104, Reset Value = 0x0300_2200
- Address = Base Address + 0x0108, Reset Value = 0x0300_2500
- Address = Base Address + 0x010C, Reset Value = 0x0300_2800
- Address = Base Address + 0x0110, Reset Value = 0x0300_2B00
- Address = Base Address + 0x0114, Reset Value = 0x0300_2E00
- Address = Base Address + 0x0118, Reset Value = 0x0300_3100
- Address = Base Address + 0x011C, Reset Value = 0x0300_3400
- Address = Base Address + 0x0120, Reset Value = 0x0300_3700
- Address = Base Address + 0x0124, Reset Value = 0x0300_3A00
- Address = Base Address + 0x0128, Reset Value = 0x0300_3D00
- Address = Base Address + 0x012C, Reset Value = 0x0300_4000
- Address = Base Address + 0x0130, Reset Value = 0x0300_4300
- Address = Base Address + 0x0134, Reset Value = 0x0300_4600
- Address = Base Address + 0x0138, Reset Value = 0x0300_4900
- Address = Base Address + 0x013C, Reset Value = 0x0300_4C00

Name	Bit	Type	Description	Reset Value
INEPnTxFDep	[31:16]	RW	IN Endpoint TxFIFO Depth (INEPnTxFDep) This value is in terms of 32-bit words • Minimum value is 4 • Maximum value is 768 The Power-on reset value of this registeris specified as the Largest IN Endpoint FIFO number Depth. It can be write a new value in this field.	User selected
INEPnTxFStAddr	[15:0]	RW	IN Endpoint FIFOn Transmit RAM Start Address This field contains the memory start address for IN endpoint Transmit FIFOn. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth. You have programmed a new value for RxFIFO depth, you can write that value in this field. Programmed values must not exceed the power-on value.	User selected

This register holds the memory start address of IN endpoint TxFIFOs to implement in device mode. Each FIFO holds the data for one IN endpoint FIFOs. This register is repeated for IN endpoint FIFO instantiated.

32.8.4 Device Mode Register (Device Global Register)

These registers are visible only in device mode and must not be accessed in Host mode, as the results are unknown. Some of them affect all the endpoints uniformly, while others affect only a specific endpoint. Device Mode registers fall into two categories:

- Device Global registers
- Device logical endpoint-specific registers

32.8.4.1 DCFG

- Base Address: 0x1248_0000
- Address = Base Address + 0x0800, Reset Value = 0x0820_0000

Name	Bit	Type	Description	Reset Value
ResValid	[31:26]	RW	Resume Validation Period This field controls the period when the core resumes from a suspend. When this bit is set, the core counts for the ResValid number of clock cycles to detect a valid resume. This field is effective only when DCFG.Ena32 kHz Susp is set	6'h2
PerSchIntvl	[25:24]	RW	Periodic Scheduling Interval PerSchIntvl must be programmed only for Scatter/Gather DMA mode. This field specifies the amount of time the Internal DMA engine must allocate for fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25, 50 or 75 % of (micro) frame. • When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data • When no periodic endpoints are active, then the internal DMA engine services non-periodic endpoints, ignoring this field. • After the specified time within a (micro) frame, the DMA switches to fetching for non-periodic endpoints. 2'b00 = 25 % of (micro) frame. 2'b01 = 50 % of (micro) frame. 2'b10 = 75 % of (micro) frame. 2'b11 = Reserved.	2'b00
DescDMA	[23]	RW	Enable Scatter/Gather DMA in Device Mode. This bit must be modified only once after a reset. The following combinations are available for programming: GAHBCFG.DMAEn = 0,DCFG.DescDMA = 0 => Slave mode GAHBCFG.DMAEn = 0,DCFG.DescDMA = 1 => Invalid GAHBCFG.DMAEn = 1,DCFG.DescDMA = 0 => Buffered DMA mode GAHBCFG.DMAEn = 1,DCFG.DescDMA = 1 => Scatter/Gather DMA mode	1'b0
RSVD	[22:13]	–	Reserved	10'h100
PerFrInt	[12:11]	RW	Periodic Frame Interval	2'h0

Name	Bit	Type	Description	Reset Value
			Indicates the time within a (micro) frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine if all the isochronous traffic for that (micro) frame is complete. 2'b00 = 80 % of the (micro) frame interval 2'b01 = 85 % 2'b10 = 90 % 2'b11 = 95 %	
DevAddr	[10:4]	RW	Device Address The application must program this field after every Setaddress control command.	7'h0
RSVD	[3]	–	Reserved	1'b0
NZSts OUTHShk	[2]	RW	Non-Zero-Length Status OUT Handshake The application uses this field to select the handshake that the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage. 1'b0 = Sends a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. 1'b1 = Sends the received OUT packet to the application and send a handshake based on the NAK and stall bits for the endpoint in the device Endpoint Control register.	1'b0
DevSpd	[1:0]	RW	Device Speed. Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application supports. However the actual bus speed is determined only after the chirp sequence is complete, and is based on the speed of the USB host to which the core is connected. 2'b00 = High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 2'b01 = Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 2'b10 = Low speed (USB 1.1 transceiver clock is 6 MHz). If you select 6 MHz LS mode, you must do a soft reset. 2'b11 = Full speed (USB 1.1 transceiver clock is 48 MHz).	2'b0

This register configures the core after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.

32.8.4.2 DCTL

- Base Address: 0x1248_0000
- Address = Base Address + 0x0804, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	15'h0
NakOnBable	[16]	RW	Set NAK automatically on babble (NakOnBble). The core sets NAK automatically for the endpoint on which babble is received	1'b0
IgnrFrmNum	[15]	RW	Ignore frame number for Isochronous end points Do NOT program IgnrFrmNum bit to 1'b1 when the core is operating in threshold mode. This feature is not applicable to High Speed, High bandwidth transfers. When this bit is enabled, there must be only one packet per descriptor. 0 = The core transmits the packets only in the frame number in which they are intended to be transmitted. 1 = The core ignores the frame number, sending packets immediately as the packets are ready. When Scatter/Gather DMA mode is disabled, this field is reserved, and reads 1'b0. • In Scatter/Gather DMA mode, if this bit is enabled, the packets are not flushed when a ISOC IN token is received for an elapsed frame.	1’b0
GMC	[14:13]	RW	Global Multi Count. GMC must be programmed only once after initialization. Applicable only for Scatter/Gather DMA mode. This indicates the number of packets to be serviced for that end point before moving to the next end point. It is only for non-periodic end points. 2'b00 = Invalid. 2'b01 = 1 Packet. 2'b10 = 2 Packets. 2'b11 = 3 Packets. When Scatter/Gather DMA mode is disabled, this field is reserved. And reads 2'b00.	2’b01
RSVD	[12]	–	Reserved	-
PWROnPrgDone	[11]	RW	Power-On Programming Done The application uses this bit to indicate that register programming is complete after a wake-up from Power Down mode.	1'b0
CGOUTNak	[10]	W	Clear Global OUT NAK A write to this field clears the Global OUT NAK.	1'b0
SGOUTNak	[9]	W	Set Global OUT NAK A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints.	1'b0

Name	Bit	Type	Description	Reset Value
			The application must set this bit after making sure that the Global OUT NAK Effective bit in Core Interrupt Register is cleared.	
CGNPInNAK	[8]	W	Clear Global Non-Periodic IN NAK A write to this field clears the Global Non-Periodic IN NAK.	1'b0
SGNPInNAK	[7]	W	Set Global Non-Periodic IN NAK A write to this field sets the Global Non-Periodic IN NAK. The application uses this bit to send a NAK handshake on all non-periodic IN endpoints. The core sets this bit if a timeout condition is detected on a non-periodic endpoint. The application must set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register is cleared.	1'b0
TstCtl	[6:4]	RW	Test Control 3'b000 = Test mode disabled 3'b001 = Test_J mode 3'b010 = Test_K mode 3'b011 = Test_SE0_NAK mode 3'b100 = Test_Packet mode 3'b101 = Test_Force_Enable Others = Reserved	3'b0
GOUTNakSts	[3]	R	Global OUT NAK Status 1'b0 = A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. 1'b1 = No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped.	1'b0
GNPINNakSts	[2]	R	Global Non-Periodic IN NAK Status 1'b0 = A handshake is sent based on the data availability in the transmit FIFO. 1'b1 = A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.	1'b0
SftDiscon	[1]	RW	Soft Disconnect The application uses this bit to signal the OTG core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device will not receive signals on the USB. The core stays in the disconnected state until the application clears this bit. 1'b0 = Normal operation. If this bit is cleared after a soft disconnect, the core drives the opmode signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. If the device is reconnected, the USB host restarts device enumeration. 1'b1 = The core drives the opmode signal on the UTMI+ to 2'b01, which generates a device disconnect event to	1'b0

Name	Bit	Type	Description	Reset Value
			the USB host.	
RmtWkUpSig	[0]	RW	Remote Wakeup Signaling If the application sets this bit, the core initiates remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1-15 ms after setting it.	1'b0

The following table lists the minimum duration under various conditions for which the Soft Disconnect bit must be set for the USB host to detect a device disconnect. To accommodate clock jitter, it is recommended that the application add some extra delay to the specified minimum duration.

Operating Speed	Device state	Minimum Duration
High speed	Suspended	1 ms + 2.5 μs
High speed	Idle	3 ms + 2 × μs
High speed	Not Idle or suspended (performing transactions)	125 μs
Full speed/Low speed	Suspended	1 ms + 2.5 μs
Full speed/Low speed	Idle	2.5 μs
Full speed/Low speed	Not Idle or Suspended (performing transactions)	2.5 μs

32.8.4.3 DSTS

- Base Address: 0x1248_0000
- Address = Base Address + 0x0808, Reset Value = 0x0000_0002

Name	Bit	Type	Description	Reset Value
RSVD	[31:22]	–	Reserved	10'h0
SOFFN	[21:8]	R	Frame or Microframe Number of the Received SOF If the core is operating at high speed; this field contains a microframe number. If the core is operating at full or low speed, this field contains a frame number.	14'h0
RSVD	[7:4]	–	Reserved	4'h0
ErrticErr	[3]	R	Erratic Error The core sets this bit to report any erratic errors seen on the UTMI+. Due to erratic errors, the OTG core goes into Suspended state and an interrupt is generated to the application with Early Suspend bit of the Core Interrupt register. If the early suspend is asserted due to an erratic error, the application performs a soft disconnect recover.	1'b0
EnumSpd	[2:1]	R	Enumerated Speed Indicates the speed at which the OTG core has come up after speed detection through a chirp sequence. 2'b00 = High speed (PHY clock is 30 MHz) 2'b01 = Full speed (PHY clock is 30 MHz) 2'b10 = Low speed (PHY clock is 6 MHz) 2'b11 = Full speed (PHY clock is 48 MHz) Low speed is not supported for devices using a UTMI + PHY.	2'b01
SuspSts	[0]	R	Suspend Status In device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspended state if there is no activity on the line_state signal for an extended period of time. The core comes out of the suspend: • If there is any activity on the line_state signal • If the application writes to the Remote Wakeup Signaling bit in the device Control register.	1'b0

This register indicates the status of the core with respect to USB-related events. It must be read on interrupts from device ALL Interrupts (DAINT) register.

32.8.4.4 DIEPMSK

- Base Address: 0x1248_0000
- Address = Base Address + 0x0810, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	22'h0
BNAInIntrMsk	[9]	RW	BNA interrupt mask	1’b0
TxfifoUndrnMsk	[8]	RW	Fifo UNDERRUN mask	1’b0
RSVD	[7]	–	Reserved	1'b0
INEPNakEffMsk	[6]	RW	IN Endpoint NAK effective mask	1'b0
RSVD	[5]	–	Reserved	1'b0
INTknTXFEmpMsk	[4]	RW	IN Token received with TxFIFO empty mask	1'b0
TimeOUTMsk	[3]	RW	Timeout condition mask	1'b0
AHBErrMsk	[2]	RW	AHB error mask	1'b0
EPDisbldMsk	[1]	RW	Endpoint disabled interrupt mask	1'b0
XferComplMsk	[0]	RW	Transfer completed interrupt mask	1'b0

This register works with each of the device IN Endpoint Interrupt registers for all endpoints to generate an interrupt per IN endpoint. The IN endpoint interrupt for a specific status in the DIEPINTn register is masked by writing to the corresponding bit in this register. Status bits are masked by default.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

32.8.4.5 DOEPMSK

- Base Address: 0x1248_0000
- Address = Base Address + 0x0814, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	22'h0
BnaOutIntrMsk	[9]	RW	BNA interrupt mask	1'b0
OutPktErrMsk	[8]	RW	OUT Packet Error mask	1'b0
RSVD	[7]	–	Reserved	–
Back2BackSETup	[6]	RW	Back-to-Back SETUP Packets received mask Applies to control OUT endpoints only.	1'b0
RSVD	[5]	–	Reserved	1'b0
OUTTknEPdisMsk	[4]	RW	OUT Token received when endpoint disabled applies to control out endpoints only.	1'b0
SetUPMsk	[3]	RW	SETUP Phase done mask applies to control endpoints only.	1'b0
AHBErrMsk	[2]	RW	AHB Error	1'b0
EPDisbldMsk	[1]	RW	Endpoint disabled interrupt mask	1'b0
XferComplMsk	[0]	RW	Transfer completed interrupt mask	1'b0

This register works with each of the device OUT Endpoint Interrupt registers for all endpoints to generate an interrupt per OUT endpoint. The OUT endpoint interrupts for a specific status in the DOEPINTn register is masked by writing to the corresponding bit in this register. Status bits are masked by default

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

32.8.4.6 DAINT

- Base Address: 0x1248_0000
- Address = Base Address + 0x0818, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OutEPInt	[31:16]	R	OUT Endpoint Interrupt bits One bit per OUT endpoint: bit[16] for OUT endpoint 0, bit[31] for OUT endpoint 15	16'h0
InEpInt	[15:0]	R	IN Endpoint Interrupt bits One bit per IN endpoint: bit[0] for IN endpoint 0, bit[15] for endpoint 15	16'h0

If a significant event occurs on an endpoint, a device All Endpoints Interrupt register interrupts the application using the device OUT Endpoints Interrupt bit or device IN Endpoints Interrupt bit of the Core Interrupt register. There is one interrupt bit per endpoint, up to a maximum of 16 bits for OUT endpoints and 16 bits for IN endpoints. For a bidirectional endpoint, the corresponding IN and OUT interrupt bits are used. Bits in this register are set and cleared if the application sets and clears bits in the corresponding device Endpoint – n Interrupt register.

32.8.4.7 DAINTMSK

- Base Address: 0x1248_0000
- Address = Base Address + 0x081C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OutEPMsk	[31:16]	RW	OUT EP Interrupt Mask bits One bit per out endpoint: bit[16] for out endpoint 0, bit[31] for out endpoint 15	16'h0
InEpMsk	[15:0]	RW	IN EP Interrupt Mask bits One bit per in endpoint: bit[0] for in endpoint 0, bit[15] for in endpoint 15	16'h0

The device Endpoint Interrupt Mask register works with the device Endpoint Interrupt register to interrupt the application if an event occurs on a device endpoint. However, the Device all Endpoints Interrupt register bit corresponding to that interrupt remains set.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

32.8.4.8 DVBUSDIS

- Base Address: 0x1248_0000
- Address = Base Address + 0x0828, Reset Value = 0x0000_0B8F

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	16'h0
DVBUSDis	[15:0]	RW	Device VBUS Discharge Time Specifies the VBUS discharge time after VBUS pulsing during SRP. This value equals: VBUS discharge time in PHY clocks/1,024	16'h0B8F

This register specifies the VBUS discharge time after VBUS pulsing during SRP.

32.8.4.9 DVBUSPULSE

- Base Address: 0x1248_0000
- Address = Base Address + 0x082C, Reset Value = 0x0000_02C6

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	16'h0
DVBUSPulse	[11:0]	RW	Device VBUS Pulsing Time Specifies the VBUS pulsing time during SRP. This value equals: VBUS pulse time in PHY clocks/1,024	12'h02C6

This register specifies the VBUS discharge time during SRP.

32.8.4.10 DTHRCTL

- Base Address: 0x1248_0000
- Address = Base Address + 0x0830, Reset Value = 0x0C10_0020

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	4'h0
ArbPrkEn	[27]	RW	Arbiter Parking Enable. This bit controls internal DMA arbiter parking for IN endpoints. When thresholding is enabled and this bit is set to one, then the arbiter parks on the IN endpoint for which there is a token received on the USB. This is done to avoid getting into underrun conditions. By default the parking is enabled.	1'b1
RSVD	[26]	–	Reserved	–
RxThrLen	[25:17]	RW	Receive Threshold Length This field specifies Receive thresholding size in DWORDS. This field also specifies the amount of data received on the USB before the core can start transmitting on the AHB. The threshold length has to be at least eight DWORDS. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).	9'h8
RxThrEn	[16]	RW	Receive Threshold Enable When this bit is set, the core enables thresholding in the receive direction.	1'b0
RSVD	[15:13]	–	Reserved	–
AHBThrRatio	[12:11]	RW	AHB Threshold Ratio (AHBThrRatio) These bits define the ratio between the AHB threshold and the MAC threshold for the transmit path only. The AHB threshold always remains less than or equal to the USB threshold, because this does not increase overhead. Both the AHB and the MAC threshold must be DWORD-aligned. If the AHB threshold value is not DWORD-aligned after AHBThrRatio is calculated (based on the application's programming of the MAC threshold and AHBThrRatio), the core automatically aligns to the next lower DWORD value. When programming the TxThrLen and AHBThrRatio, the application must ensure that the minimum AHB threshold value does not go below 8 DWORDS to meet the USB turnaround time requirements. 2'b00 = AHB threshold = MAC threshold 2'b01 = AHB threshold = MAC threshold/2 2'b10 = AHB threshold = MAC threshold/4 2'b11 = AHB threshold = MAC threshold/8	2'h0
TxThrLen	[10:2]	RW	Transmit Threshold Length This field specifies Transmit thresholding size in DWORDS. This field specifies the amount of data in bytes to be in the corresponding endpoint transmit FIFO, before	9'h8

Name	Bit	Type	Description	Reset Value
			the core can start transmit on the USB. The threshold length has to be at least eight DWORDS. This field controls both isochronous and non-isochronous IN endpoint thresholds. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).	
ISOThrEn	[1]	RW	ISO IN Endpoints Threshold Enable When this bit is set, the core enables thresholding for isochronous IN endpoints	1'b0
NonISOThrEn	[0]	RW	Non-ISO IN Endpoints Threshold Enable When this bit is set, the core enables thresholding for Non Isochronous IN endpoints.	1'b0

Thresholding is not supported in Slave mode and so this register must not be programmed in Slave mode.

32.8.4.11 DIEPEMPMSK

- Base Address: 0x1248_0000
- Address = Base Address + 0x1248_0834, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	16'h0
InEpTxfEmpMsk	[15:0]	RW	IN EP Tx FIFO Empty Interrupt Mask Bits These bits acts as mask bits for DIEPINTn. TxFEmp interrupt One bit per IN Endpoint: Bit[0] for IN endpoint 0, bit[15] for IN endpoint 15	16'h0

This register is used to control the IN endpoint FIFO empty interrupt generation (DIEPINTn.TxfEmp).

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

32.8.5 Device Logical Endpoint-Specific Register

A logical endpoint is unidirectional: it is either IN or OUT. To represent a bidirectional endpoint, two logical endpoints are required, one for the IN direction and the other for the OUT direction. This is also true for control endpoints. The registers and register fields described in this section may pertain to IN or OUT endpoints, or both, or specific endpoint types are noted.

32.8.5.1 DIEPCTL0

- Base Address: 0x1248_0000
- Address = Base Address + 0x0900, Reset Value = 0x0000_8000

Name	Bit	Type	Description	Reset Value
EPEna	[31]	R_WS_SC	Endpoint Enable Indicates that data is ready to be transmitted on the endpoint. The core clears this bit before setting any of the following interrupts on this endpoint. • Endpoint Disabled • Transfer Completed	1'b0
EPDis	[30]	R_WS_SC	Endpoint Disable The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.	1'b0
RSVD	[29:28]	–	Reserved	2'b0
SetNAK	[27]	W	Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application controls the transmission of NAK handshakes on an endpoint. The core also sets this bit for an endpoint after a SETUP packet is received on that endpoint.	1'b0
CNAK	[26]	W	Clear NAK A write to this bit clears the NAK bit for the endpoint.	1'b0
TxFNum	[25:22]	R	TxFIFO Number This value is set to the FIFO number that is assigned to IN Endpoint 0.	4'h0
Stall	[21]	R_WS_SC	STALL Handshake The application sets this bit, and the core clears it, if a SETUP token is received for this endpoint. If a NAK bit, Global Non-Periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority.	1'b0
RSVD	[20]	–	Reserved	1'b0
EPType	[19:18]	R	Endpoint Type	2'h0

Name	Bit	Type	Description	Reset Value
			Hardcoded to 00 for control	
NAKsts	[17]	R	NAK Status Indicates the following: 1'b0 = The core is transmitting non-NAK handshakes based on the FIFO status 1'b1 = The core is transmitting NAK handshakes on this endpoint If this bit is set, either by the application or core, the core stops transmitting data, even if there is data available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	1'b0
RSVD	[16]	–	Reserved	1'b0
USBActEP	[15]	R	USB Active Endpoint This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.	1'b1
RSVD	[14:2]	–	Reserved	13'h0
MPS	[1:0]	RW	Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. 2'b00 = 64 bytes 2'b01 = 32 bytes 2'b10 = 16 bytes 2'b11 = 8 bytes	2'h0

This section describes the Control IN Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1-15.

32.8.5.2 DOEPCTL0

- Base Address: 0x1248_0000
- Address = Base Address + 0x0B00, Reset Value = 0x0000_8000

Name	Bit	Type	Description	Reset Value
EPEna	[31]	R_WS_SC	Endpoint Enable When Scatter/Gather DMA mode is enabled, for OUT endpoints this bit indicates that the descriptor structure and data buffer to receive data is setup. • When Scatter/Gather DMA mode is disabled - (such as for buffer-pointer based DMA mode) - this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: • SETUP Phase Done • Endpoint Disabled • Transfer Completed NOTE: In DMA mode, this bit must be set for the core to transfer SETUP data packets into memory.	1'b0
EPDis	[30]	R	Endpoint Disable The application cannot disable control OUT endpoint 0.	1'b0
RSVD	[29:28]	–	Reserved	2'b0
SetNAK	[27]	W	Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application controls the transmission of NAK handshakes on an endpoint. The core sets this bit on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.	1'b0
CNAK	[26]	W	Clear NAK A write to this bit clears the NAK bit for the endpoint.	1'b0
RSVD	[25:22]	–	Reserved	4'h0
Stall	[21]	R_WS_SC	STALL Handshake The application sets this bit, and the core clears it, if a SETUP token is received for this endpoint. If a NAK bit or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	1'b0
Snp	[20]	RW	Snoop Mode This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.	1'b0
EPType	[19:18]	R	Endpoint Type Hardcoded to 2'b00 for control.	2'h0
NAKsts	[17]	R	NAK Status	1'b0

Name	Bit	Type	Description	Reset Value
			Indicates the following: 1'b0 = The core is transmitting non-NAK handshakes based on the FIFO status 1'b1 = The core is transmitting NAK handshakes on this endpoint If application or the core sets this bit, the core stops receiving data, even if there is space in the RxFIFO to accommodate the incoming packet. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake	
RSVD	[16]	–	Reserved	1'b0
USBActEP	[15]	R	USB Active Endpoint This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces.	1'b1
RSVD	[14:2]	–	Reserved	13'h0
MPS	[1:0]	R	Maximum Packet Size The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. 2'b00 = 64 bytes 2'b01 = 32 bytes 2'b10 = 16 bytes 2'b11 = 8 bytes	2'h0

This section describes the Control OUT Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1-15.

32.8.5.3 DIEPCTLn/DOEPCTLn (n = 1 to 15)

- Base Address: 0x1248_0000
- Address = Base Address + 0x0920, + 0x0940, + 0x0960, + 0x0980, + 0x09A0, + 0x09C0, + 0x09E0, + 0x0A00, + 0x0A20, + 0x0A40, + 0x0A60, + 0x0A80, + 0x0AA0, + 0x0AC0, + 0x0AE0, + 0x0B20, + 0x0B40, + 0x0B60, + 0x0B80, + 0x0BA0, + 0x0BC0, + 0x0BE0, + 0x0C00, + 0x0C20, + 0x0C40, + 0x0C60, + 0x0CA0, + 0x0CC0, + 0x0CE0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
EPEna	[31]	R_WS_SC	Endpoint Enable Applies to IN and OUT endpoints. • When Scatter/Gather DMA mode is enabled. • For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. • For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. • When Scatter/Gather DMA mode is enabled-such as for buffer-pointer based DMA mode: – For IN endpoints, this bit indicates that data is ready to be transmitted on the endpoint. – For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: • SETUP Phase Done • Endpoint Disabled • Transfer Completed NOTE: For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.	1'b0
EPDis	[30]	R_WS_SC	Endpoint Disable Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.	1'b0
SetD1PID	[29]	W	Set DATA1 PID Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. This field is applicable both for Scatter/Gather DMA mode and non- Scatter/Gather DMA mode.	1'b0
SetOddFr	–	–	Set Odd (micro) frame Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro) frame (EO_FrNum) field to odd (micro) frame. This field is not applicable for	–

Name	Bit	Type	Description	Reset Value
			Scatter/Gather DMA mode.	
SetD0PID	[28]	W	Set DATA0 PID Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the endpoint Data PID (DPID) field in this register to DATA0. This field is applicable both for Scatter/Gather DMA mode and non- Scatter/Gather DMA mode.	1'b0
SetEvenFr	–	–	In non-Scatter/Gather DMA mode: Set Even (micro) frame (SetEvenFr) Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro) frame (EO_FrNum) field to even (micro) frame. When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is in the transmit descriptor structure. The frame in which to receive data is updated in receive descriptor structure.	–
SNAK	[27]	W	Set NAK Applies to IN and OUT endpoints. A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.	1'b0
CNAK	[26]	W	Clear NAK Applies to IN and OUT endpoints. A write to this bit clears the NAK bit for the endpoint.	1'b0
TxFNum	[25:22]	RW	TxFIFO Number These bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints.	4'h0
Stall	[21]	RW	STALL Handshake Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core.	1'b0
RSVD	–	R_WS_SC	Applies to control endpoints only The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	
Snp	[20]	R/W	Snoop Mode Applies to OUT endpoints only.	1'b0

Name	Bit	Type	Description	Reset Value
			This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.	
EPType	[19:18]	R	Endpoint Type Applies to IN and OUT endpoints. This is the transfer type supported by this logical endpoint. 2'b00 = Control 2'b01 = Isochronous 2'b10 = Bulk 2'b11 = Interrupt	2'h0
NAKsts	[17]	R	NAK Status Applies to IN and OUT endpoints. Indi'ates the following: 1'b0 = The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1 = The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: • The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. • For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. • For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	1'b0
DPID	[16]	R	Endpoint Data PID Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0 = DATA0 1'b1 = DATA1 This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode.	1'b0
EO_FrNum	–	–	Even/Odd (Micro) Frame In non-Scatter/Gather DMA mode: Applies to isochronous IN and OUT endpoints only. Indicates the (micro) frame number in which the core transmits/receives isochronous data for this endpoint. The application must program the even/odd (micro) frame number in which it intends to transmit/receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register.	

Name	Bit	Type	Description	Reset Value
			1'b0 = Even (micro) frame 1'b1 = Odd (micro) frame When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is provided in the transmit descriptor structure. The frame in which data is received is updated in receive descriptor structure.	
USBActEP	[15]	R_W_SC	USB Active Endpoint Applies to IN and OUT endpoints. Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.	1'b0
RSVD	[14:11]	RW	Reserved	4'h0
MPS	[10:0]	RW	Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.	11'h0

The application uses this register to control the behavior of each logical endpoint other than endpoint 0.

32.8.5.4 DIEPINTn/DOEPINTn (n = 0 to 15)

- Base Address: 0x1248_0000
- Address = Base Address + 0x0908, + 0x0928, + 0x0948, + 0x0968, + 0x0988, + 0x09A8, + 0x09C8, + 0x09E8, + 0x0A08, + 0x0A28, + 0x0A48, + 0x0A68, + 0x0A88, + 0x0AA8, + 0x0AC8, + 0x0AE8, + 0x0B08, + 0x0B28, + 0x0B48, + 0x0B68, + 0x0B88, + 0x0BA8, + 0x0BC8, + 0x0BE8, + 0x0C08, + 0x0C28, + 0x0C48, + 0x0C68, + 0x0C88, + 0x0CA8, + 0x0CC8, + 0x0CE8, Reset Value = 0x0000_0080

Name	Bit	Type	Description	Reset Value
EPEna	[31:15]	–	Reserved	17'h0
NYETIntrpt	[14]	R_SS_WC	NYET interrupt (NYETIntrpt) The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint.	1'b0
NAKIntrpt	[13]	R_SS_WC	NAK interrupt (NAKIntrpt) The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TxFIFO.	1'b0
BbleErrIntrpt	[12]	R_SS_WC	BbleErr (Babble Error) interrupt (BbleErrIntrpt) The core generates this interrupt when babble is received for the endpoint.	1'b0
Packet Dropped Status	[11]	R_SS_WC	PktDrpSts (Packet Dropped Status) This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt. Dependency: This bit is valid in non Scatter/Gather DMA mode when periodic transfer interrupt feature is selected.	1'b0
RSVD	[10]	–	Reserved	–
BNAIntr	[9]	R_SS_WC	Buffer Not Available Interrupt This bit is valid only when Scatter/Gather DMA mode is enabled. The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done	–
TxfifoUndrn	[8]	R_SS_WC	Fifo Underrun Applies to IN endpoints Only This bit is valid only when thresholding is enabled. The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint.	1’b0
OutPktErr		–	OUT Packet Error Applies to OUT endpoints Only This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet.	
TxFEmp	[7]	–	Transmit FIFO Empty This bit is valid only for IN Endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the	

Name	Bit	Type	Description	Reset Value
			TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)).	
INEPNakEff	[6]	R	IN Endpoint NAK Effective Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit.	1'b0
Back2Back SETup		RW	Back-to-Back SETUP Packets Receive Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt.	–
INTknEPMis	[5]	R_SS_WC	IN Token Received with EP Mismatch Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received.	1'b0
StsPhseRcvd		–	Status Phase Received For Control Write This interrupt is valid only for Control OUT endpoints and only in Scatter Gather DMA mode. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in case of Scatter Gather DMA mode.	
INTknTXFEmp	[4]	R_SS_WC	IN Token Received When TxFIFO is Empty Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received.	1'b0
OUTTknEPdis		–	OUT Token Received When Endpoint Disabled Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received.	
TimeOUT	[3]	R_SS_WC	Timeout Condition • In dedicated FIFO mode, applies only to Control IN	1'b0

Name	Bit	Type	Description	Reset Value
			endpoints. • In Scatter/Gather DMA mode, the TimeOUT interrupt is not asserted. Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint.	
SetUp		–	SETUP Phase Done Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet.	–
AHBErr	[2]	R_SS_WC	AHB Error Applies to IN and OUT endpoints. This is generated only in Internal DMA mode if there is an AHB error during an AHB read/write. The application reads the corresponding endpoint DMA address register to get the error address.	1'b0
EPDisbld	[1]	R_SS_WC	Endpoint Disabled Interrupt Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.	1'b0
XferCompl	[0]	R_SS_WC	Transfer Completed Interrupt (XferCompl) Applies to IN and OUT endpoints. • When Scatter/Gather DMA mode is enabled • For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. • For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. • When Scatter/Gather DMA mode is disabled, this field indicates that the programmed transfer is complete on the AHB as well as on the USB, for this endpoint.	1'b0

This register indicates the status of an endpoint with respect to USB- and AHB-related events. The application must read this register if the OUT Endpoints Interrupt bit or IN Endpoints Interrupt bit of the Core Interrupt register is set. Before the application reads this register, it must first read the device All Endpoints Interrupt (DAINT) register to get the exact endpoint number for the device Endpoint-n Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the DAINT and GINTSTS registers.

32.8.5.5 DIEPTSIZ0

- Base Address: 0x1248_0000
- Address = Base Address + 0x0910, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:21]	–	Reserved	11'h0
PktCnt	[20:19]	RW	Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for endpoint 0. This field is decremented every time a packet is read from the TxFIFO.	2'b0
RSVD	[18:7]	–	Reserved	12'h0
XferSize	[6:0]	RW	Transfer Size Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet from the external memory is written to the TxFIFO.	7'h0

The application must modify this register before enabling endpoint 0. Once endpoint 0 is enabled using Endpoint Enable bit of the device Control Endpoint 0 Control registers (DIEPCTL0.EPEna/DOEPCTL0.EPEna), the core modifies this register. The application can only read this register once the core has cleared the Endpoint Enable bit.

Nonzero endpoints use the registers for endpoints 1-15.
When Scatter/Gather DMA mode is enabled, this register must not be programmed by the application. If the application reads this register when Scatter/Gather DMA mode is enabled, the core returns all zeros.

32.8.5.6 DOEPTSIZ0

- Base Address: 0x1248_0000
- Address = Base Address + 0x0B10, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	1'b0
SUPCnt	[30:29]	RW	SETUP Packet Count This field specifies the number of back-to-back SETUP data packets the endpoint can receive. 2'b01 = 1 Packet 2'b10 = 2 Packets 2'b11 = 3 Packets	2'h0
RSVD	[28:21]	–	Reserved	9'h0
PktCnt	[20:19]	RW	Packet Count This field is decremented to zero after a packet is written into the RxFIFO.	2'b0
RSVD	[18:7]	–	Reserved	12'h0
XferSize	[6:0]	RW	Transfer Size Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet is read from RxFIFO and written to the external memory.	7'h0

32.8.5.7 DIEPTSIZn/DOEPTSIZn (n = 0 to 15)

- Base Address: 0x1248_0000
- Address = Base Address + 0x0910, + 0x0930, + 0x0950, + 0x0970, + 0x0990, + 0x09B0, + 0x09D0, + 0x09F0, + 0x0A10, + 0x0A20, + 0x0A40, + 0x0A80, + 0x0AB0, + 0x0AD0, + 0x0AF0, + 0x0B10, + 0x0B30, + 0x0B50, + 0x0B70, + 0x0B90, + 0x0BB0, + 0x0BD0, + 0x0BF0, + 0x0C10, + 0x0C30, + 0x0C50, + 0x0C70, + 0x0C90, + 0x0CB0, + 0x0CD0, + 0x0CF0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Reserved	[31]	RW	Reserved	1'b0
MC	[30:29]	RW	Multi Count Applies to IN endpoints only. For periodic IN endpoints, this field indicates the number of packets that must be transmitted per microframe on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints. 2'b01 = 1 packet 2'b10 = 2 packets 2'b11 = 3 packets	
		R	For non-periodic IN endpoints, this field is valid only in Internal DMA mode. It specifies the number of packets the core must fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the device Endpoint-n Control register (DIEPCTLn.NextEp).	
RxDPID		R	Received Data PID Applies to isochronous OUT endpoints only. This is the data PID received in the last packet for this endpoint. 2'b00 = DATA0 2'b01 = DATA1 2'b10 = DATA2 2'b11 = MDATA	2'b0
SUPCnt		RW	SETUP Packet Count Applies to control OUT Endpoints only. This field specifies the number of back-to-back SETUP data packets the endpoint can receive. 2'b01 = 1 packet 2'b10 = 2 packets 2'b11 = 3 packets	
PktCnt	[28:19]	RW	Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for this endpoint. • IN Endpoints: This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO. • OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the RxFIFO.	10'h0

Name	Bit	Type	Description	Reset Value
XferSize	[18:0]	RW	Transfer Size This field contains the transfer size in bytes for the current endpoint. The core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. • IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO. • OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.	19'h0

The application must modify this register before enabling the endpoint. Once the endpoint is enabled using Endpoint Enable bit of the device Endpoint-n Control registers (DIEPCTLn.EPEna/DOEPCTLn.EPEna), the core modifies this register. The application can only read this register once the core has cleared the Endpoint Enable bit. This register is used only for endpoints other than Endpoint 0.
When Scatter/Gather DMA mode is enabled, this register must not be programmed by the application. If the application reads this register when Scatter/Gather DMA mode is enabled, the core returns all zeros

32.8.5.8 DIEPDMAn/DOEPDMAn (n = 0 to 15)

- Base Address: 0x1248_0000
- Address = Base Address + 0x0914, + 0x0934, + 0x0954, + 0x0974, + 0x0994, + 0x09B4, + 0x09D4, + 0x09F4, + 0x0A14, + 0x0A34, + 0x0A54, + 0x0A74, + 0x0A94, + 0x0AB4, + 0x0AD4, + 0x0AF4, + 0x0B14 + 0x0B34, + 0x0B54, + 0x0B74, + 0x0B94, + 0x0BB4, + 0x0BD4, + 0x0BF4, + 0x0C14, + 0x0C34, + 0x0C54, + 0x0C74, + 0x0C94, + 0x0CB4 + 0x0CD4, + 0x0CF4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DMAAddr	[31:0]	RW	DMA Address Holds the start address of the external memory for storing or fetching endpoint data. NOTE: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. • When Scatter/Gather DMA mode is not enabled, When Scatter/Gather DMA mode is not enabled, the application programs the start address value in this field. • When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.	32'h0

The starting DMA address must be DWORD-aligned.

32.8.5.9 DTXFSTSn (n = 0 to 15)

- Base Address: 0x1248_0000
- Address = Base Address + 0x0918, + 0x0938, + 0x0958, + 0x0978, + 0x0998, + 0x09B8, + 0x09D8, + 0x09F8, + 0x0A18, + 0x0A38, + 0x0A58, + 0x0A78, + 0x0A98, + 0x0AB8, + 0x0AD8, + 0x0AF8, Reset Value = 0x0000_0100

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	16'h0
INEPTxFSpcAvail	[15:0]	R	IN Endpoint TxFIFO Space Avail Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. 16'h0 = Endpoint TxFIFO is full 16'h1 = 1 word available 16'h2 = 2 words available 16'hn = n words available (where 0 ≤ n ≤ 32,768) 16'h8000 = 32,768 words available Others = Reserved	16'h100

This read-only register contains the free space information for the device IN endpoint TxFIFO.

32.8.5.10 DIEPDMABn/DOEPDMABn (n = 0 to 15)

- Base Address: 0x1248_0000
- Address = Base Address + 0x091C, + 0x093C, + 0x095C, + 0x097C, + 0x099C, + 0x09BC, + 0x09DC, + 0x09FC, + 0x0A1C, + 0x0A3C, + 0x0A5C, + 0x0A7C, + 0x0A7C, + 0x 0A9C, + 0x0ABC, + 0x0ADC, + 0x0AFC, + 0x0B1C, + 0x0B3C, + 0x0B5C, + 0x0B7C, + 0x0B9C, + 0x0BBC, + 0x0BDC, + 0x0BFC, + 0x0C1C, + 0x0C3C, + 0x0C5C, + 0x0C7C, + 0x0C9C, + 0x0CBC, + 0x0CDC, + 0x0CFC,
  Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DMABufferAddr	[31:0]	R	DMA Buffer Address Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.	32'h0

These fields are present only in case of Scatter/Gather DMA.

33 Transport Stream Interface

33.1 Overview

The Transport Stream Interface (TSI) receives transport stream data from channel chip, which it writes to a specific address of the output buffer (SDRAM). Depending on the bus bandwidth, TSI has 32 words FIFO.

Using word-aligned address, the TSI sends data streams to the SDRAM. One packet size is equal to 47 words. The output buffer size should be equal to a multiple of 47 words (one packet size).

If the data is written in the output buffer, the SDRAM full interrupt occurs.

33.1.1 Features

The features of TSI are:

- Writes transport stream received from channel chip to output buffer (supports 1-/4-/8-beat burst, word-aligned)
- Supports TS interface in DVB-H/DVB-T/ISDB-T/T-DMB/DAB mode
- Supports TS_CLK falling/rising edge data fetch mode
- Supports active high or active low mode for TS signals (TS_VALID, TS_SYNC, and TS_ERROR)
- Supports MSB to LSB or LSB to MSB data byte order
- Specifies the maximum size of output buffer for store transport stream as 256KBytes
- Supports two sync detecting modes:
  - TS_SYNC signal
  - Sync byte
- Supports Program ID (PID) filter mode with 32 PID filters
- Supports six error cases with SKIP/STOP mode
- Supports TS_CLK filter.
  - TS_CLK maximum frequency
    - with TS_CLK filter: to 1/2 HCLK
    - without TS_CLK filter: to 1/4 HCLK

33.1.2 Broadcast Mode

TSI supports DVB-H, DVB-T, ISDB-T, T-DMA, and DAB modes. *Figure 33-1* illustrates an example of the broadcasting support scheme.

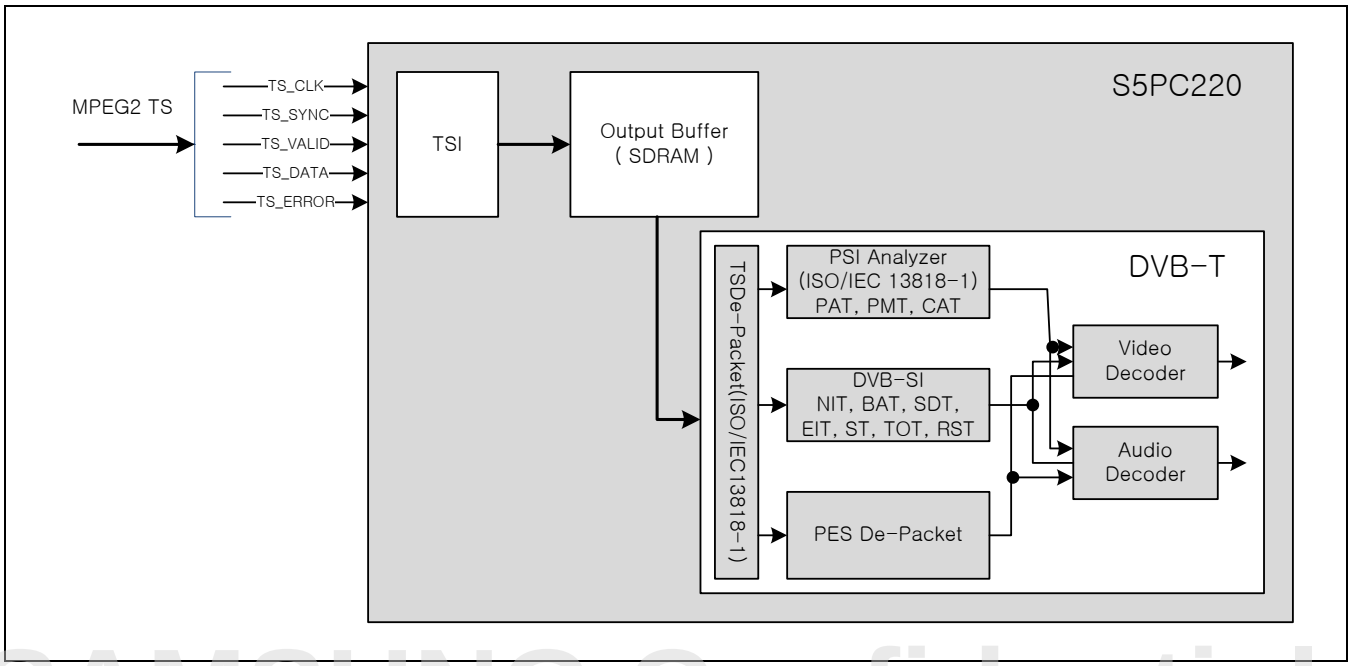


Figure 33-1 Support TSI in the Broadcasting Mode

The MPEG-2 transport stream contains a set of service information for Program Specific Information (PSI). External channel chip injects the transport stream and TSI stores it into output buffer.
Then, TS De-Packet, PSI Analyzer, DVB-SI, and PED De-Packet blocks de-multiplex individual PSI from the transport stream and it transfers these packets to audio and video decoders.

PSI comprises Program Association Table (PAT), Program Map Table (PMT), and Conditional Access Table (CAT), defined in MPEG-2.

- PAT: Transmits the PID information of PMT and NID, and information about various services offering in transmitter.
- PMT: Transmit the PID information of transmit port packet and PCR information transferred with various services.
- CAT: Transmits information for charging broadcasting system used in transmitter.

Table 33-1 lists the characteristics of several mobile-TV standard modes.

Table 33-1 Characteristics of Several Mobile-TV Standard Modes

Mobile TV Standard	Video Codec	Characteristics
DVB-T	MPEG-2	MP @ ML 10 Mbps, 720 × 480 @ 30 fps
DVB-H	H.264	Up to Baseline 352 × 288 @ 15 fps @ 384 kbps
	WMV9	Up to SP @ ML 356 × 288 @ 15 fps @ 384 kbps
T-DMB	H.264	Up to Baseline Profile 352 × 288 @ 30 fps @ 768 kbps
ISDB-T	H.264	Up to Baseline Profile 320 × 240 @ 15 fps @ 384 kbps

33.1.3 Block Diagram

Figure 33-2 illustrates the overall functional block diagram of TS Interface.

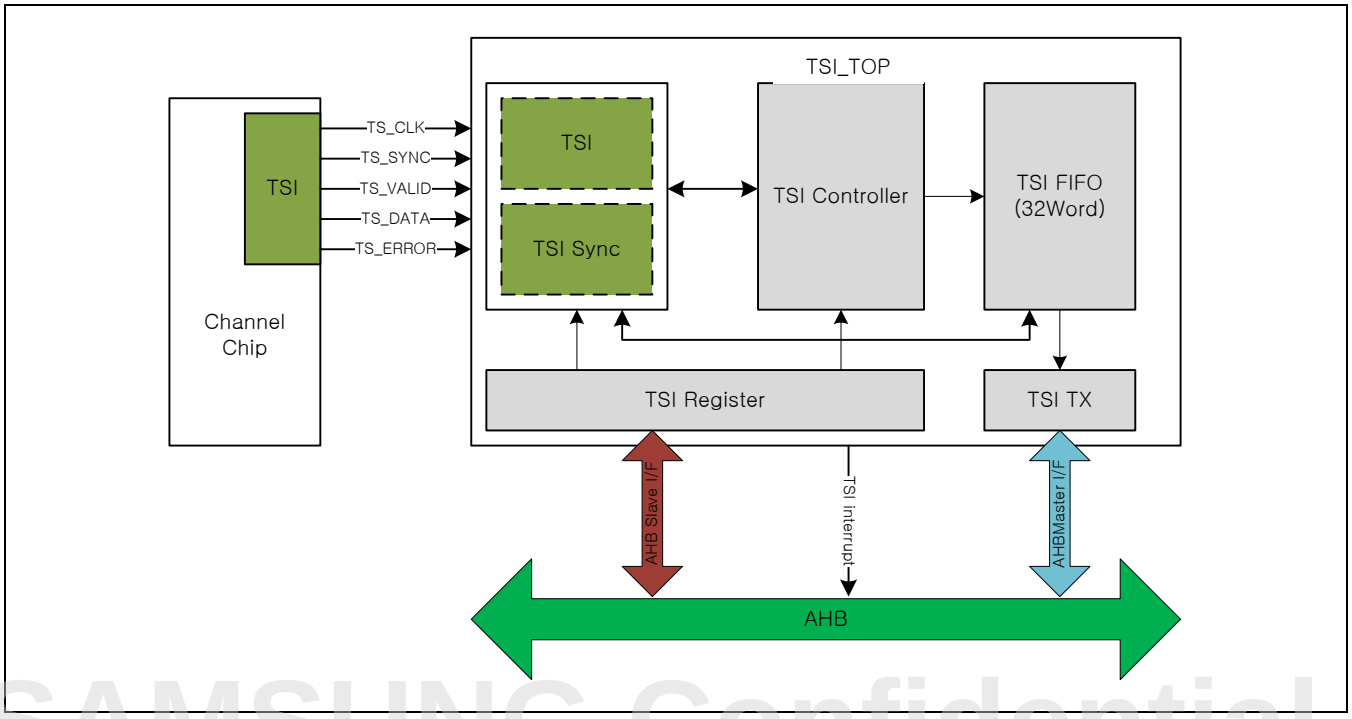


Figure 33-2 TSI Block Diagram

External channel chip transfers the transport stream data through five signals. The five signals are:

- TS_CLK
- TS_SYNC
- TS_VALID
- TS_DATA
- TS_ERROR

TSI controller receives transport stream data by capturing the five signals with TSI synchronizer and TSI sync detector. The received transport stream data is stored on TSI FIFO, TSI TX module transfers the transport stream data into output buffer using AHB master interface. You can control operations of TSI by setting TSI registers on AHB slave interface.

33.1.4 I/O Description of TSI

Table 33-2 describes the I/O ports of TSI.

Table 33-2 TSI I/O Description

Signal	I/O	Description (Primary Function)	Pad	Type
TS_CLK	I	Specifies the TSI system clock (66 MHz)	XmdmWEn	Muxed
TS_SYNC	I	Specifies the TSI synchronization control signal	XmdmCSn	Muxed
TS_VAL	I	Specifies the TSI valid signal	XmdmRn	Muxed
TS_DATA	I	Specifies the TSI input data	XmdmIRQn	Muxed
TS_ERROR	I	Specifies the TSI error indicate signal	XmdmADVN	Muxed

33.1.5 Functional Description

The functional description section includes:

- TSI Operation
- Transport Stream Signals
- Sync Detection

33.1.5.1 TSI Operation

The TS packet has 188 bytes and it consists of the header, adaptation field, and payload. The TS header includes a Packet Identifier (PID), sync byte, and several control signals (for example, transport scrambling, adaptation field, and continuity control).

Figure 33-3 illustrates the transport stream packet data format.

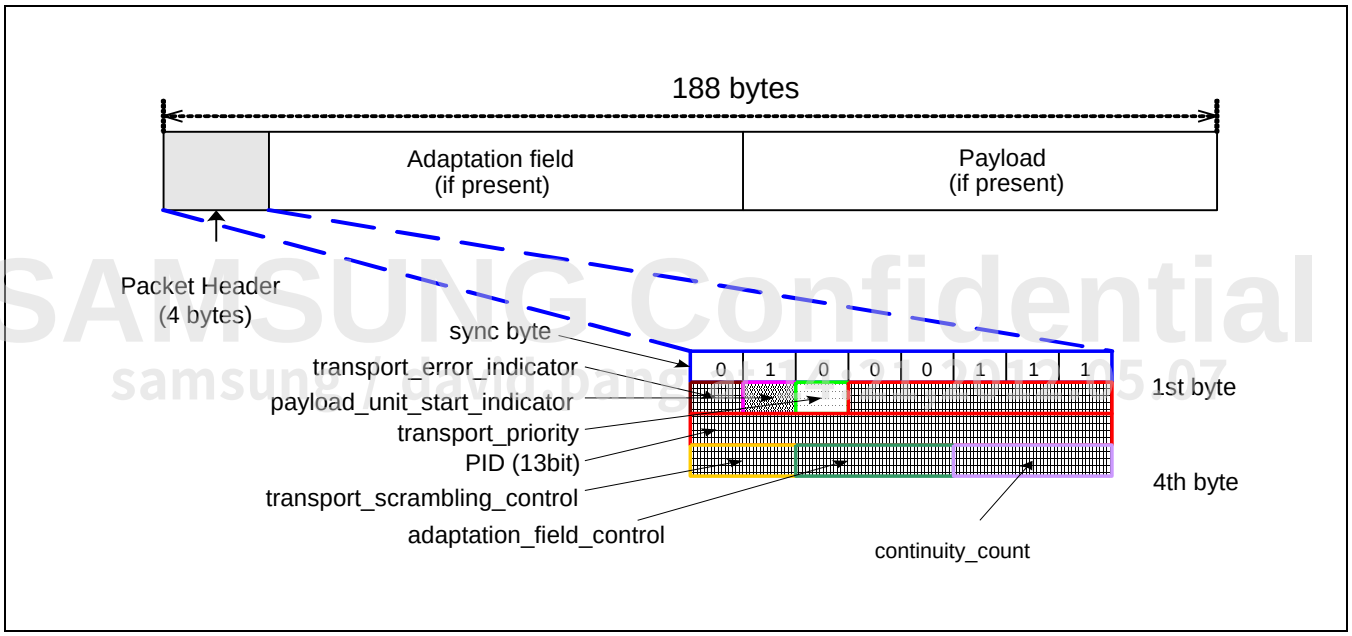


Figure 33-3 Transport Stream Packet Data Format

33.1.5.2 Transport Stream Signals

You can set the transport stream signals (TS_CLK, TS_SYNC, TS_VALID, TS_ERROR, and TS_DATA) as active-high or active-low. You can also set the active mode of each signal independently.

Figure 33-4 illustrates the timing diagram of transport stream signals and describes the timing operation of each signal.

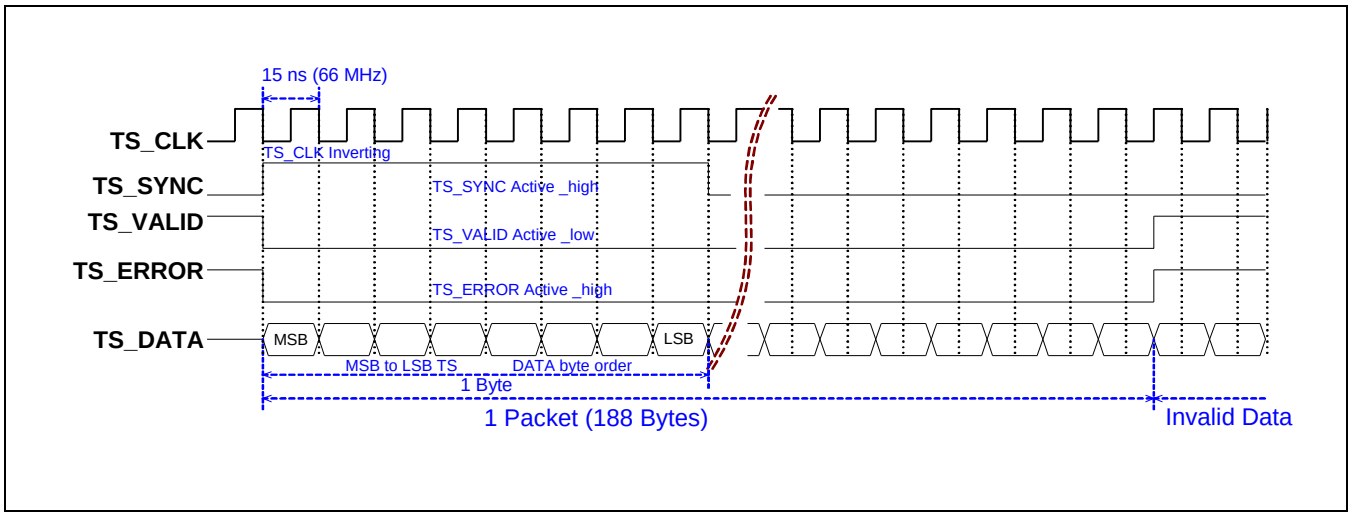


Figure 33-4 Transport Stream Signals

33.1.5.3 Sync Detection

It does the sync detection of transport stream using TS_SYNC signal and sync byte.

33.1.5.3.1 Using TS_SYNC Signal

It does the sync detection of transport stream using TS_SYNC signal in two different bits:

- Considering the consecutive 8 bits TS_SYNC signal
- Observing only 1-bit TS_SYNC signal

Figure 33-5 illustrates the sync detection using TS_SYNC signal.

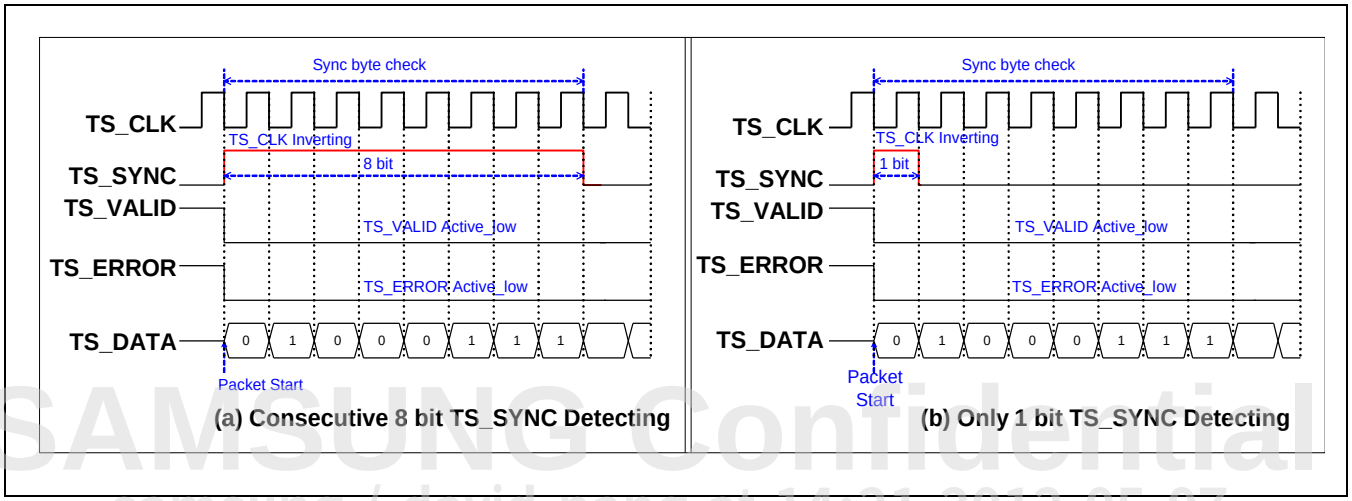


Figure 33-5 Sync Detection Using TS_SYNC Signal

33.1.5.3.2 Using Sync Byte

When sync detection using sync byte (0x47) is done, the 0x47 value is transferred in the middle of transport stream. This value can be wrongly recognized as the starting point of the packet. To prevent this situation, when the sync byte is set at TSI sync control register (sync_det_cnt (0 to 15)), TSI recognizes it as the starting point of the packet.

Three data counters and three current sync detecting counters are used in the TSI module. The current sync detecting counter displays the sync byte that is being input up to that time. The data detecting counter displays the size of the transferring transport stream after the input of sync byte.

Consider that data counter is equal to 187 bytes. If the sync byte is inputted, it initializes the data counter by zero and increases the sync detecting counter by one. On the other hand, when the input data is not the sync byte, it disables the data counter and initializes the sync detecting counter by zero.

Consider that enabled total data counter is less than 187. If the sync byte is inputted, it enables the data counter and increases the related sync detecting counter by one.

When the register value of one among the three sync detecting counters is the same as the value of TSI sync control register (sync_det_cnt), the sync detecting operation is completed.

Figure 33-6 illustrates the example of the sync detecting operation.

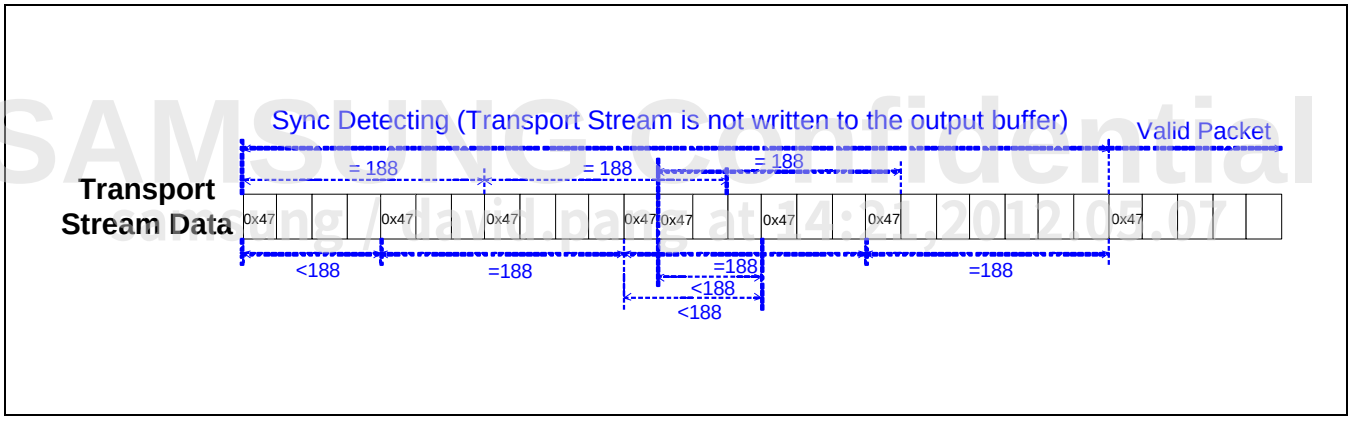


Figure 33-6 Sync Detection Using Sync Byte (sync_det_cnt = 3)

Table 33-3 lists the process of the sync detecting operation.

Table 33-3 Sync Detection Process using Sync Byte (sync_det_cnt = 3)

#	Condition	Data Count	Sync Detecting Count
–	Sync detecting idle	Disables all data count	All sync detecting count set to 0
1	Sync byte input	Enables Data count1	csdc1 = 1
2	Sync byte input, data count1 < 187	Enables Data count2	csdc2 = 1
3	Sync byte input, data count1 == 187	Data count1 set to 0	csdc1 = 2
4	Sync byte input, data count2 == 187	Data count2 set to 0	csdc2 = 2
5	Sync byte input, data count1 < 87, data count2 < 187	Enables Data count3	csdc3 = 1
6	Sync byte not input, data count1 == 187	Disables Data count1	csdc1 = 0
7	Sync byte input, data count2 < 87, data count3 < 187	Enables Data count1	csdc1 = 1
8	Sync byte input, data count2 == 187	Data count2 set to 0	csdc2 = 3
9	Sync byte not input, data count3 == 187	Disables Data count3	csdc3 = 0
10	Sync byte input, data count2 == 187	Data count2 set to 0	csdc2 == 4. sync detecting complete

33.1.5.4 Error Detection

TSI can generate six errors, as described in *Table 33-4*. Each error occurs at the SKIP mode or STOP mode.

Whenever an error occurs at the SKIP mode, TSI generates an interrupt. Only after fixing the error (Padding or Skip) at the packet where the error occurred, TSI continues to send data streams to the output buffer.

Whenever an error occurs at the STOP mode, TSI generates an interrupt and stops sending data streams to the output buffer.

In case the error is caused by SKIP or STOP mode in the TSI control register, the interrupt flag in TSI interrupt register is set to 1.

If the TSI mask register is enabled and if an error occurs, the error interrupt signal is sent to the MCU.

Table 1-4 describes the sync detection process using sync byte (sync_det_cnt = 3).

Table 33-4 Sync Detection Process Using Sync Byte (sync_det_cnt = 3)

#	Error Case		Description
1	Sync mismatch (only in TS_SYNC mode)		Sync byte is not received at the start of packet
2	Packet size underflow (only in TS_SYNC mode)		TS_SYNC signal is activated when the packet reception does not end
3	Packet size overflow	TS_SYNC mode	TS_SYNC signal is deactivated at the start of packet
		Sync byte mode	Sync byte is not received at the start of packet
4	TS_ERROR	Bit detecting	TS_ERROR signal is activated when the packet reception is operating.
		Packet detecting	TS_ERROR signal is activated for receiving one packet (Packet data is written to the OUTPUT buffer. Only error flag is generated).
5	TS_CLK timeout (only STOP mode)		TS_CLK is not toggled for n-cycles of HCLK (n is determined by "TSI CLOCK COUNT register").
6	Internal FIFO full (only STOP mode)		Internal FIFO is filled with received data

Figure 33-7 illustrates the timing diagram of several TSI error cases. *Figure 33-7*-(a) illustrates the sync mismatch case. *Figure 33-7*-(b) illustrates the packet size underflow case. *Figure 33-7*-(c) and *Figure 33-7*-(d) illustrate the packet size overflow case. *Figure 33-7*-(e), *Figure 33-7*-(f), and *Figure 33-7*-(g) illustrate the ts_error case.

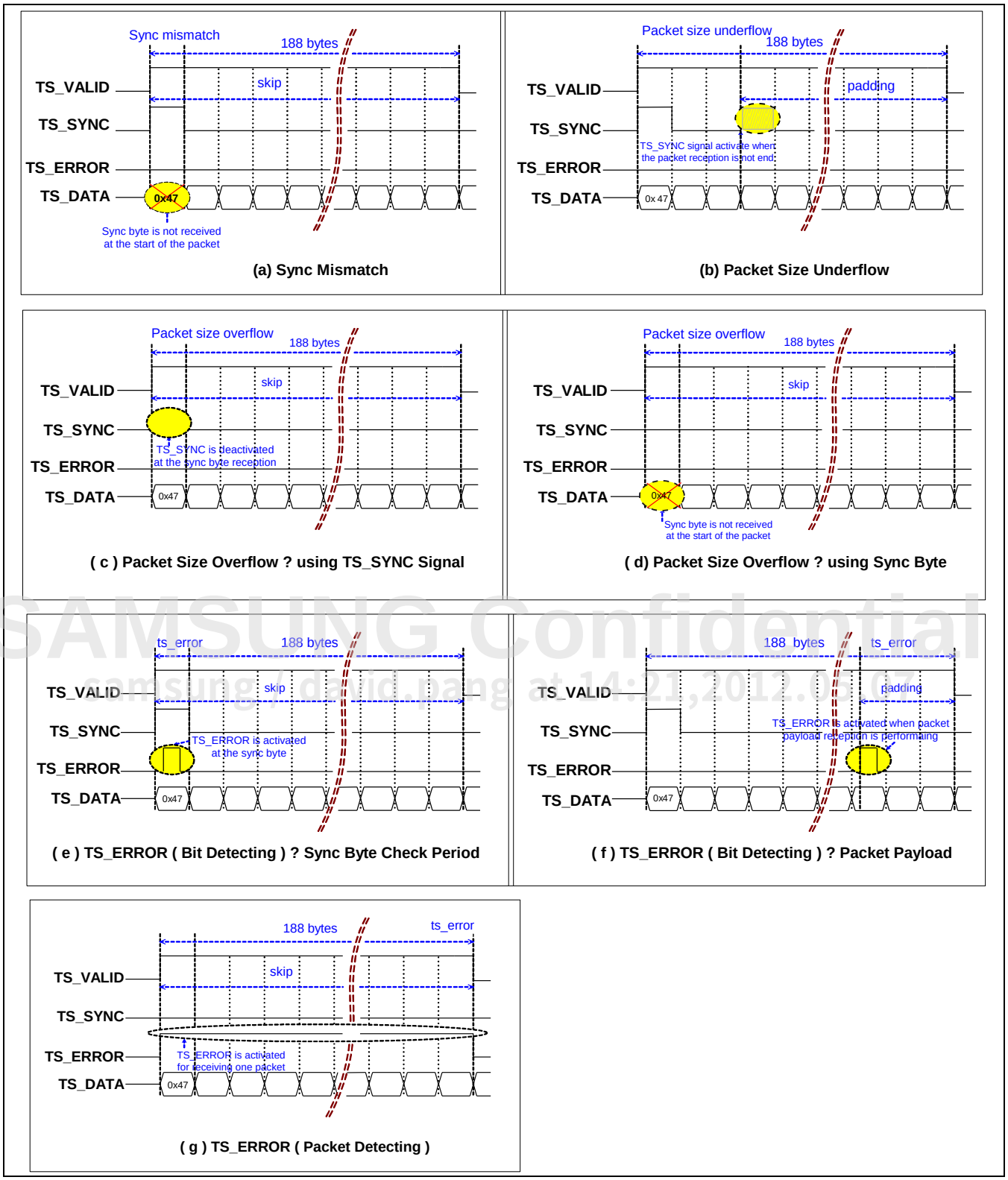


Figure 33-7 TSI Error Cases (with Skip Mode, TS_VALID/TS_SYNC/TS_Error is Active High)

33.1.5.5 TS_CLK Filter

When HCLK maintains the same level as TS_CLK at the rising edge during two consecutive clock cycles, TS_CLK filter considers TS_CLK as valid.

In case TS_CLK filter is used, a half-period of the TS_CLK should have two or more periods of HCLK. The maximum frequency of TS_CLK changes depending on whether TS_CLK passes two-stage flip-flops (noise filter is disabled) or TS_CLK passes three-stage flip-flops (noise filter is enabled).

If HCLK with an operating frequency of 132 MHz is used, the maximum frequency of TS_CLK is limited by 66 MHz (without TS_CLK filter) and 33 MHz (with TS_CLK filter).

Figure 33-8 illustrates the block diagram of TS_CLK filter.

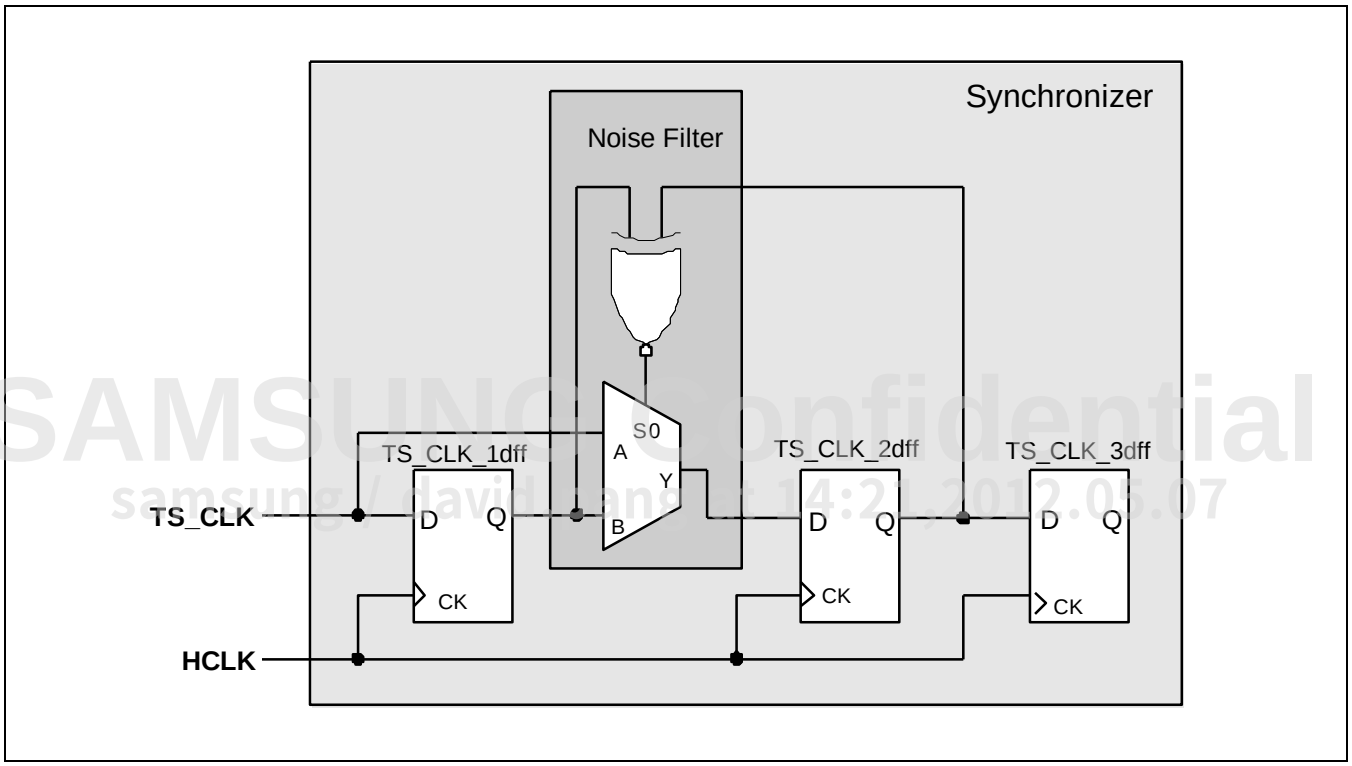


Figure 33-8 Block Diagram of TS_CLK Filter

33.1.5.6 Transport Stream Write

After the transport stream is received from the channel chip, it is stored in an internal FIFO (32-word). The TSI sends the transport stream to the output buffer (DRAM) depending on the selected burst length (1-/4-/8-beat). Using word-aligned address, the TSI sends data streams to the output buffer (SDRAM). One packet size is equal to 47 words. The output buffer size should be equal to a multiple of 47 words (one packet size). If the data is written in the output buffer, the SDRAM full interrupt is generated and the destination address is reloaded in the base address.

33.1.5.7 Program ID filter

TSI supports a PID filter mode that uses 32 PID filters. It can switch on/off PID filter independently. Consider that PID filter mode enabled. In case of the transmitted PID header, packet is filtered by one of the 32 filters. In case the PID of transmitted packet header is one of the 32 filters, the transport stream is treated as normal and stored to output buffer. However, if the PID filter mode is disabled, the PID filter value is not checked and all transport streams are recognized as normal.

33.1.5.8 TSI Control FSM

TSI has several operating modes. *Figure 33-9* illustrates how the TSI can switch from one mode to another according to the condition (state) of control signals.

Figure 33-9 illustrates the TSI control Finite State Machine (FSM).

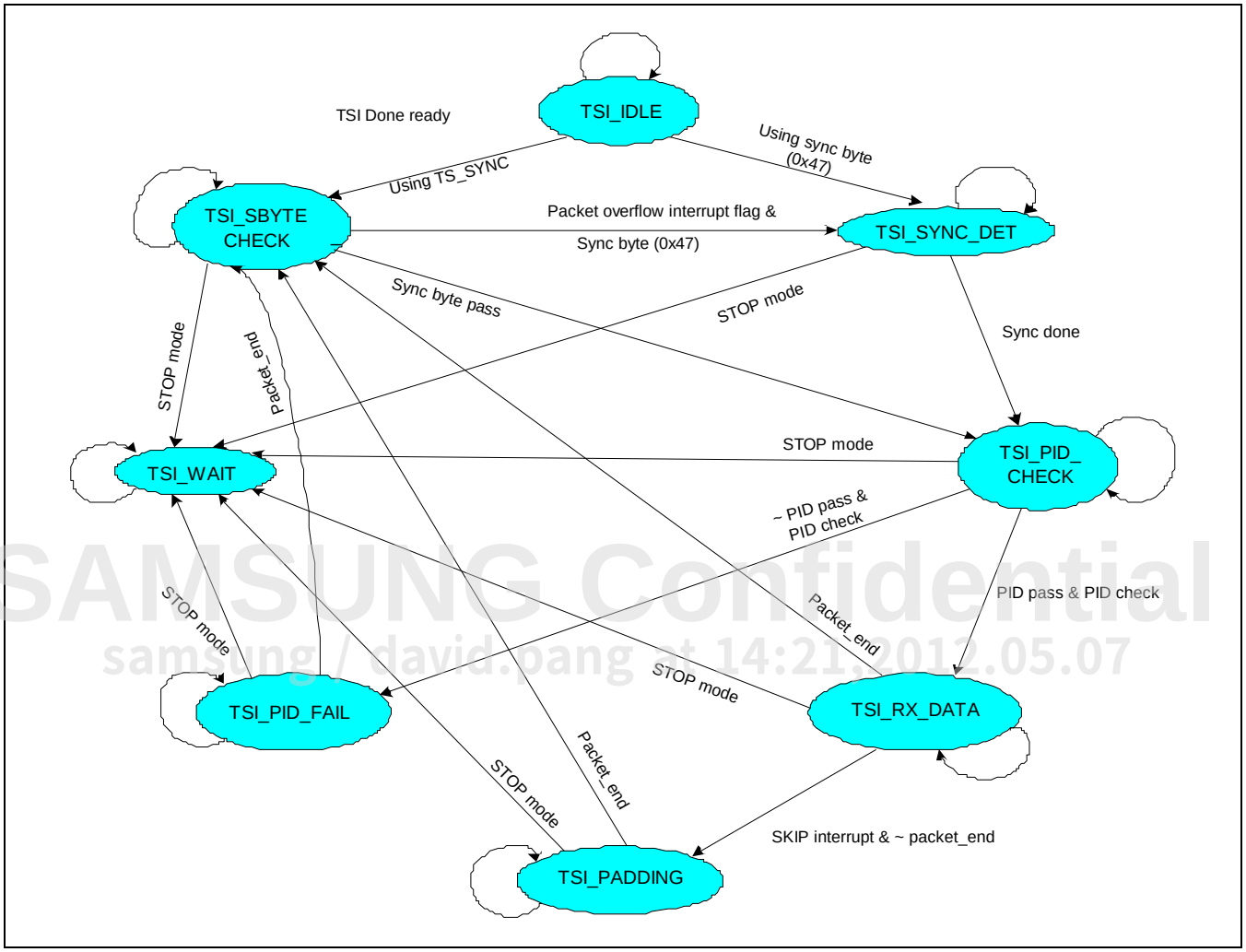


Figure 33-9 TSI Control Finite State Machine (FSM)

33.1.5.9 Shadow Base Address

TSI automatically changes the destination address of SDRAM with base address. When TSI is started (tsi_on = 1) or output buffer full interrupt is generated, address value of TS_BASE register is set as new destination address to store received packet data.

An example of Shadow Base Address usage is:

1. Set the first address in TS_BASE register.
2. Start TSI (the first address is set as destination address).
3. Set the second address in TS_BASE register right after TSI starts. Output buffer becomes full (The second address is set as new destination address).
4. Set the third address in TS_BASE register when output buffer full interrupt is generated. When the output buffer becomes full, (the third address is set as new destination address).

33.2 Register Description

33.2.1 Register Map Summary

- Base Address: 0x1250_0000

Register	Offset	Description	Reset Value
TS_CLKCON	0x0000	Specifies the TSI clock control register	0x0000_0002
TS_CON	0x0004	Specifies the TSI control register	0x1000_0000
TS_SYNC	0x0008	Specifies the TSI sync control register	0x0000_00F0
TS_CNT	0x000C	Specifies the TSI clock count register	0x00FF_FFFF
TS_BASE	0x0010	Specifies the TS buffer base address	0x0000_0000
TS_SIZE	0x0014	Specifies the TS buffer size address	0x0000_0000
TS_CADDR	0x0018	Specifies the TS buffer current write address	0x0000_0000
TS_INT_MASK	0x001C	Specifies the TSI interrupt mask register	0x0000_0000
TS_INT	0x0020	Specifies the TSI interrupt flag register	0x0000_0000
TS_PID0	0x0024	Specifies the TSI PID filter0	0x0000_0000
TS_PID1	0x0028	Specifies the TSI PID filter1	0x0000_0000
TS_PID2	0x002C	Specifies the TSI PID filter2	0x0000_0000
TS_PID3	0x0030	Specifies the TSI PID filter3	0x0000_0000
TS_PID4	0x0034	Specifies the TSI PID filter4	0x0000_0000
TS_PID5	0x0038	Specifies the TSI PID filter5	0x0000_0000
TS_PID6	0x003C	Specifies the TSI PID filter6	0x0000_0000
TS_PID7	0x0040	Specifies the TSI PID filter7	0x0000_0000
TS_PID8	0x0044	Specifies the TSI PID filter8	0x0000_0000
TS_PID9	0x0048	Specifies the TSI PID filter9	0x0000_0000
TS_PID10	0x004C	Specifies the TSI PID filter10	0x0000_0000
TS_PID11	0x0050	Specifies the TSI PID filter11	0x0000_0000
TS_PID12	0x0054	Specifies the TSI PID filter12	0x0000_0000
TS_PID13	0x0058	Specifies the TSI PID filter13	0x0000_0000
TS_PID14	0x005C	Specifies the TSI PID filter14	0x0000_0000
TS_PID15	0x0060	Specifies the TSI PID filter15	0x0000_0000
TS_PID16	0x0064	Specifies the TSI PID filter16	0x0000_0000
TS_PID17	0x0068	Specifies the TSI PID filter17	0x0000_0000
TS_PID18	0x006C	Specifies the TSI PID filter18	0x0000_0000
TS_PID19	0x0070	Specifies the TSI PID filter19	0x0000_0000
TS_PID20	0x0074	Specifies the TSI PID filter20	0x0000_0000
TS_PID21	0x0078	Specifies the TSI PID filter21	0x0000_0000
TS_PID22	0x007C	Specifies the TSI PID filter22	0x0000_0000
TS_PID23	0x0080	Specifies the TSI PID filter23	0x0000_0000

Register	Offset	Description	Reset Value
TS_PID24	0x0084	Specifies the TSI PID filter24	0x0000_0000
TS_PID25	0x0088	Specifies the TSI PID filter25	0x0000_0000
TS_PID26	0x008C	Specifies the TSI PID filter26	0x0000_0000
TS_PID27	0x0090	Specifies the TSI PID filter27	0x0000_0000
TS_PID28	0x0094	Specifies the TSI PID filter28	0x0000_0000
TS_PID29	0x0098	Specifies the TSI PID filter29	0x0000_0000
TS_PID30	0x009C	Specifies the TSI PID filter30	0x0000_0000
TS_PID31	0x00A0	Specifies the TSI PID filter31	0x0000_0000
BYTE_SWAP	0x00BC	Specifies the TSI Tx byte swap enable register for little endian	0x0000_0001

33.2.1.1 TS_CLKCON

- Base Address: 0x1250_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0002

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	–
TSI clock down ready	[1]	R	If this field is set to 1, TSI block is ready to be down. 0 = Not ready 1 = Ready	1'b1
TSI on	[0]	RW	TSI on/off 0 = TSI off 1 = TSI on	1'b0

33.2.1.2 TS_CON

- Base Address: 0x1250_0000
- Address = Base Address + 0x0004, Reset Value = 0x1000_0000

Name	Bit	Type	Description	Reset Value
TSI SWRESET	[31]	W	Initializes all registers and states of TSI block 0 = No effect 1 = Reset (equals to hardware reset)	–
TS_CLK filter mode	[30]	RW	Specifies the filter mode 0 = Off 1 = On (use ts_clk filter. max. ts_clk frequency is 1/4 HCLK)	1'b0
TSI burst length	[29:28]	RW	Sets the TSI burst length 00 = 1-beat burst 01 = 4-beat burst 10 = 8-beat burst 11 = Reserved	2'b01
output_buffer _full_int_mode	[27]	RW	Sets the output buffer full interrupt mode 0 = Disable 1 = Enable	1'b0
int_fifo_full _int_mode	[26]	RW	Sets the internal FIFO full interrupt mode 0 = Disable 1 = Enable	1'b0
sync_mismatch _int_mode	[25:24]	RW	Sets the sync mismatch interrupt mode 0x = Disable 10 = Enables with skip mode 11 = Enables with stop mode	2'b00
psuf_int_mode	[23:22]	RW	Sets the packet size underflow interrupt mode 0x = Disables 10 = Enables with skip mode 11 = Enables with stop mode	2'b00
psof_int_mode	[21:20]	RW	Sets the packet size overflow interrupt mode 0x = Disables 10 = Enables with skip mode 11= Enables with stop mode	2'b00
ts_clk_timeout _int_mode	[19]	RW	Sets the TS_CLK timeout interrupt mode 0 = Disable 1 = Enable	2'b00
ts_error_int _mode	[18:16]	RW	Sets the TS_ERROR interrupt mode 0xx = Disables 100 = Enables with skip mode (detecting size: bit) 101 = Enables with stop mode (detecting size: bit) 110 = Enables with skip mode (detecting size: packet) 111 = Enables with stop mode (detecting size: packet)	3'b000
padding_pattern	[15:8]	RW	Specifies the Padding pattern	8'h00

Name	Bit	Type	Description	Reset Value
pid_filter_mode	[7]	RW	Specifies the PID filtering mode 0 = Bypass mode 1 = Filtering mode	1'b0
ts_error_active	[6]	RW	Specifies the TS_ERROR active mode 0 = Active high 1 = Active low	1'b0
data_byte_order	[5]	RW	Specifies the TS_DATA byte ordering 0 = MSB to LSB 1 = LSB to MSB	1'b0
ts_valid_active	[4]	RW	Specifies the TS_VALID active mode 0 = Active high 1 = Active low	1'b0
ts_sync_active	[3]	RW	Specifies the TS_SYNC active mode 0 = Active high 1 = Active low	1'b0
ts_clk_invert	[2]	RW	Specifies the TS_CLK inverting mode 0 = Non-inverting (falling-edge data fetch) 1 = Inverting (ringing-edge data fetch)	1'b0
RSVD	[1:0]	–	Reserved	–

33.2.1.3 TS_SYNC

- Base Address: 0x1250_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_00F0

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	–
sync_csdc3	[19:16]	R	Specifies the current sync detecting count3	4'h0
sync_csdc2	[15:12]	R	Specifies the current sync detecting count2	4'h0
sync_csdc1	[11:8]	RW	Specifies the current sync detecting count1	4'h0
sync_det_cnt	[7:4]	–	Specifies the Sync detecting count. When the sync detecting mode uses sync byte, this field indicates the initial detecting count.	4'hF
RSVD	[3:2]	–	Reserved	–
sync_det_mode	[1:0]	–	Specifies the Sync detecting mode 00 = Using TS_SYNC (detecting consecutive 8-bit) 01 = Using TS_SYNC (detecting only 1-bit) 1x = Using sync byte (0x47)	2'b00

33.2.1.4 TS_CNT

- Base Address: 0x1250_0000
- Address = Base Address + 0x000C, Reset Value = 0x00FF_FFFF

Name	Bit	Type	Description	Reset Value
ts_clk_error_cnt	[31:0]	RW	Specifies the TS_CLK timeout period. When the ts_clk does not toggle for n-times of hclk, ts_clk_timeout generates an interrupt. TS_CLK timeout period: HCLK (7.5 ns) × n	32'h00FF_FFFF

33.2.1.5 TS_BASE

- Base Address: 0x1250_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ts_base_addr	[31:2]	RW	Specifies the TS buffer base address (word aligned)	30'h0
RSVD	[1:0]	–	Reserved	–

33.2.1.6 TS_SIZE

- Base Address: 0x1250_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	–
ts_buffer_size	[15:0]	RW	This field should be 47-word (188-byte) × n, where n specifies the stream packet word count (0 to 348 word). If the buffer is full, the buffer write address is cleared to the buffer base address.	16'h0

33.2.1.7 TS_CADDR

- Base Address: 0x1250_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ts_caddr	[31:2]	R	Specifies the TS buffer current write address	30'h0
RSVD	[1:0]	–	Reserved	–

33.2.1.8 TS_INT_MASK

- Base Address: 0x1250_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
dma_complete_ mask	[7]	RW	Specifies the DMA complete interrupt mask 0 = Masking 1 = Not masking	1'b0
output_buffer_full_ mask	[6]	RW	Specifies the TSI interrupt mask: output buffer full 0 = Masking 1 = Not masking	1'b0
int_fifo_full_mask	[5]	RW	Specifies the TSI interrupt mask: internal FIFO full 0 = Masking 1 = Not masking	1'b0
sync_mismatch_ mask	[4]	RW	Specifies the TSI interrupt mask: sync mismatch 0 = Masking 1 = Not masking	1'b0
packet_size_ underflow_mask	[3]	RW	Specifies the TSI interrupt mask: packet size underflow 0 = Masking 1 = Not masking	1'b0
packet_size_ overflow_mask	[2]	RW	Specifies the TSI interrupt mask: packet size overflow 0 = Masking 1 = Not masking	1'b0
TS_CLK_mask	[1]	RW	Specifies the TSI interrupt mask: TS_CLK 0 = Masking 1= Not masking	1'b0
TS_ERROR_mask	[0]	RW	Specifies the TSI interrupt mask: TS_ERROR 0 = Masking 1 = Not masking	1'b0

33.2.1.9 TS_INT

- Base Address: 0x1250_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
dma_complete_flag	[7]	RW	Specifies the DMA transfer complete flag 0 = Interrupt is not generated 1 = Interrupt is generated (Writing "1" clears this field and writing "0" has no effect.)	1'b0
output_buffer_full_flag	[6]	RW	Specifies the Output buffer full interrupt flag 0 = Interrupt is not generated 1 = Interrupt is generated (Writing "1" clears this field and writing "0" has no effect.)	1'b0
int_fifo_full_flag	[5]	RW	Specifies the Internal FIFO full interrupt flag 0 = Interrupt is not generated 1 = Interrupt is generated (Writing "1" clears this field and writing "0" has no effect.)	1'b0
sync_mismatch_flag	[4]	RW	Specifies the Sync mismatch interrupt flag 0 = Interrupt is not generated 1 = Interrupt is generated (Writing "1" clears this field and writing "0" has no effect.)	1'b0
psuf_flag	[3]	RW	Specifies the Packet underflow interrupt flag 0 = Interrupt is not generated 1 = Interrupt is generated (Writing "1" clears this field and writing "0" has no effect.)	1'b0
psof_flag	[2]	RW	Specifies the Packet overflow interrupt flag 0 = Interrupt is not generated 1 = Interrupt is generated (Writing "1" clears this field and writing "0" has no effect.)	1'b0
ts_clk_timeout_flag	[1]	RW	Specifies the TS_CLK timeout interrupt flag 0 = Interrupt is not generated 1 = Interrupt is generated (Writing "1" clears this field and writing "0" has no effect.)	1'b0
ts_error_flag	[0]	RW	Specifies the TS_ERROR interrupt flag 0 = Interrupt is not generated 1 = Interrupt is generated (Writing "1" clears this field and writing "0" has no effect.)	1'b0

33.2.1.10 TS_PIDn (n = 0 to 31)

- Base Address: 0x1250_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000
- Address = Base Address + 0x0038, Reset Value = 0x0000_0000
- Address = Base Address + 0x003C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000
- Address = Base Address + 0x0048, Reset Value = 0x0000_0000
- Address = Base Address + 0x004C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000_0000
- Address = Base Address + 0x0058, Reset Value = 0x0000_0000
- Address = Base Address + 0x005C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000
- Address = Base Address + 0x0064, Reset Value = 0x0000_0000
- Address = Base Address + 0x0068, Reset Value = 0x0000_0000
- Address = Base Address + 0x006C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0070, Reset Value = 0x0000_0000
- Address = Base Address + 0x0074, Reset Value = 0x0000_0000
- Address = Base Address + 0x0078, Reset Value = 0x0000_0000
- Address = Base Address + 0x007C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000
- Address = Base Address + 0x0084, Reset Value = 0x0000_0000
- Address = Base Address + 0x0088, Reset Value = 0x0000_0000
- Address = Base Address + 0x008C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000_0000
- Address = Base Address + 0x0094, Reset Value = 0x0000_0000
- Address = Base Address + 0x0098, Reset Value = 0x0000_0000
- Address = Base Address + 0x009C, Reset Value = 0x0000_0000
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RVSD	[31:14]	–	Reserved	–
pid0_en	[13]	RW	Specifies the PID filter0 0 = Disables 1 = Enables	1'b0
pid0_value	[12:0]	RW	Specifies the PID0 value. If an input stream's PID is different from this value, then the stream is ignored.	13'h0

33.2.1.11 BYTE_SWAP

- Base Address: 0x1250_0000
- Address = Base Address + 0x00BC, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:0]	–	Reserved	–
byte_swap	[0]	–	Specifies the TSI Tx byte swap enable register for little endian 0 = Disables big endian 1 = Enables little endian	1'b1

34 Audio Subsystem

34.1 Overview

Audio Subsystem (AudioSS) is a special subsystem that supports audio playback with a very low power consumption. It consists of Samsung Reconfigurable Processor (SRP), I2S, and 228 (36 + 64 + 128) KB SRAM.

34.2 Features

The key features of audio subsystem are:

- Low power music playback
- Supports various audio codec
- I2S (which includes a AHB master port) to get data to the internal SRAM
- Totally 228 KB SRAM
- A JTAG interface for software debugging.

34.3 Block Diagram of Audio Block

Figure 34-1 illustrates the block diagram of audio block.

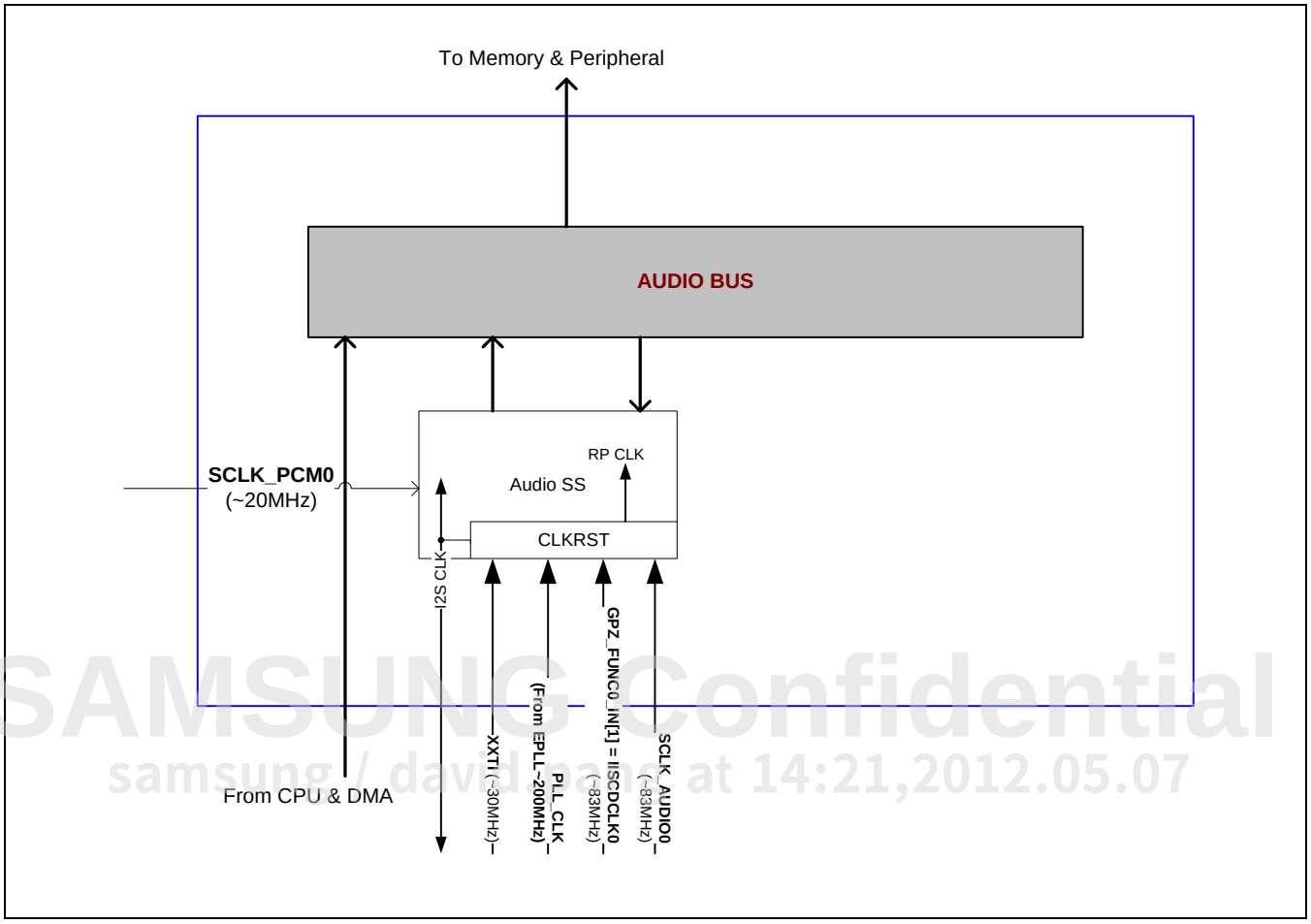


Figure 34-1 Block Diagram of Audio Block

There are two modes for turning power on and off. They are:

- Normal audio playback mode
- Low-power audio playback mode

Audio subsystem and TOP (Exynos 4412 SCP) works in Normal mode. However, to save power, you can power-off TOP. Turning off TOP means that Exynos 4412 SCP goes into low-power audio playback mode. In this mode, audio subsystem can remain on. The audio subsystem uses wake up source such as SRP and I2S to wake up the Exynos 4412 SCP from low-power audio playback mode.

The audio subsystem receives four kinds of clock from CMU. They are:

- XXTI
- EPLL
- I2SCDCLK0
- SCLK_AUDIO0.

The XXTI is a selected clock between main OSC (XXTI) and USB OSC (XusbXTI) which is selected by OM[0] pin Refer to Booting Option and Clock Management user's chapter for more information.

XUSBXTI (or XXTI), EPLL, and I2SCDCLK0 can be supplied to audio subsystem when audio system is on and TOP (Exynos 4412 SCP) is power-off.

Audio subsystem comprises of I2S and its own interrupt, and DMA REQ/ACK ports.

The master port accesses DRAM and IRAM using bus-master modules in audio subsystem.

Alternatively, the slave port accesses all modules in audio subsystem using external audio subsystem modules. All bus-master modules in Exynos 4412 SCP can assess modules in audio subsystem (excluding SRP core).

Audio DMEM, I$, and CMEM are located at Audio SS.

UART for SRP debugging and UART pad are muxed with system UART2 or UART3. Refer to GPIO User’s Manual for more information.

NOTE: Refer to Power Management Unit (PMU) User's Manual for more information about power modes such as low-power audio playback.

Figure 34-2 illustrates the GPIO for audio sub-system.

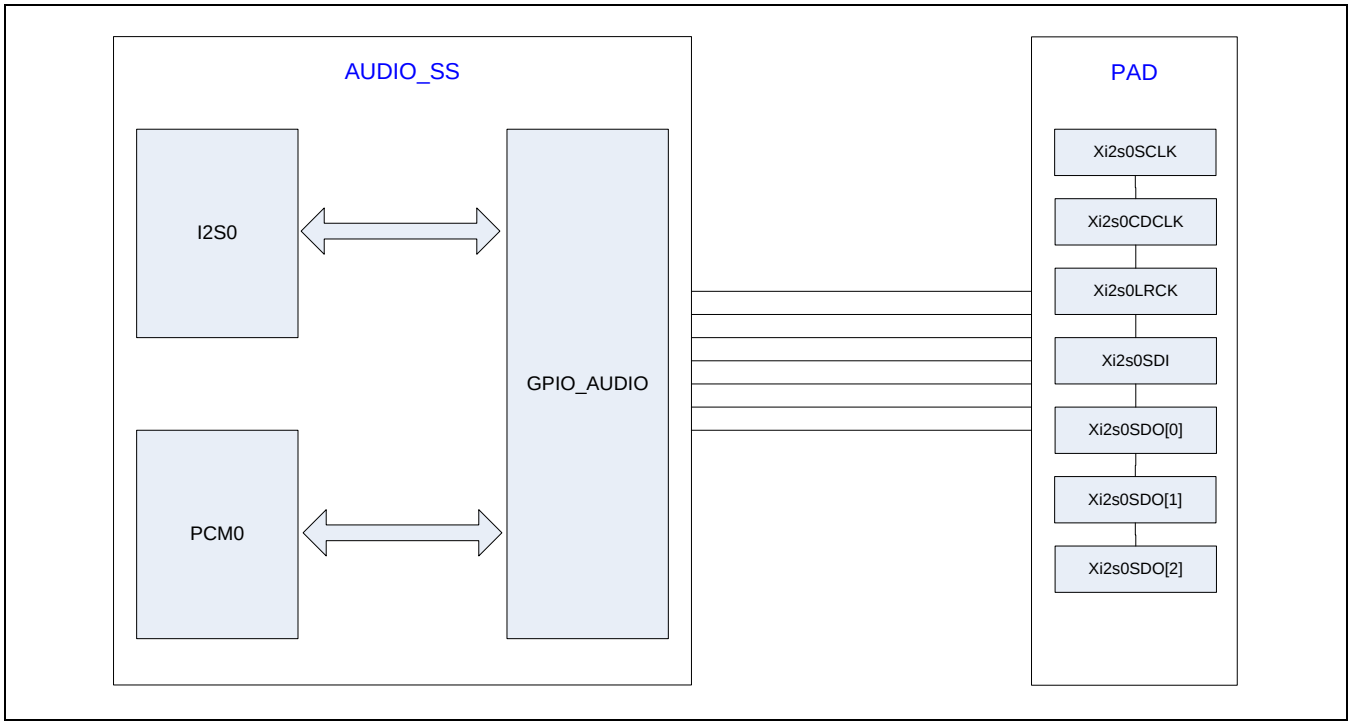


Figure 34-2 GPIO for Audio Sub-System

The blocks that audio SS in MADUIO BLK comprises are:

- SRP: Specifies DSP core, which is an audio-dedicated DSP for Exynos 4412 SCP.
- DMEM, CMEM, and I$: DMEM is used for data memory of SRP, I$ is used for instruction cache of SRP, and CMEM is used for CGA for SRP.
  If SRP is not used, the external modules can use DMEM, and I$ as just SRAM (DMEM: 128 KB and I$: 64 KB).
- ASS_CLKCON: Specifies internal clock controller in audio subsystem.
- COMMBOX: Specifies communication channel between ARM and SRP.
- I2S: Specifies main I2S (I2S0) module of Exynos 4412 SCP.

Figure 34-3 illustrates the JTAG port of SRP debugging.

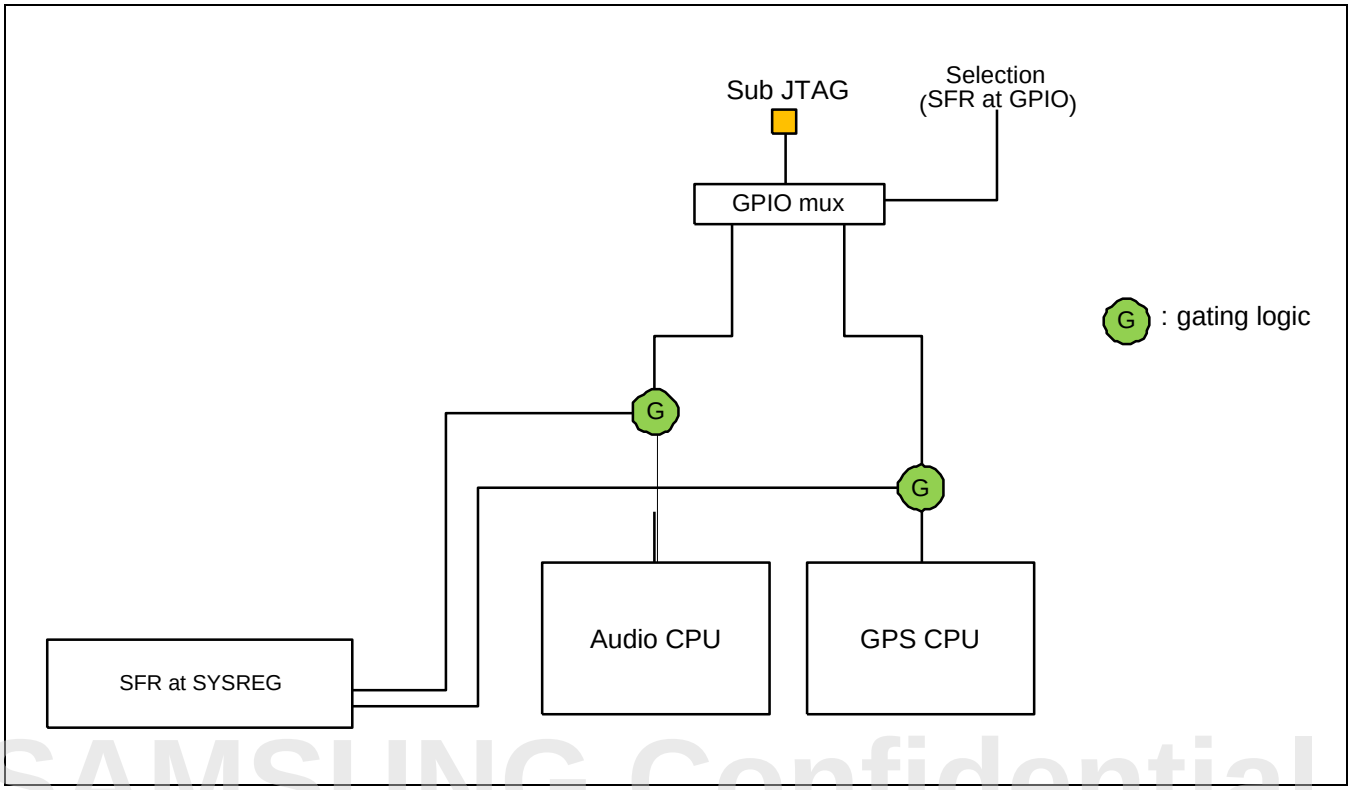


Figure 34-3 JTAG Port of SRP Debugging

There is Sub JTAG port for Audio CPU (= SRP) and GPS CPU to debug with (from External interrupt PAD through GPIO mux to Audio CPU path) in Exynos 4412 SCP.

To pass JTAG signal to Audio CPU, you should set AUDIO_CPU_JTAG_ENB SFR bit as 1 in SYSREG. The Sub JTAG pads are muxed with other function (Main function of the PAD is external interrupt.). To use pads as Sub JTAG for Audio CPU, you should use GPX0CON SFR in GPIO.

Refer to GPIO and SYSREG user's manual to set JTAG port for SRP CPU properly.

34.4 Functional Description

34.4.1 Reconfigurable Processor

Samsung Reconfigurable Processor (SRP) is a Samsung proprietary configurable DSP core. SRP in Exynos 4412 SCP is configured for low power audio applications.

34.4.2 ASS CLK CON

ASS CLK CON specifies clock controller for audio subsystem. It also provides clock for modules in audio subsystem.

Figure 34-4 illustrates the clock controller in audio subsystem.

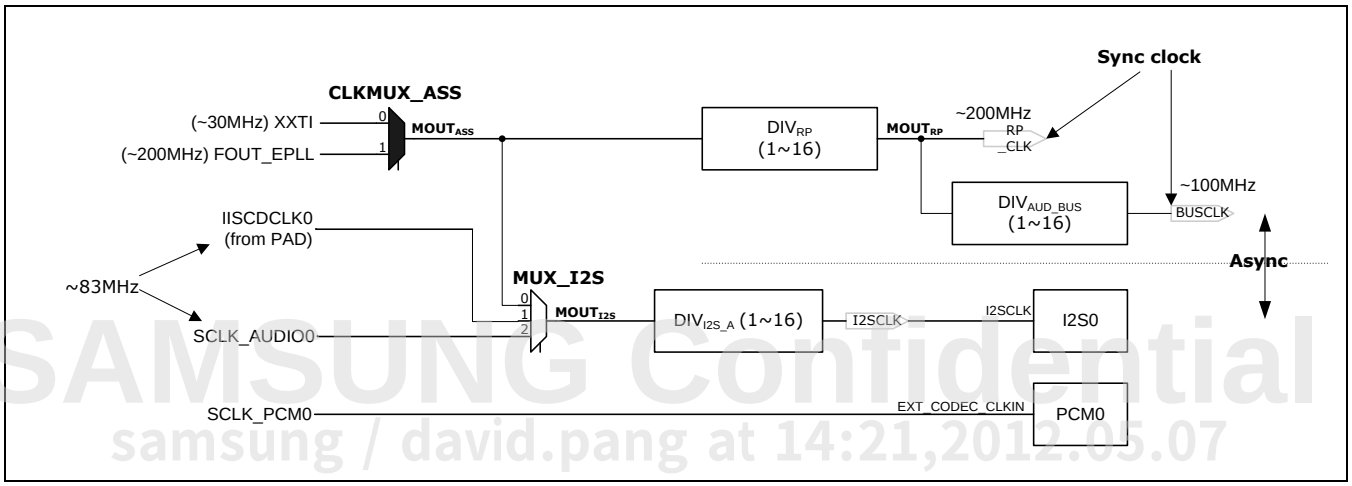


Figure 34-4 Clock Controller in Audio Subsystem

You can change CLKMUX_ASS at any time. However, it should stop running before MUX_I2S changes. You can change value of divider at any time.

34.4.3 Commbox

COMMBOX specifies communication box. It denotes SFR communication channel between ARM and SRP.

34.4.4 I2S

I2S in AudioSS has a small AHB DMA, as well as has interface with external DMA. The I2S allows audio data to play with its own DMA. I2S DMA issues only 32-bit single read transaction.

The I2S has an interrupt request signal to wake up ARM, if Exynos 4412 SCP is in Idle and low-power audio playback modes. Interrupt request signal occurs if the pre-defined configuration of I2S DMA operations is complete. After CPU wakes up, it prepares, generates, and saves next audio data to be played in SRAM at audio subsystem. Then CPU is powered off again to save power.

Refer to I2S (5.1Ch.) User's Manual for more information.

34.4.5 SRAM

The total memory size in AudioSS is 228 KB, out of which 64 KB is reserved for I$, 128 KB is for DMEM, and 36 KB is for CMEM. You can access these memories by both internal and external modules in audio subsystem.

34.5 Input/Output Description

Signal	I/O	Description	Pad	Type
I2S_0_SCLK	I/O	Specifies bit clock.	Xi2s0SCLK	Dedicated
I2S_0_LRCK	I/O	Specifies LR channel clock.	Xi2s0LRCK	Dedicated
I2S_0_CDCLK	I/O	Specifies codec clock.	Xi2s0CDCLK	Dedicated
I2S_0_SDI	I	Specifies I2S serial data input.	Xi2s0SDI	Dedicated
I2S_0_SDO[0]	O	Specifies I2S serial data out 0.	Xi2s0SDO_0	Dedicated
I2S_0_SDO[1]	O	Specifies I2S serial data out 1.	Xi2s0SDO_1	Dedicated
I2S_0_SDO[2]	O	Specifies I2S serial data out 2.	Xi2s0SDO_2	Dedicated
UART_AUDIO_RXD	I	Specifies UART data input.	XuRXD_2 or XuRXD_3	Muxed
UART_AUDIO_TXD	O	Specifies UART data output.	XuTXD_2 or XuTXD_3	Muxed
AUD_TCK	I	Specifies JTAG clock input.	XEINT_0	Muxed
AUD_TMS	I	Specifies JTAG TMS signal input.	XEINT_1	Muxed
AUD_TDI	I	Specifies JTAG data input.	XEINT_2	Muxed
AUD_TDO	O	Specifies JTAG data out.	XEINT_3	Muxed
AUD_TRSTn	I	Specifies JTAG reset input.	XEINT_4	Muxed

34.6 Register Description

34.6.1 Register Map Summary

- Base Address: 0x0300_0000

Register	Offset	Description	Reset Value
Audio Subsystem Internal Memory			
DMEM	0x0000 to 0xFFFF	For SRP, data memory For external audio subsystem, 128 KB SRAM	Undefined
I$	0x0000 to 0xFFFF	For SRP, instruction cache For external audio subsystem, 64 KB SRAM	Undefined
CMEM	0x0000 to 0x8FFF	For SRP, configuration memory For external audio subsystem, 36 KB SRAM You should use this memory only for SRP decoding mode, not for ARM decoding mode.	Undefined

- Base Address: 0x0381_0000

Register	Offset	Description	Reset Value
Audio Subsystem CLKCON			
ASS CLK SRC	0x0000	Specifies the clock source select register.	0x0
ASS CLK DIV	0x0004	Specifies the clock divider register.	0x0
ASS CLK GATE	0x0008	Specifies the clock gate register.	0x1FF

- Base Address: 0x0382_0000

Register	Offset	Description	Reset Value
Communication Box (Commbox)			
SRP_RST_CONT	0x0000	Specifies the reset control of SRP.	0x1
SRP_CONF	0x0004	Specifies the SRP configurations of operation.	0x40
ASS_INTR	0x0008	Specifies the interrupt from audio subsystem to ARM. Also, it can be used as wake up source.	0x0
SRP_PENDING	0x000C	Specifies the pending control of SRP.	0x1
SW_DEFINE00	0x0010	Software can freely use this register.	0x0
SW_DEFINE01	0x0014	Software can freely use this register.	0x0
SW_DEFINE02	0x0018	Software can freely use this register.	0x0
SW_DEFINE03	0x001C	Software can freely use this register.	0x0
SW_DEFINE04	0x0020	Software can freely use this register.	0x0
SW_DEFINE05	0x0024	Software can freely use this register.	0x0
SW_DEFINE06	0x0028	Software can freely use this register.	0x0
SW_DEFINE07	0x002C	Software can freely use this register.	0x0

Register	Offset	Description	Reset Value
SW_DEFINE08	0x0030	Software can freely use this register.	0x0
SW_DEFINE09	0x0034	Software can freely use this register.	0x0
SW_DEFINE10	0x0038	Software can freely use this register.	0x0
SW_DEFINE11	0x0100	Software can freely use this register.	0x0
SW_DEFINE12	0x0104	Software can freely use this register.	0x0
SW_DEFINE13	0x0108	Software can freely use this register.	0x0
SW_DEFINE14	0x010C	Software can freely use this register.	0x0
SW_DEFINE15	0x0110	Software can freely use this register.	0x0
SW_DEFINE16	0x0114	Software can freely use this register.	0x0
SW_DEFINE17	0x0118	Software can freely use this register.	0x0
SW_DEFINE18	0x011C	Software can freely use this register.	0x0
SW_DEFINE19	0x0120	Software can freely use this register.	0x0
SW_DEFINE20	0x0124	Software can freely use this register.	0x0
SW_DEFINE21	0x0128	Software can freely use this register.	0x0
SW_DEFINE22	0x012C	Software can freely use this register.	0x0
INTREN	0x0180	Specifies SRP_IRQ Interrupt enable register	0x0
INTRMASK	0x0184	SRP_IRQ Interrupt Masking Register	0x7F
INTRSRC	0x0188	Specifies SRP_IRQ Interrupt Source Register	0x0

34.6.2 Audio Subsystem CLKCON

Refer to *Figure 34-4* to set the registers of audio subsystem CLK CON.

34.6.2.1 ASS CLK SRC

- Base Address:0x0381_0000
- Address = Base Address + 0x0000, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0
MUX_I2S	[3:2]	RW	Selects I2S clock mux. 00 = Main CLK 01 = IISCDCLK0 (from PAD) 10 = SCLK_AUDIO0	0
RSVD	[1]	–	Reserved (This bit must be set as 0.)	0
CLKMUX_ASS	[0]	RW	Selects ASS clock mux. 0 = XXTI 1 = FOUT_EPLL	0

34.6.2.2 ASS CLK DIV

- Base Address: 0x0381_0000
- Address = Base Address + 0x0004, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0
I2S_CLK_RATIO	[11:8]	RW	Specifies the I2S clock divider ratio. I2SCLK = $MOUT_{I2S}$/(I2S_CLK_RATIO + 1)	0
BUS_CLK_RATIO	[7:4]	RW	Specifies the BUS clock divider ratio. BUS_CLK = $MOUT_{SRP}$/(BUS_CLK_RATIO + 1)	0
SRP_CLK_RATIO	[3:0]	RW	Specifies the BUS clock divider ratio. SRP_CLK = $MOUT_{ASS}$/(SRP_CLK_RATIO + 1)	0

34.6.2.3 ASS CLK GATE

- Base Address: 0x0381_0000
- Address = Base Address + 0x0008, Reset Value = 0x1ff

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	0
Timer	[8]	RW	Specifies the gating CLK to Timer 0 = Mask 1 = Pass	1
UART	[7]	RW	Specifies the gating CLK to UART 0 = Mask 1 = Pass	1
GPIO	[6]	RW	Specifies the gating clock of GPIO 0 = Mask 1 = Pass	1
PCM special	[5]	RW	Specifies the gating CLK to PCM function 0 = Mask 1 = Pass	1
PCM Bus	[4]	RW	Specifies the gating CLK to PCM bus 0 = Mask 1 = Pass	1
I2S special	[3]	RW	Specifies the gating CLK to I2S function 0 = Mask 1 = Pass	1
I2S Bus	[2]	RW	Specifies the gating CLK to I2S bus 0 = Mask 1 = Pass	1
IntMEM	[1]	RW	Specifies the gating CLK to IntMEM in Audio Subsystem 0 = Mask 1 = Pass	1
SRP	[0]	RW	Specifies the gating CLK to SRP (including I$ and DMEM) 0 = Mask 1 = Pass	1

34.6.3 Communication Box (Commbox)

34.6.3.1 SRP_RST_CONT

- Base Address: 0x0382_0000
- Address = Base Address + 0x0000, Reset Value = 0x1

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
SRP_RST_CONT	[0]	RW	0 = Resets SRP, this bit is automatically cleared to "1" after one cycle of bus clock	1

34.6.3.2 SRP_CONF

- Base Address: 0x0382_0000
- Address = Base Address + 0x0004, Reset Value = 0x40

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0
I2S_INTR_WKUP_ENB	[6]	RW	Specifies if I2S interrupt is used as a wakeup source 0 = When I2S interrupt isn't used as a wakeup source 1 = When I2S interrupt is used as a wakeup source	1
SRP_HALTED	[5]	R	Specifies SRP is halted.	0
AUDIO_MODE	[4]	RW	Specifies if SRP accepts level interrupt from I2S 0 = When SRP does not decode audio bitstream 1 = When SRP decodes audio bitstreaml	0
SRP_DECODING_MODE	[3]	RW	Specifies if SRP is in operation 0 = External masters can use SRP I$ memory 1 = External masters cannot access SRP I$ memory	0
SRP_BOOT	[2]	RW	Controls SRP booting type 0 = SRP boots from boot_rom and fetches instructions from external memory 1 = SRP boots from internal cache memory	0
PCM_BIT	[1:0]	RW	Decoding Output PCM Control Register	0

34.6.3.3 ASS_INTR

- Base Address: 0x0382_0000
- Address = Base Address + 0x0008, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
SRP_RST_CONT	[0]	RW	Specifies the interrupt from audio subsystem to ARM. Also, you can use it as wake up source.	0

34.6.3.4 SRP_PENDING

- Base Address: 0x0382_0000
- Address = Base Address + 0x000C, Reset Value = 0x1

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
SRP_PENDING	[0]	RW	Specifies the pending control of SRP. 0 = SRP is running 1 = SRP is pending	1

34.6.3.5 INTREN

- Base Address: 0x0382_0000
- Address = Base Address + 0x0180, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
INTREN	[0]	RW	SRP_IRQ interrupt enable register 0 = Disables Interrupt 1 = Enables Interrupt	0

34.6.3.6 INTRMASK

- Base Address: 0x0382_0000
- Address = Base Address + 0x0184, Reset Value = 0x7F

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0
ARM_INTR_MSK	[6]	RW	ARM Interrupt Mask Register 0 = Interrupt does not mask 1 = Masks Interrupt	0x1
SRPDMA_INTR_MSK	[5]	R	SRPDMA Interrupt Mask Register 0 = Interrupt does not mask 1 = Masks Interrupt	0x1
TMR_INTR_MSK	[4:0]	R	Timer Interrupt Mask Register 0 = Interrupt does not mask 1 = Masks Interrupt	0x1F

34.6.3.7 INTRSRC

- Base Address: 0x0382_0000
- Address = Base Address + 0x0188, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	–	Reserved	0
ARM_INTR_SRC	[6]	RW	ARM Interrupt Source Register	0x0
SRPDMA_INTR_SRC	[5]	R	SRPDMA Interrupt Source Register	0x0
TMR_INTR_SRC	[4:0]	R	Timer Interrupt Source Register	0x0

35 IIS Multi Audio Interface

35.1 Overview

Inter-IC Sound (IIS) is one of the popular digital audio interfaces.

The IIS bus handles only audio data and the other signals such as sub-coding and control are transferred separately. Serial data, word select and clock signals are used to minimize the number of pins required and to keep wiring simple.

- Serial data signals (I2SSDO0, I2SSDO1, I2SSDO2, I2SSDI)
- Word select signal (I2SLRCLK)
- Clock signals(I2SSCLK, I2SSDCLKO, I2SSDCLKI)

IIS interface transmits or receives sound data from external stereo audio codec. There're two 32x64 FIFOs (First-In-First-Out) data structures for transmitting and receiving data. DMA transfer mode can be supported to transmit and receive samples.

IIS Version 5.1 can handle up to two sound sources. For example, OS (Operating system)-controlled sound can be delivered to primary sound path and OS-independent sound can be delivered to secondary sound path. IIS Version 5.1 can mix primary sound source and secondary sound source.

35.2 Features

The features of IIS Multi Audio Interface are:

- Mixes up to two sound sources: Primary and Secondary sound source.
- Primary sound source can drive up to 5.1Ch. IIS-bus for audio interface with external DMA-based operation
- Secondary sound source can support stereo sound channels with internal DMA
- Serial, 8/16/24-bit per channel data transfers
- Supports IIS, MSB-justified and LSB-justified data format
- IIS Version 5.1 interrupt can wake-up system from any power mode except sleep
- Supports master/slave mode
- Supports auxiliary clock out for codec chip

35.3 Block Diagram

Figure 35-1 illustrates the IIS- Bus block diagram.

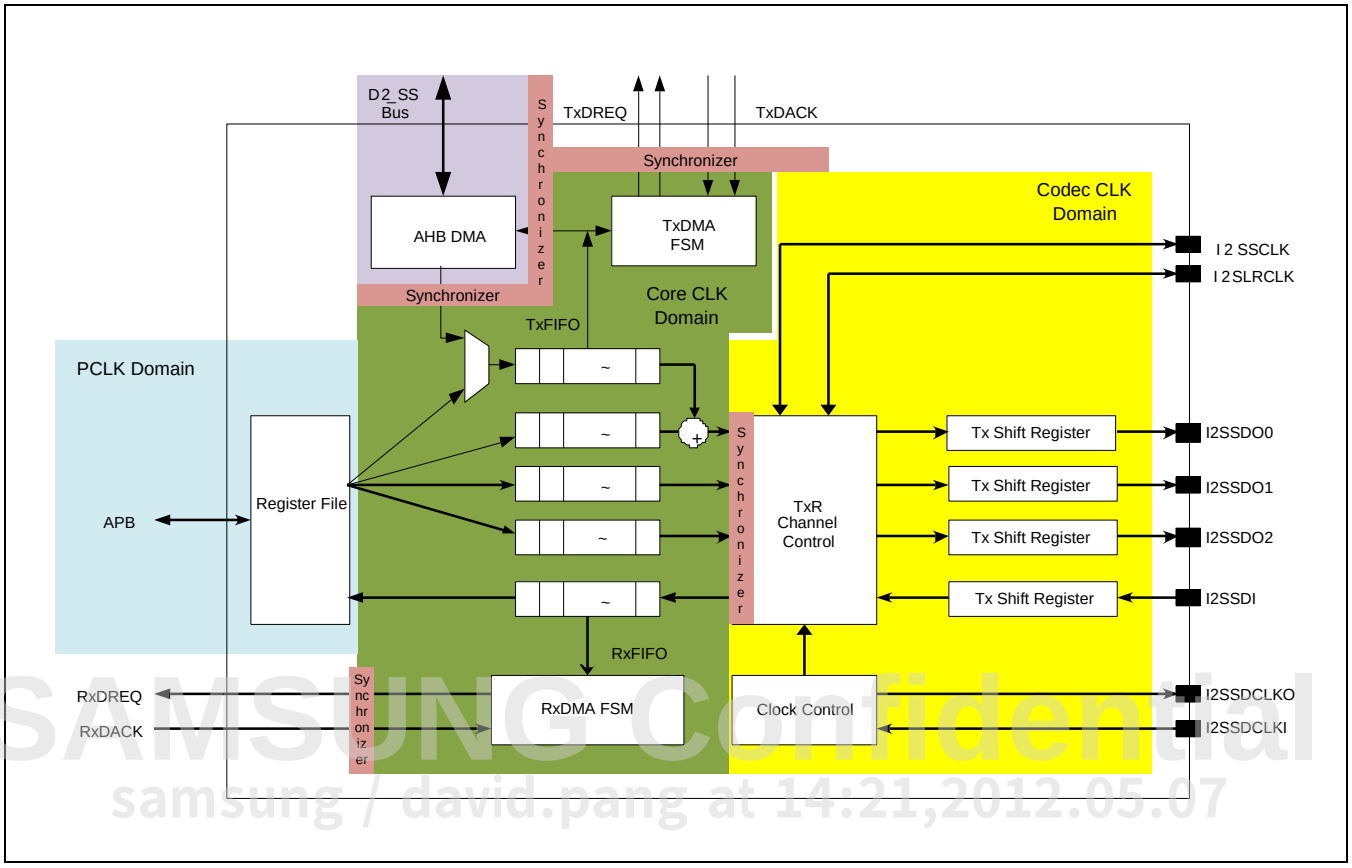


Figure 35-1 IIS-Bus Block Diagram

NOTE: I2SSDCLKO are I2SSDCLKI are I2S codec clock. They are muxed at GPIO block, and there is only one external pin (I2S_0_CDCLK).

35.4 Functional Description

IIS interface consists of register bank, FIFOs, shift registers, clock control, DMA finite state machine, and channel control block as illustrated in *Figure 35-1*. Note that each FIFO has 32-bit width and 64 depth structure, which contains left/right channel data. Therefore, FIFO access and data transfer are handled with left/right pair unit.

35.4.1 Master/Slave Mode

Master/Slave mode shows direction of I2SLRCLK and I2SSCLK. If IIS bus interface transmits I2SLRCLK and I2SSCLK to IIS codec, then IIS bus is master mode. If IIS bus interface receives I2SLRCLK and I2SSCLK from IIS codec, then IIS bus is slave mode. To select master or slave mode, set MSS bit of IISMOD register.

Tx/Rx mode indicates the direction of data flow. If IIS bus interface transmits data to IIS codec, this indicates Tx mode. Alternatively, IIS bus interface receives data from IIS codec, this indicates Rx mode.

Figure 35-2 illustrates the AUDIO BUS CLK. Refer to Chapter 37 Audio Subsystem for more information.

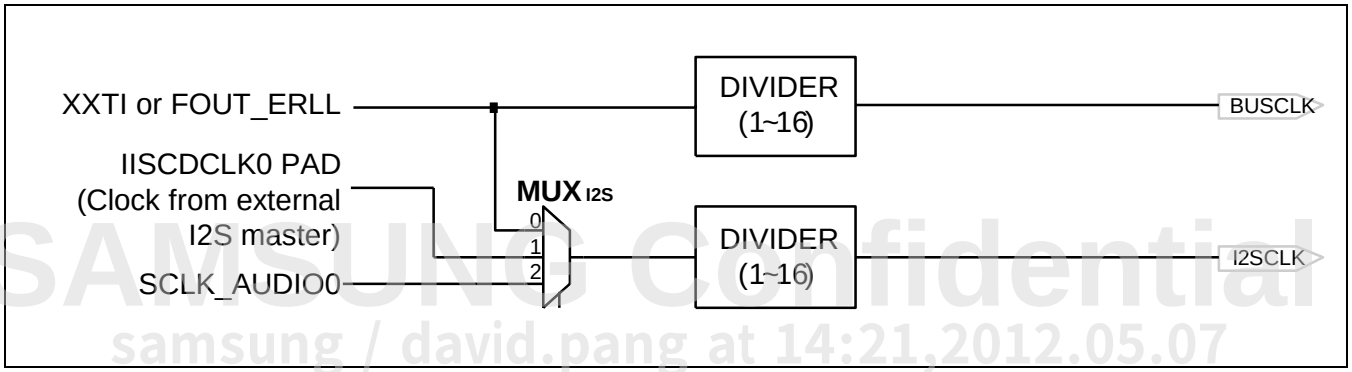


Figure 35-2 Clock Controller in Audio Sub-System

Figure 35-3 illustrates the IIS clock control block diagram.

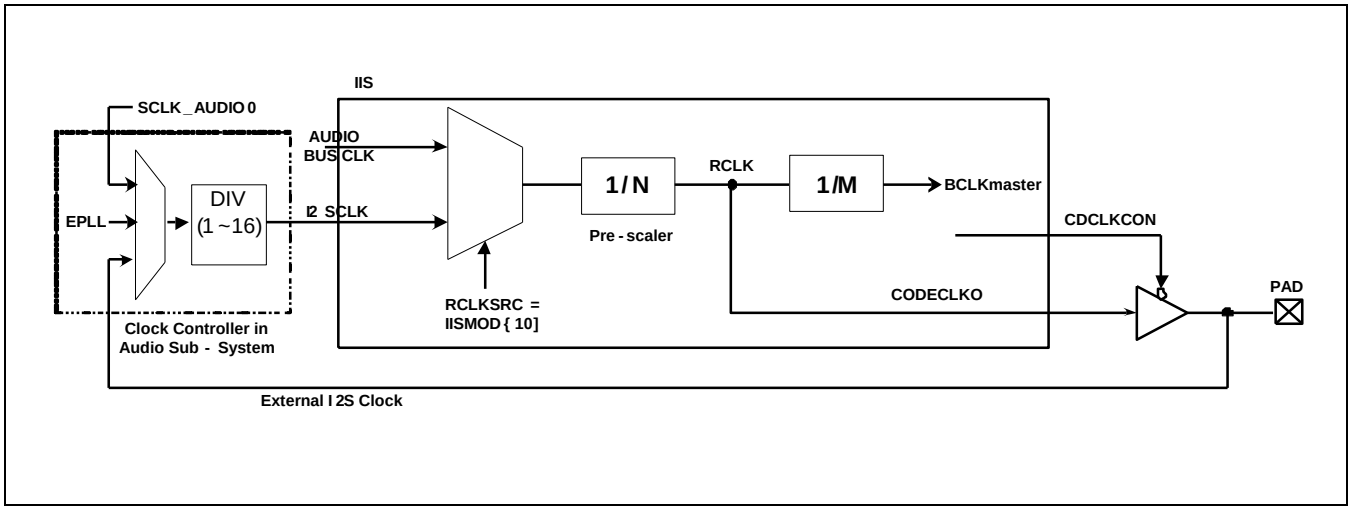


Figure 35-3 IIS Clock Control Block Diagram

Figure 35-3 illustrates the route of root clock with setting in IIS clock control block and system controller. RCLK indicates root clock and RCLKSRC selects a clock source of RCLK between AUDIO BUS CLK (BUSCLK at *Figure 35-2*) and I2SCLK. The IIS pre-scaler (clock divider) is employed to generate a root clock with divided frequency from source clock.

In master mode, the root clock is divided to generate I2SSCLK and I2SLRCLK. In slave mode, this clock is not used to generate I2SSCLK and I2SLRCLK.

CDCLKCON controls direction of CDCLK GPIO pad. The direction is set by CDCLKCON SFR bit (IISMOD [12]). When CDCLKCON SRF bit is 0, auxiliary clock out is supported for Codec chip at both cases of Master/Slave mode. In this case, RCLK can be supplied to external IIS CODEC chip. When CDCLKCON SRF bit is 1, External I2S clock is supplied from external device. This is useful when internal clock sources are not adequate for generating exact I2SSCLK and I2SLRCLK.

Figure 35-4 illustrates the master/slave modes of IIS.

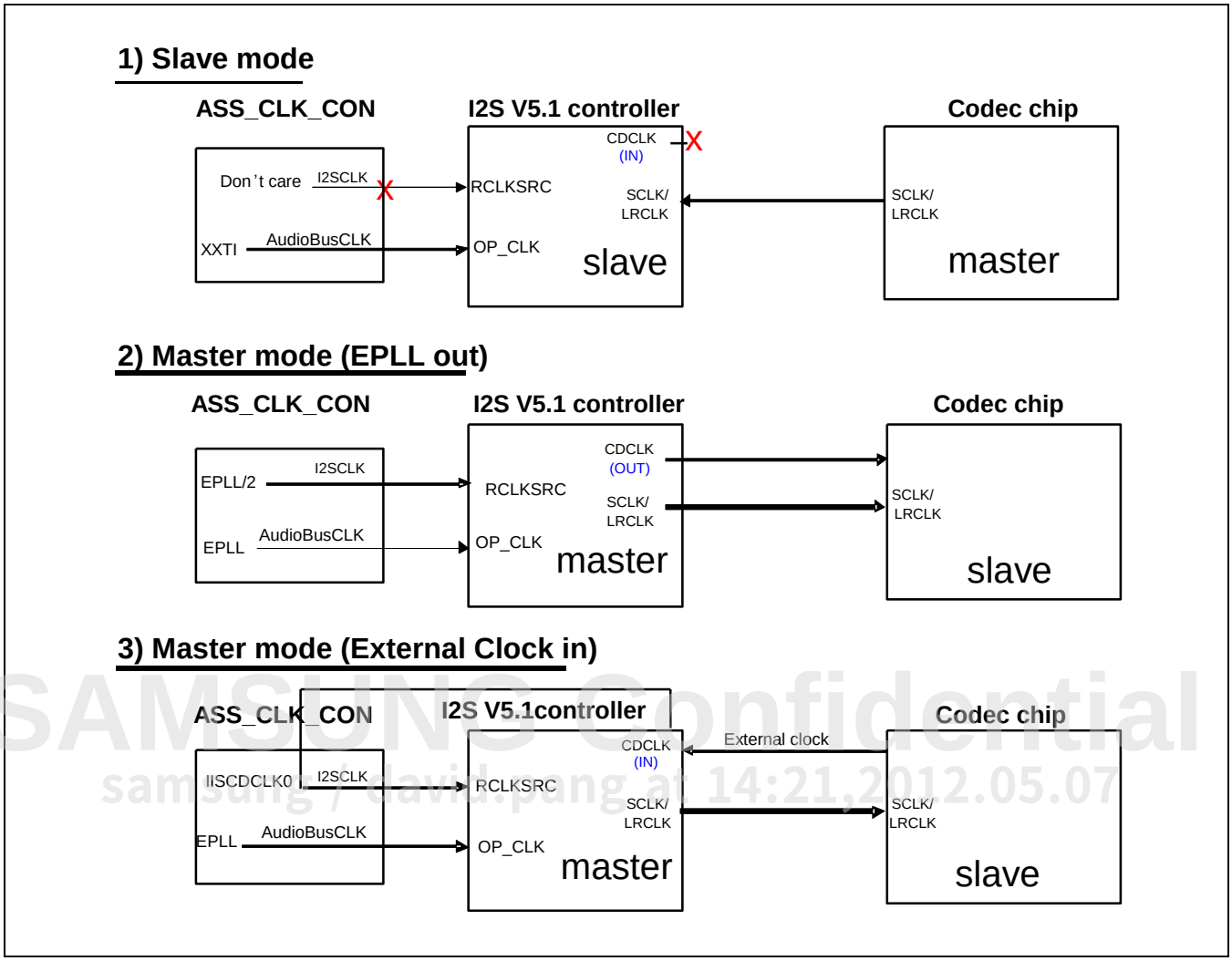


Figure 35-4 Master/Slave Modes of IIS

Table 35-1 lists the typical usage of master/slave modes.

Table 35-1 Typical Usage of Master/Slave Modes

Mode	AudioSS CLK_CON		IIS Version 5.1 IISMOD			
	AudioBusClk	I2SCLK	MSS	RCLKSRC	OP_CLK	CDCLKCON
Slave mode	XXTI or EPLL	Gating	1 (Slave)	1 (I2SCLK)	3 (AudioBusClk)	1 (In)
Master mode (EPLL out)	EPLL	EPLL/2	0 (Master)	1 (I2SCLK)	3 (AudioBusClk)	0 (Out)
Master mode (External clock in)	EPLL	IISCDCLK0	0 (Master)	1 (I2SCLK)	3 (AudioBusClk)	1 (In)

35.4.1.1 External DMA Transfer

You should use external DMA or SFR interface to transfer up to 5.1 channel primary sounds from software mixer to IIS. To play primary sound for 5.1 channels or record two channel sound, IIS has registers. They are:

- TXFIFO0
- TXFIFO1
- TXFIFO2
- TXFIFO_S
- RXFIFO

IIS will mix primary sound in TXFIFO0 and secondary sound in TXFIFO_S and output mixed sound stream to external codec logic.

In external DMA transfer mode, use external DMA controller to access transmitter or receiver FIFO. The transmitter or receiver FIFO state activates DMA service request internally. The FTXEMPT, FRXEMPT, FTXFULL, and FRXFULL bits of I2SCON register represents transmitter or receiver FIFO data state. Especially, FTXEMPT and FRXFULL bits are ready flag for DMA service request. The DMA service request for transmitting is activated when TXFIFO is not empty and the DMA service request for receiving is activated when RXFIFO is not full.

The external DMA transfer uses only handshaking method for single data. Note that during external DMA acknowledge activation; the data read or write operation should be performed during DMA acknowledge activation.

NOTE: Reference: DMA request point

- Tx mode: (FIFO is not full) and (TXDMACTIVE is active)
- Rx mode: (FIFO is not empty) and (RXDMACTIVE is active)

35.4.1.2 Internal DMA Transfer

You should use internal DMA or SFR interface to transfer up to two channel secondary sounds to IIS. To play secondary sound for two channels, internal DMA in IIS gets sound data from somewhere in DMEM (0x0300_0000 to 0x0301_FFFF) or somewhere in IRAM (0x0202_0000 to 0x0205_FFFF) to TXFIFO_S. IIS will mix primary sound in TXFIFO0 and secondary sound in TXFIFO_S and output mixed sound stream to external codec logic.

Like external DMA transfer mode, in internal DMA transfer mode, the internal DMA is activated when TXFIFO_S is not full. After activation, internal DMA runs according to SFR configurations and signals an interrupt after completion.

- It only supports single transfer in both Internal and External DMA transfer mode.
- Refer to Chapter 37 Audio Sub system for more information.

35.4.1.3 Sound Mixing

IIS can mix primary sound in TXFIFO0 and secondary sound in TXFIFO_S when two sound sources have similar sampling rate and PCM format.

- If overflow occurs, then mixer saturates output value.
- Mixer can handle Different Bit Length. (Controlled by BLC bit at IISMOD SFR)

Figure 35-5 illustrates the concept of mixer in IIS.

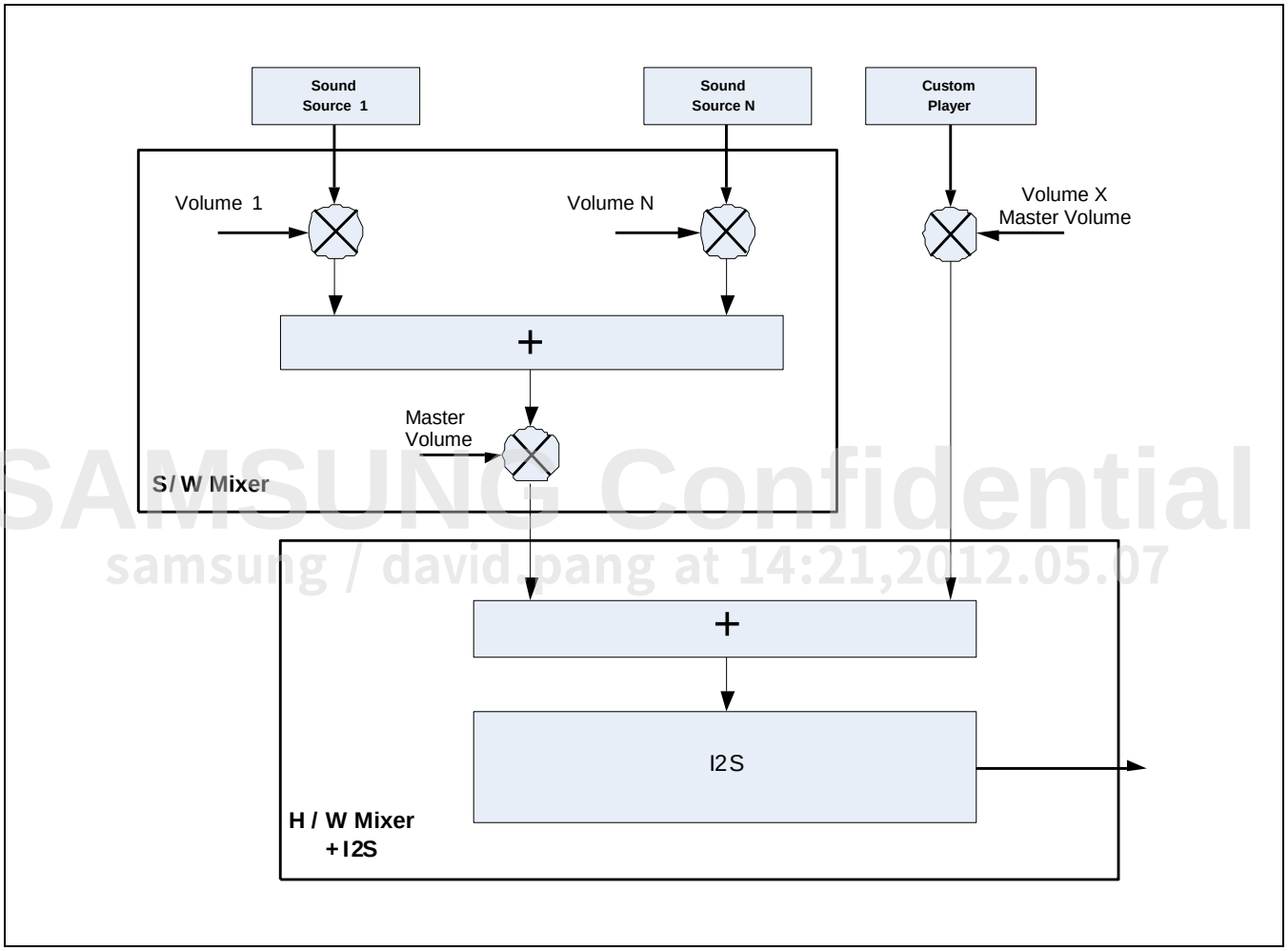


Figure 35-5 Concept of Mixer in IIS

This function has two limitations:

1. Normalization should be pre-processed in software configurations or settings.
2. Synchronization between two sound sources is not guaranteed.

35.5 Audio Serial Data Format

This section includes:

- IIS-Bus Format
- MSB (Left) Justified
- LSB (Right) Justified

35.5.1 IIS-Bus Format

The IIS bus has four lines. It also includes serial data input I2SSDI, serial data output I2SSDO, left/right channel select clock I2SLRCLK, and serial bit clock I2SSCLK; master generates I2SLRCLK and I2SSCLK.

Serial data is transmitted in 2's complement with MSB first with a fixed position, whereas the position of LSB depends on word length. The transmitter sends MSB of the next word at one clock period after the I2SLRCLK is changed. Serial data sent by the transmitter can be synchronized either with trailing or with leading edge of clock signal.

The LR channel select line indicates the direction of left or right channel being transmitted. I2SLRCLK may be changed either on a trailing or leading edge of serial clock, but it does not need to be symmetrical. In the slave, this signal is latched on leading edge of clock signal. The I2SLRCLK line changes one clock period before the MSB is transmitted. This allows the slave transmitter to derive synchronous timing of serial data that will be set up for transmission. Furthermore, it enables receiver to store previous word and clear input for the next word.

35.5.2 MSB (Left) Justified

MSB-Justified (Left-Justified) format is similar to IIS bus format. In MSB-justified format, the transmitter always sends the MSB of the next word simultaneously whenever the I2SLRCLK is changed.

35.5.3 LSB (Right) Justified

LSB-Justified (Right-Justified) format is opposite to MSB-justified format. In other word, the transferring serial data is aligned with ending point of I2SLRCLK transition.

Note that in this figure, the word length is 16-bit and I2SLRCLK makes transition every 24 cycle of I2SSCLK (BFS is 48fs, where fs is sampling frequency; I2SLRCLK frequency).

Figure 35-6 illustrates the audio serial format of IIS, MSB-justified, and LSB-justified.

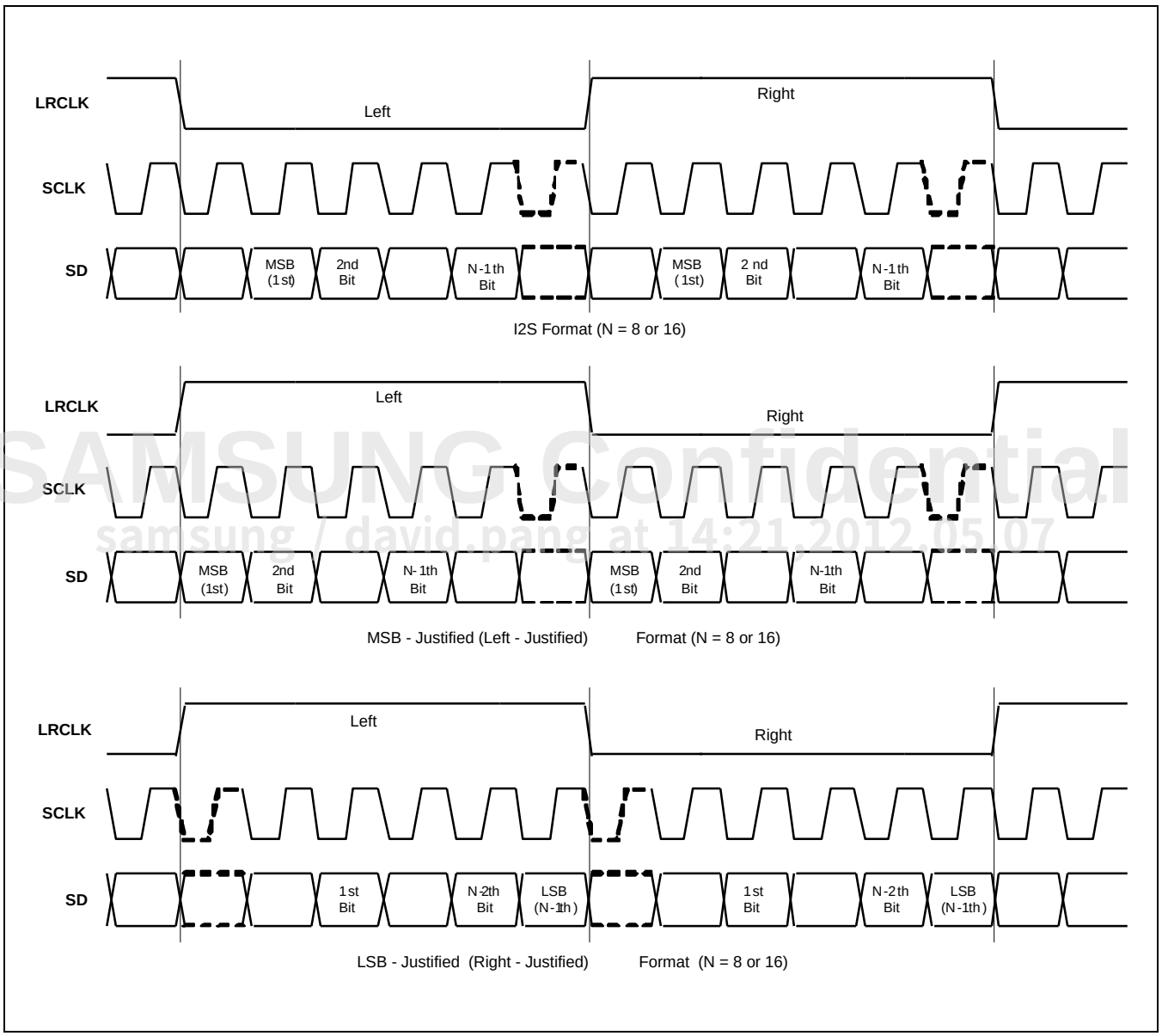


Figure 35-6 IIS Audio Serial Data Formats

35.6 PCM Bit Length (BLC), RFS Divider and BFS Divider for Sampling Frequency (IISLRCLK), Serial bit CLK (IISSCLK), and Root Clock (RCLK)

When IIS interface Controller operates as master, IIS interface Controller generates IISLRCLK and IISSCLK. That is Root Clock is divided using RFS and BFS value. To decide Sampling Frequency – IISLRCLK -, BLC, BFS, and RFS are selected first. Optionally, IIS interface Controller clocks out Root clock as IISCDCLK for codec master clock (if source of root clock is not IISEXTCDCLK).

In slave mode, you should set the value of BLC, BFS, and RFS similar to master (ex: Codec). Because IIS interface controller needs these value for correct operation.

35.6.1 PCM Word Length and BFS Divider

PCM Word Length (BLC) value is selected first, because the value affects BFS value.

Table 35-2 lists the BFS available value as BLC.

Table 35-2 Allowed BFS Value as BLC

PCM Bit length (BLC)	8-bit	16-bit	24-bit
Available BFS value	16 fs, 24 fs, 32 fs, 48 fs	32 fs, 48 fs	48 fs

35.6.2 BFS Divider and RFS Divider

RFS value is selected when BFS is selected

Table 35-3 lists the RFS available value as BFS.

Table 35-3 Allowed RFS Value as BFS

BFS Divider	16 fs, 32 fs	24 fs, 48 fs
Available RFS value	256 fs, 384 fs, 512 fs, 768 fs	384 fs, 768 fs

35.6.3 RFS Divider and Root Clock

RCLK is clock divided by IIS pre-scaler (IISPSR) that is selected by IMS

Table 35-4 lists the relationship between Root Clock, IISLRCLK and RFS

Table 35-4 Root Clock Table (MHz)

IISLRCK/ RFS	8.000 kHz	11.025 kHz	16.000 kHz	22.050 kHz	32.000 kHz	44.100 kHz	48.000 kHz	64.000 kHz	88.200 kHz	96.000 kHz
256 fs	2.0480	2.8224	4.0960	5.6448	8.1920	11.2896	12.2880	16.3840	22.5792	24.5760
384 fs	3.0720	4.2336	6.1440	8.4672	12.2880	16.9344	18.4320	24.5760	33.8688	36.8640
512 fs	4.0960	5.6448	8.1920	11.2896	16.3840	22.5792	24.5760	32.7680	45.1584	49.1520
768 fs	6.1440	8.4672	12.2880	16.9344	24.5760	33.8688	36.8640	49.1520	67.7376	73.7280

Root Clock Frequency = fs × (256, 384, 512 or 768)

35.7 Programming Guide

The IIS bus interface can be accessed either by the processor using programmed I/O instructions or by DMA controller.

35.7.1 Initialization

The procedure for initialization is:

1. Before you use IIS bus interface, you must configure GPIOs (refer to GPZCON at GPIO user's manual for more information.) to IIS mode. Verify direction of signal. I2SLRCLK, I2SSCLK and I2SCDCLK is inout-type. The I2SSDI and I2SSDO is input and output respectively.
2. Select clock source. Exynos 4412 SCP has three clock sources. They are:
   - Audio bus clock
   - EPLL
   - External codec
     Refer to *Figure 35-2* and *Figure 35-3* for more information.

35.7.2 Play Mode (Tx Mode) with DMA

The procedure for Play Mode with DMA is:

1. TXFIFO is flushed before operation. If you do not distinguish Master/Slave mode from Tx/Rx mode, you should study Master/Slave mode and Tx/Rx mode. Refer to Master/Slave chapter for more information.
2. Configure I2SMOD register and I2SPSR (IIS pre-scaler register).
3. To operate system in stability, internal TXFIFO should be almost full before transmission.
   To satisfy this, start TXDMA before asserting I2SACTIVE.
4. IIS bus does not support interrupt. Therefore, you can only verify state by polling through accessing SFR.
5. If TXFIFO is full, you can assert I2SACTIVE.

35.7.3 Recording Mode (Rx Mode) with DMA

The procedure for Recording Mode with DMA is:

1. RXFIFO is flushed before operation. Also, if you do not distinguish between Master/Slave mode and Tx/Rx mode, you should study Master/Slave mode and Tx/Rx mode. Refer to Master/Slave chapter for more information.
2. Configure I2SMOD register and I2SPSR (IIS pre-scaler register).
3. To operate system in stability, the internal RXFIFO should have at least one data before DMA operation. To satisfy this, assert I2SACTIVE before starting RXDMA.
4. Verify RXFIFO state by polling through accessing SFR.
5. If RXFIFO is not empty, start RXDMACTIVE.

35.7.4 Example Code

This section includes:

- Tx Channel
- Rx Channel

35.7.4.1 Tx Channel

The IIS Tx channel provides a single stereo compliant output. The transmit channel can operate in master or slave mode. Data is transferred between processor and IIS controller through an APB access or a DMA access.
The processor must write words in multiples of two (that is, for left and right audio sample). The words are serially shifted out timed with respect to the audio bit clk, SCLK, word select clock, and LRCLK.

Tx Channel has 64 × 32 bits wide FIFO where processor or DMA can write up to 16 left/right data samples after enabling channel for transmission.

An example sequence is as follows:

Ensure Audio bus clock and CDCLK are coming correctly to IIS controller and Flush TX FIFO using TFLUSH bit in I2SFIC register (IIS FIFO Control Register).

Ensure that IIS Controller is configured in one of the modes:

- Tx only mode
- Tx/Rx simultaneous mode

This can be done by programming TXR bit in I2SMOD register (IIS Mode Register).

1. Then program these parameters according to the requirement:
   - MSS, RCLKSRC
   - SDF
   - BFS
   - BLC
   - LRP: For programming the above-mentioned fields refer to I2SMOD Register (IIS Mode Register).

2. Once ensured that input clocks for IIS controller are up and running and step 1 and 2 have been completed we can write to Tx FIFO.

Writing to Tx FIFO has to be carried out through I2STXD Register (IIS Tx FIFO Register). I2STXD Register (IIS Tx FIFO Register).

This 32-bit data will occupy position 0 of FIFO and any further data will be written to position 2, 3, and so on.

The Data is aligned in Tx FIFO for 8-bit/Channel or 16-bit/Channel BLC.

Figure 35-7 illustrates the Tx FIFO structure for BLC = 00 or BLC = 01.

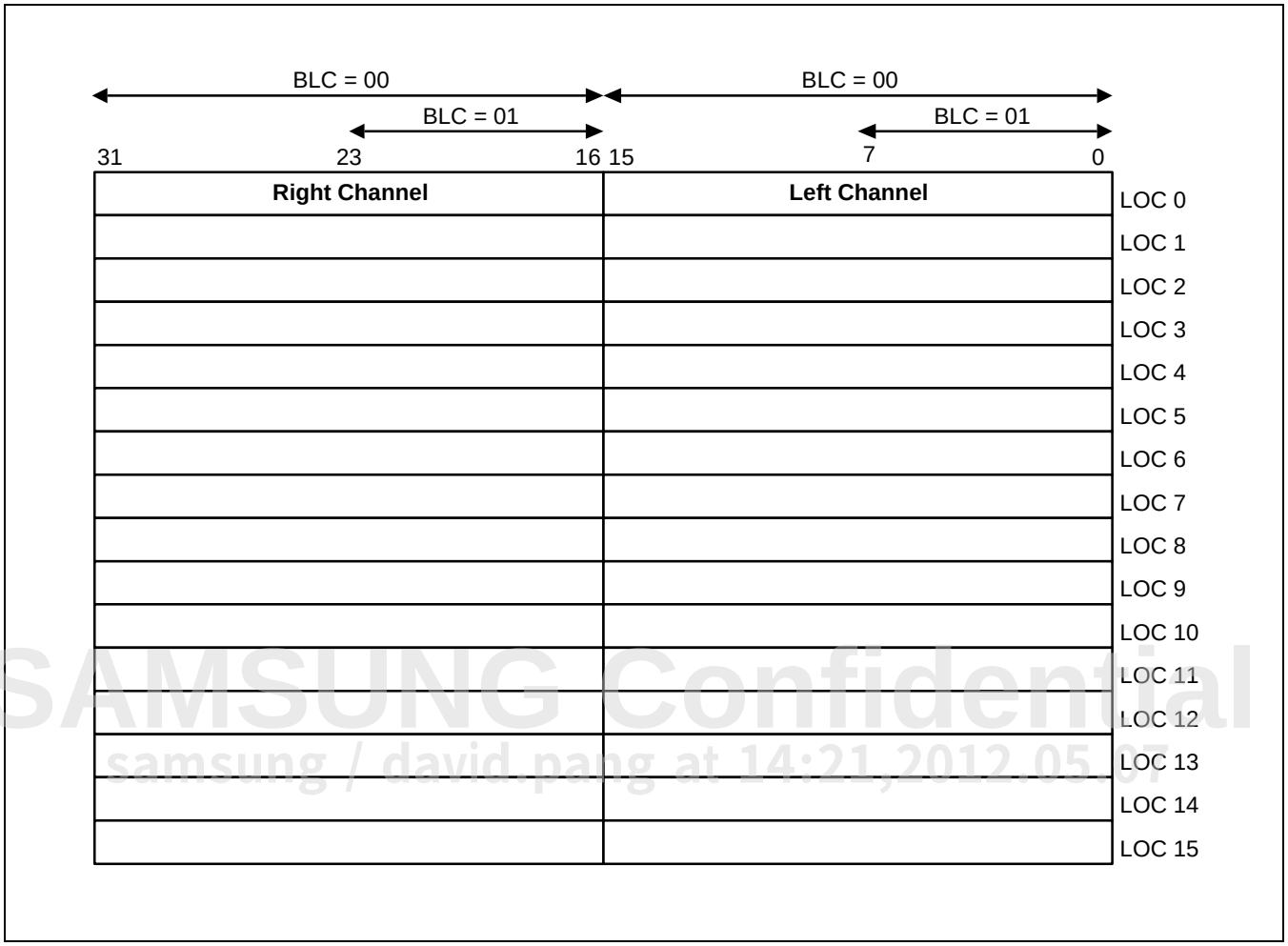


Figure 35-7 Tx FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in Tx FIFO for 24-bit/Channel BLC.

Figure 35-8 illustrates the Tx FIFO structure for BLC = 10 (24-bit/channel).

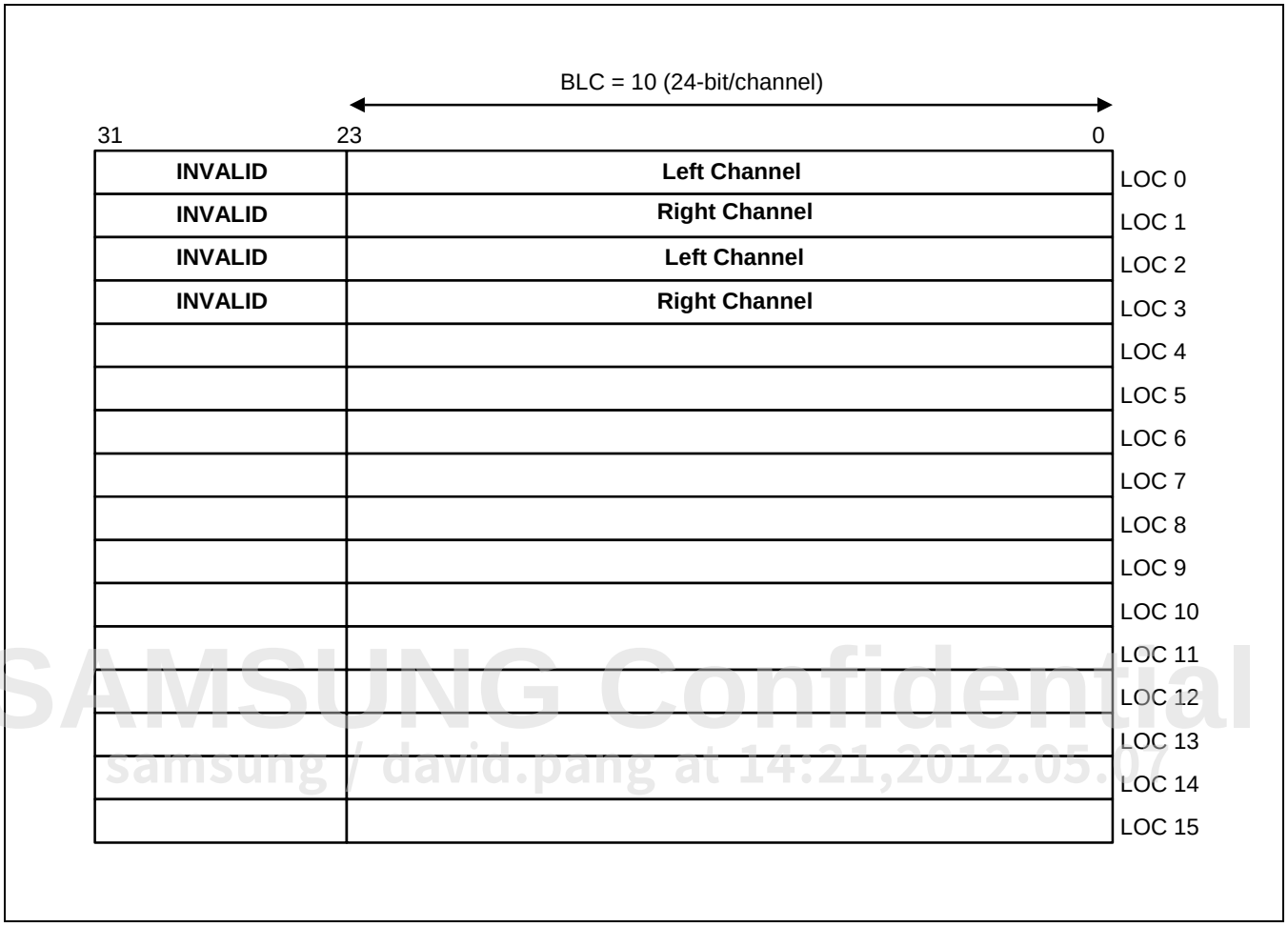


Figure 35-8 Tx FIF0 Structure for BLC = 10 (24-bit/Channel)

Once the data is written to Tx FIFO the Tx channel can be made active by enabling I2SACTIVE bit in I2SCON Register (IIS Control Register).
The data is then serially shifted out with respect to bit clock SCLK and word select clock LRCLK.

The TXCHPAUSE in I2SCON Register (IIS Control Register) can stop serial data transmission on I2SSDO.The transmission stops after the current Left/Right channel is transmitted.
If the control registers in I2SCON Register (IIS Control Register) and I2SMOD Register (IIS Mode Register) then it is advisable to disable Tx channel.

If the Tx channel is enabled while the FIFO is empty, no samples are read from the FIFO.
The Status of Tx FIFO can be checked by verifying the bits in I2SFIC Register (IIS FIFO Control Register).

35.7.4.2 Rx Channel

The IIS Rx channel provides a single stereo compliant output. The receive channel can operate in master or slave mode. Data is received from input line and transfers into Rx FIFO. The processor can then read this data through an APB read or a DMA access can access this data.

Rx Channel has 64 × 32-bit wide Rx FIFO where the processor or DMA can read up to 16 left/right data samples after enabling channel for reception.

An example sequence is as follows:

Ensure Audio bus clock and CDCLK are coming correctly to IIS controller and Flush the Rx FIFO using RFLUSH bit in I2SFIC Register (IIS FIFO Control Register).I2S controller is configured in any of the modes:

- Receive only.
- Receive/Transmit simultaneous mode

This is done by Programming TXR bit in I2SMOD Register (IIS Mode Register)

1. Then program these parameters according to the requirement:
   - MSS, RCLKSRC
   - SDF
   - BFS
   - BLC
   - LRP: For programming the above mentioned fields refer to I2SMOD Register (IIS Mode Register)
2. After ensured that input clocks for IIS controller are up and running and step 1 and 2 have been completed user must put I2SACTIVE high to enable any reception of data.
   The IIS Controller receives data on the LRCLK change.

The Data must be read from Rx FIFO using I2SRXD Register (IIS Rx FIFO Register) only after looking at the Rx FIFO count in the I2SFIC Register (IIS FIFO Control Register). The count would only increment once the complete left channel and right have been received.

The Data is aligned in Rx FIFO for 8 bits/channel or 16 bits/Channel BLC.

Figure 35-9 illustrates the Rx FIFO structure for BLC = 00 or BLC = 01.

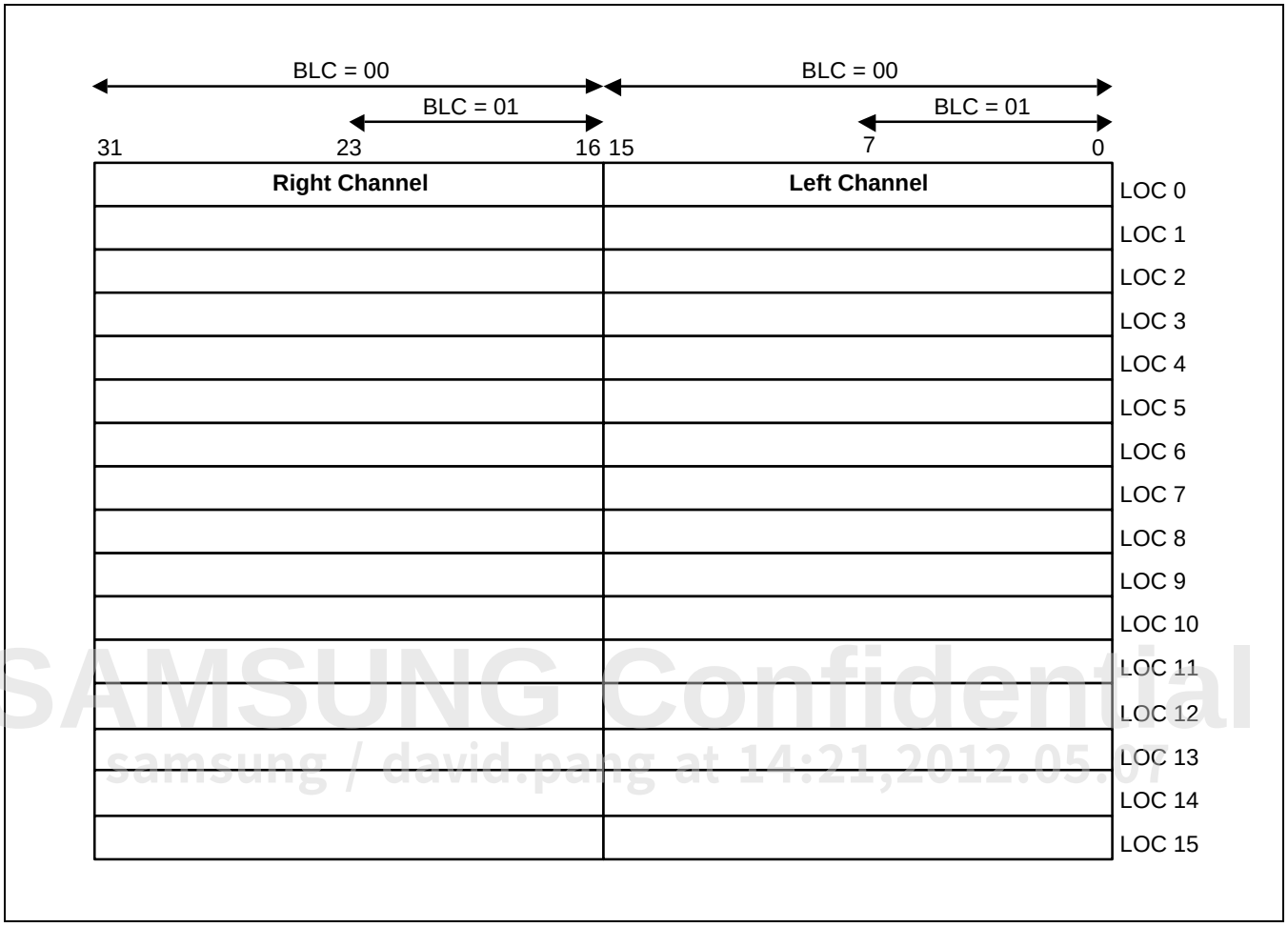


Figure 35-9 Rx FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in Rx FIFO for 24-bit/Channel BLC.

Figure 35-10 illustrates the Rx FIFO structure for BLC = 10 (24-bit/channel).

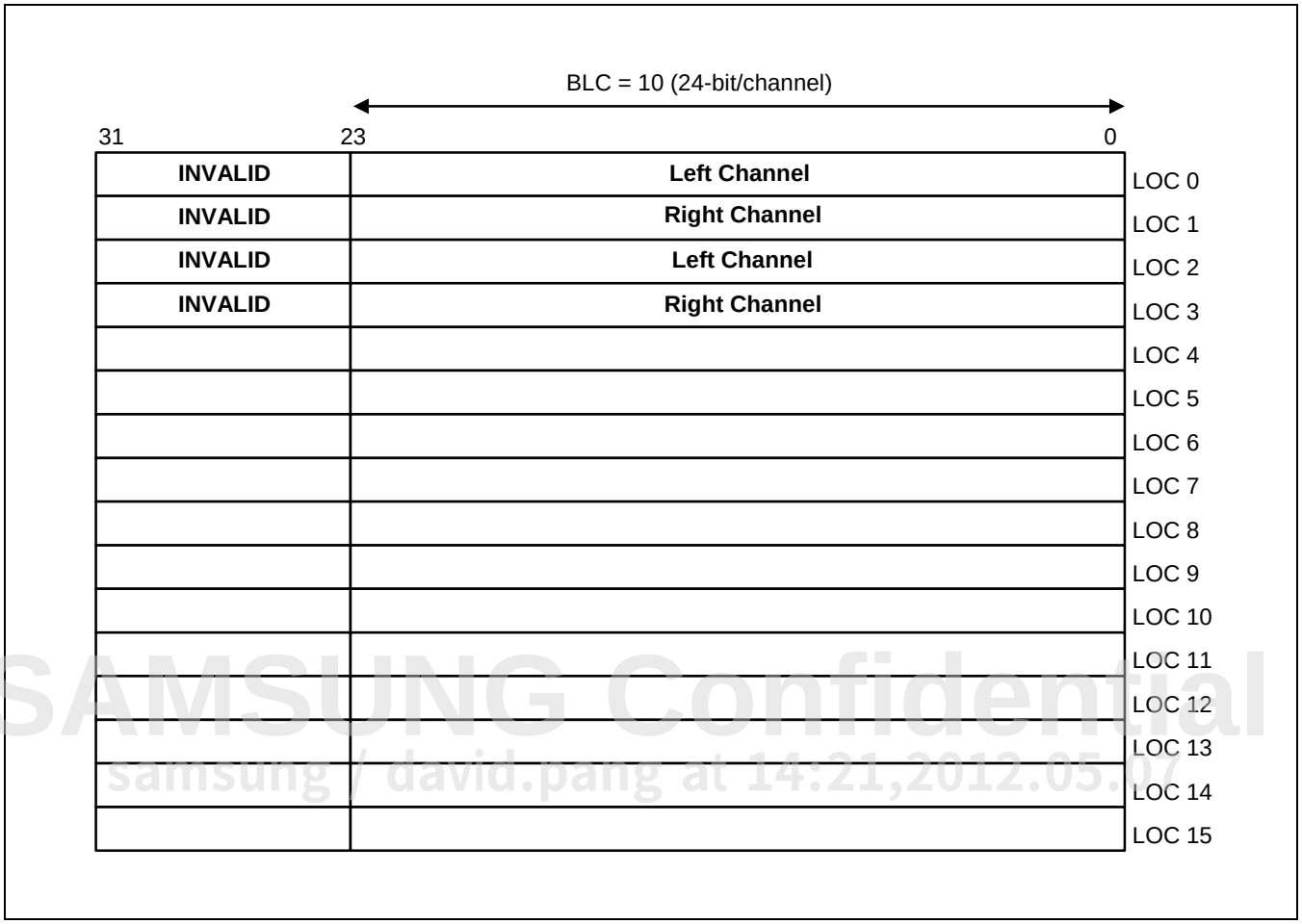


Figure 35-10 Rx FIF0 Structure for BLC = 10 (24-bit/Channel)

The RXCHPAUSE in I2SCON register can stop serial data reception on I2SSDI. The reception is stopped once the current Left/Right channel is received.
If control registers in I2SCON Register (IIS Control Register) and I2SMOD Register (IIS Mode Register) then it is advisable to disable Rx channel.

The Status of Rx FIFO can be checked by verifying the bits in I2SFIC Register (IIS FIFO Control Register).

35.8 IO Description

Signal	I/O	Description	Pad	Type
I2S_0_SCLK	I/O	Specifies bit clock.	Xi2s0SCLK	Dedicated
I2S_0_LRCK	I/O	Specifies LR channel clock.	Xi2s0LRCLK	Dedicated
I2S_0_CDCLK	I/O	Specifies codec clock.	Xi2s0CDCLK	Dedicated
I2S_0_SDI	I	Specifies I2S serial data input.	Xi2s0SDI	Dedicated
I2S_0_SDO[0]	O	Specifies I2S serial data out 0.	Xi2s0SDO_0	Dedicated
I2S_0_SDO[1]	O	Specifies I2S serial data out 1.	Xi2s0SDO_1	Dedicated
I2S_0_SDO[2]	O	Specifies I2S serial data out 2.	Xi2s0SDO_2	Dedicated

35.9 Register Description

35.9.1 Register Map Summary

- Base Address: 0x0383_0000

Register	Offset	Description	Reset Value
IISCON	0x0000	Specifies IIS interface control register	0x000
IISMOD	0x0004	Specifies IIS interface mode register	0x0
IISFIC	0x0008	Specifies IIS interface primary Tx FIFO & Rx FIFO control register	0x0
IISPSR	0x000C	Specifies IIS interface clock divider control register	0x0
IISTXD	0x0010	Specifies IIS interface transmit primary sound data register	0x0
IISRXD	0x0014	Specifies IIS interface receive data register	0x0
IISFICS	0x0018	Specifies IIS interface secondary TXFIFO_S control register	0x0
IISTXDS	0x001C	Specifies IIS interface secondary transmit data register	0x0
IISAHB	0x0020	Specifies IIS AHB DMA control register	0x0
IISSTR0	0x0024	Specifies IIS AHB DMA start address0 register	0x0
IISSIZE	0x0028	Specifies IIS AHB DMA size register	0x7FFF_0000
IISTRNCNT	0x002C	Specifies IIS AHB DMA transfer count register	0x0
IISLVL0ADDR	0x0030	Specifies IIS AHA DMA interrupt level 0 register	0x0000_0000
IISLVL1ADDR	0x0034	Specifies IIS AHA DMA interrupt level 1 register	0x0000_0000
IISLVL2ADDR	0x0038	Specifies IIS AHA DMA interrupt level 2 register	0x0000_0000
IISLVL3ADDR	0x003C	Specifies IIS AHA DMA interrupt level 3 register	0x0000_0000
IISSTR1	0x0040	Specifies IIS AHB DMA start address1 register	0x0

NOTE: All registers of IIS interface are accessible by word unit with STR/LDR instructions.

35.9.1.1 IISCON

- Base Address: 0x0383_0000
- Address = Base Address + 0x0000, Reset Value = 0x000

Name	Bit	Type	Description	Reset Value
SW_RST	[31]	RW	IIS software reset control. This should be set to 1 after IIS clock is stable. 0 = Resets IIS module (default) 1 = Un-reset IIS module Before reading SFR of IIS, you should set this bit.	0
RSVD	[30:27]	R	Reserved	0x0
FRXOFSTATUS	[26]	RW	Rx FIFO Over Flow Interrupt Status. And this is used by interrupt clear bit. When this is high, you can do interrupt clear by writing "1". 0 = Interrupt does not occur. 1 = Interrupt occurs	0
FRXOFINTEN	[25]	RW	Enables Rx FIFO Overflow Interrupt 0 = Disables Rx FIFO Overflow INT 1 = Enables Rx FIFO Overflow INT	1
FTXSURSTATUS	[24]	RW	Secondary Tx FIFO_S under-run interrupt status. This is used by interrupt clear bit. When this is high, you can clear interrupt clear by writing "1". 0 = Interrupt does not occur. 1 = Interrupt occurs.	0
FTXSURINTEN	[23]	RW	Secondary Tx FIFO_S Under-run Interrupt Enable 0 = Disables TXFIFO_S Under-run INT 1 = Enables TXFIFO_S Under-run INT	1
FTXSEMPT	[22]	R	Secondary Tx FIFO_S empty Status Indication 0 = TX FIFO_S is not empty(Ready to transmit Data) 1 = TX FIFO_S is empty(Not Ready to transmit Data)	1
FTXSFULL	[21]	R	Secondary Tx FIFO_S full Status Indication 0 = TX FIFO_S is not full 1 = TX FIFO_S is full	0
TXSDMAPAUSE	[20]	RW	Tx External DMA operation for secondary Tx FIFIO_S pause command. Note that when this bit is activated, the External DMA request halts after current on-going External DMA transfer is complete. 0 = No pause External DMA operation for Tx FIFO_S 1 = Pause External DMA operation for Tx FIFO_S NOTE: IISDMAEN SFR performs Internal DMA stop control.	0
RSVD	[19]	RW	Reserved (This value must be 0)	0
TXSDMACTIVE	[18]	RW	Tx External DMA active for secondary Tx FIFO_S (start External DMA request).	0

Name	Bit	Type	Description	Reset Value
			Note that when this bit is set from high to low, the External DMA operation will be forced to stop immediately. 0 = Inactive 1 = Active	
FTXURSTATUS	[17]	RW	Primary Tx FIFOx under-run interrupt status. This is used by interrupt clear bit. When this is high, you can clear interrupt by writing "1". 0 = Interrupt does not occur. 1 = Interrupt occurs.	0
FTXURINTEN	[16]	RW	Primary Tx FIFOx Under-run Interrupt Enable 0 = Disables TXFIFO Under-run INT 1 = Enables TXFIFO Under-run INT	0
FTX2EMPT	[15]	R	Primary Tx FIFO2 empty Status Indication 0 = Tx FIFO2 is not empty (Ready to transmit Data) 1 = Tx FIFO2 is empty (Not Ready to transmit Data)	0
FTX1EMPT	[14]	R	Primary Tx FIFO1 empty Status Indication 0 = Tx FIFO1 is not empty (Ready to transmit Data) 1 = Tx FIFO1 is empty (Not Ready to transmit Data)	0
FTX2FULL	[13]	R	Primary Tx FIFO2 full Status Indication 0 = Tx FIFO2 is not full 1 = Tx FIFO2 is full	0
FTX1FULL	[12]	R	Primary Tx FIFO1 full Status Indication 0 = Tx FIFO1 is not full 1 = Tx FIFO1 is full	0
LRI	[11]	R	Left/Right channel clock indication. Note that LRI meaning is dependent on value of LRP bit of I2SMOD register. 0 = Left (when LRP bit is low) or right (when LRP bit is high) 1 = Right (when LRP bit is low) or left (when LRP bit is high)	0
FTX0EMPT	[10]	R	Primary Tx FIFO0 empty status indication. 0 = FIFO is not empty (ready for transmit data to channel) 1 = FIFO is empty (not ready for transmit data to channel)	0
FRXEMPT	[9]	R	Rx FIFO empty status indication. 0 = FIFO is not empty 1 = FIFO is empty	0
FTX0FULL	[8]	R	Primary Tx FIFO0 full status indication. 0 = FIFO is not full 1 = FIFO is full	0
FRXFULL	[7]	R	Rx FIFO full status indication.	0

Name	Bit	Type	Description	Reset Value
			0 = FIFO is not full (ready for receive data from channel) 1 = FIFO is full (not ready for receive data from channel)	
TXDMAPAUSE	[6]	RW	Tx DMA operation pause command for primary Tx FIFOx. Note that when this bit is activated, the DMA request halts after current on-going DMA transfer is completed. 0 = No pause DMA operation for Tx FIFOx 1 = Pause DMA operation for Tx FIFOx	0
RXDMAPAUSE	[5]	RW	Rx DMA operation pause command. Note that when this bit is activated, the DMA request halts after current on-going DMA transfer is completed. 0 = No pause DMA operation 1 = Pause DMA operation	0
TXCHPAUSE	[4]	RW	Tx channel operation pause command for primary Tx FIFOx. Note that when this bit is activated, the channel operation halts after left-right channel data transfer is completed. 0 = No pause operation for Tx FIFOx and TX_S FIFO 1 = Pause operation for Tx FIFOx an TX_S FIFO	0
RXCHPAUSE	[3]	RW	Rx channel operation pause command. Note that when this bit is activated, the channel operation halts after left-right channel data transfer is completed. 0 = No pause operation 1 = Pause operation	0
TXDMACTIVE	[2]	RW	Tx DMA active for primary Tx FIFOx (start DMA request). Note that when this bit is set from high to low, the DMA operation will be forced to stop immediately. 0 = Inactive 1 = Active	0
RXDMACTIVE	[1]	RW	Rx DMA active (start DMA request). Note that when this bit is set from high to low, the DMA operation will be forced to stop immediately. 0 = Inactive 1 = Active	0
I2SACTIVE	[0]	RW	IIS interface active (start operation). 0 = Inactive 1 = Active	0

35.9.1.2 IISMOD

- Base Address: 0x0383_0000
- Address = Base Address + 0x0004, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
OP_CLK	[31:30]	RW	Operation clock for logic using Core CLK (green region at *Figure 35-1*). 00 = Codec clock out 01 = Codec clock in 10 = Bit clock out 11 = BUSCLK	00
RSVD	[29]	R	Reserved	0
OP_MUX_SEL	[28]	RW	Mux selection for secondary Tx FIFO_S 0 = Tx FIFO_S gets data from APB SFR interface 1 = Tx FIFO_S gets data from internal DMA interface Before trying to change this field from 1 to 0, software must poll IISTRNCNT register to confirm that all transfer is done according to internal DMA setting. There is no restriction on switching from 0 to 1.	0
BLC_S	[27:26]	RW	Bit Length Control Bit which decides transmission of 8/16/24 bits per audio channel for Secondary Tx FIFO_S 00 = 16 bits per channel 01 = 8 bits per channel 10 = 24 bits per channel 11 = Reserved	00
BLC_P	[25:24]	RW	Bit Length Control Bit which decides transmission of 8/16/24 bits per audio channel for Primary Tx FIFOx 00 = 16 bits per channel 01 = 8 bits per channel 10 = 24 bits per channel 11 = Reserved	00
RSVD	[23:22]	R	Reserved	00
CDD2	[21:20]	RW	Channel-2 Data Discard. Discard means zero padding. It only supports 8/16-bit mode. 00 = No Discard 01 = I2STXD[15:0] Discard 10 = I2STXD[31:16] Discard 11 = Reserved	00
CDD1	[19:18]	RW	Channel-1 Data Discard. Discard means zero padding. It only supports 8/16-bit mode. 00 = No Discard 01 = I2STXD[15:0] Discard 10 = I2STXD[31:16] Discard 11 = Reserved	00
DCE	[17:16]	RW	Enables Data Channel	00

Name	Bit	Type	Description	Reset Value
			[17]: Enables SD2 channel [16]: Enables SD1 channel	
RSVD	[15]	R	Reserved	0
BLC	[14:13]	RW	Bit Length Control Bit which decides transmission of 8/16/24 bits per audio channel for final mixed sound Tx output or Rx input. 00 = 16 bits per channel 01 = 8 bits per channel 10 = 24 bits per channel 11 = Reserved	00
CDCLKCON	[12]	RW	Determine direction of codec clock source (I2S_CDCLK) 0 = Supply RCLK to I2S_CDCLK (external codec chip) 1 = Get clock (to CLKAUDIO) from I2S_CDCLK (external codec chip) (Refer to *Figure 35-3*)	0
MSS	[11]	RW	Master or slave mode select 0 = Master mode 1 = Slave mode	0
RCLKSRC	[10]	RW	Select RCLK clock source 0 = Using BUSCLK 1 = Using I2SCLK (Refer to *Figure 35-3*)	0
TXR	[9:8]	RW	Transmit or receive mode select 00 = Transmit only mode 01 = Receive only mode 10 = Transmit and receive simultaneous mode 11 = Reserved	00
LRP	[7]	RW	Left/Right channel clock polarity select 0 = Low for left channel and high for right channel 1 = High for left channel and low for right channel	0
SDF	[6:5]	RW	Serial Data Format 00 = IIS format 01 = MSB-justified (left-justified) format 10 = LSB-justified (right-justified) format 11 = Reserved	00
RFS	[4:3]	RW	IIS root clock (codec clock) frequency select. 00 = 256 fs, where fs is sampling frequency 01 = 512 fs 10 = 384 fs 11 = 768 fs NOTE: Even in the slave mode, this bit should be set for correct operation.	00
BFS	[2:1]	RW	Bit clock frequency select. 00 = 32 fs, where fs is sampling frequency 01 = 48 fs	00

Name	Bit	Type	Description	Reset Value
			10 = 16 fs 11 = 24 fs NOTE: Even in the slave mode, this bit should be set for correct operation.	
RSVD	[0]	R	Reserved	0

35.9.1.3 IISFIC

- Base Address: 0x0383_0000
- Address = Base Address + 0x0008, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31]	W	Reserved	0
FTX2CNT	[30:24]	R	Primary TX FIFO2 data count. FIFO has 64 depth, so value ranges from 0 to 64. N: Data count N of FIFO	0x00
RSVD	[23]	R	Reserved	0
FTX1CNT	[22:16]	R	Primary TX FIFO1 data count. FIFO has 64 depth, so value ranges from 0 to 64. N: Data count N of FIFO	0x00
TFLUSH	[15]	RW	Primary TX FIFO flush command. 0 = No flush 1 = Flush	0
FTX0CNT	[14:8]	R	Primary TX FIFO0 data count. FIFO has 64 dept, so value ranges from 0 to 64. N: Data count N of FIFO	0x00
RFLUSH	[7]	RW	RX FIFO flush command. 0 = No flush 1 = Flush	0
FRXCNT	[6:0]	R	RX FIFO data count. FIFO has 64 dept, so value ranges from 0 to 64. N: Data count N of FIFO	0x00

35.9.1.4 IISPSR

- Base Address: 0x0383_0000
- Address = Base Address + 0x000C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	R	Reserved	0x00
PSRAEN	[15]	RW	Pre-scalar (Clock divider) active. 0 = Inactive (source clock is bypassed.) 1 = Active	0
RSVD	[14]	R	Reserved	0
PSVALA	[13:8]	RW	Pre-scalar (Clock divider) a division value. N: Division factor is N + 1	0x00
RSVD	[7:0]	R	Reserved	0x00

35.9.1.5 IISTXD

- Base Address: 0x0383_0000
- Address = Base Address + 0x0010, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
IISTXD	[31:0]	W	Primary Tx FIFO write data. Note that left/right channel data is allocated as the following bit fields. R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC Refer to *Figure 35-7* when 24-bit BLC	0x00

35.9.1.6 IISRXD

- Base Address: 0x0383_0000
- Address = Base Address + 0x0014, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
IISRXD	[31:0]	R	RX FIFO read data. Note that left/right channel data is allocated as the following bit fields. R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC Refer to *Figure 35-9* when 24-bit BLC	0x00

35.9.1.7 IISFICS

- Base Address: 0x0383_0000
- Address = Base Address + 0x0018, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	R	Reserved	0x00
TFLUSHS	[15]	RW	Secondary Tx FIFO_S flush command. 0 = No flush 1 = Flush	0
FTXSCNT	[14:8]	R	Secondary TX FIFO_S data count. FIFO has 64 depth, so value ranges from 0 to 64. N: Data count N of FIFO	0x00
RSVD	[7:0]	R	Reserved	0x00

35.9.1.8 IISTXDS

- Base Address: 0x0383_0000
- Address = Base Address + 0x001C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
IISTXDS	[31:0]	W	Secondary Tx FIFO_S write data. Note that left/right channel data is allocated as the following bit fields. R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC Refer to *Figure 35-7* when 24-bit BLC	0x00

35.9.1.9 IISAHB

- Base Address: 0x0383_0000
- Address = Base Address + 0x0020, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	R	Reserved	0x00
IISLVL3EN	[27]	RW	Enables buffer level 3 interrupt. 0 = Disables IISLVL3INT 1 = Enables IISLVL3INT	0
IISLVL2EN	[26]	RW	Enables buffer level 2 interrupt. 0 = Disables IISLVL2INT 1 = Enables IISLVL2INT	0
IISLVL1EN	[25]	RW	Enable buffer level 1 interrupt. 0 = Disables IISLVL1INT 1 = Enables IISLVL1INT	0
IISLVL0EN	[24]	RW	Enable buffer level 0 interrupt. 0 = Disables IISLVL0INT 1 = Enables IISLVL0INT	0
IISLVL3INT	[23]	R	Buffer level 3 interrupt status flag. During operation of DMA, when generated address in DMA matches with IISLVL3ADDR, this flag will be set. To clear this flag, use IISLVL3CLR field.	0
IISLVL2INT	[22]	R	Buffer level 2 interrupt status flag. During operation of DMA, when generated address in DMA matches with IISLVL2ADDR, this flag will be set. To clear this flag, use IISLVL2CLR field.	0
IISLVL1INT	[21]	R	Buffer level 1 interrupt status flag. During operation of DMA, when generated address in DMA matches with IISLVL1ADDR, this flag will be set. To clear this flag, use IISLVL1CLR field.	0
IISLVL0INT	[20]	R	Buffer level 0 interrupt status flag. During operation of DMA, when generated address in DMA matches with IISLVL0ADDR, this flag will be set. To clear this flag, use IISLVL0CLR field.	0
IISLVL3CLR	[19]	WO	Clear IISLVL3INT flag When IISLVL3INT is set, setting IISLVL3CLR to 1 will clear IISLVL3INT to 0. Writing zero has no effect.	0
IISLVL2CLR	[18]	WO	Clear IISLVL2INT flag When IISLVL2INT is set, setting IISLVL2CLR to 1 will clear IISLVL2INT to 0. Writing zero has no effect.	0
IISLVL1CLR	[17]	WO	Clear IISLVL1INT flag When IISLVL1INT is set, setting IISLVL1CLR to 1 will clear IISLVL1INT to 0. Writing zero has no effect.	0
IISLVL0CLR	[16]	WO	Clear IISLVL0INT flag When IISLVL0INT is set, setting IISLVL0CLR to 1 will	0

Name	Bit	Type	Description	Reset Value
			clear IISLVL0INT to 0. Writing zero is has effect.	
RSVD	[15:8]	R	Reserved	0x00
IISDMA_STR ADDRRST	[7]	WO	DMA start address reset Before starting address toggles, write 1 to this bit. After reset, this bit is auto-cleared.	0x0
IISDMA_STR ADDRTOG	[6]	RW	DMA start address toggle 0 = Disables start address toggling (IISSTR0 → IISSTR0 → …) 1 = Enables start address toggling (IISSTR0 → IISSTR1 → IISSTR0 → IISSTR1 → …)	0
IISDMARLD-	[5]	RW	Auto-reload IIS internal DMA Configuration when DMA operation is done and re-starts IIS internal DMA automatically. 0 = Disables auto-reload function 1 = Enables auto-reload function Before switching to 0 from 1, software must check if DMA_EN is set.	0
IISINTMASK	[3]	RW	Disables interrupt request signal 0 = Enables interrupt request when DMA auto-reload is on. 1 = Disables interrupt request when DMA auto-reload is on. After DMA transfers all of data related to DMA configuration, interrupt signal will occur. If IISINTMASK bit is set, IISDMAINT and interrupt signal will not be set. IISINTMASK does not affect IISLVLxINT and under-run interrupt.	0
IISDMAINT	[2]	R	DMA interrupt status flag. After DMA operation is end, this flag will be set. To clear this flag, use IISDMACLR field. When ARM is used for decoder, do not use this interrupt as controlling timing of filling buffer. In that case, use level 0 to 3 interrupts. This interrupt is just used for ending condition.	0
IISDMACLR	[1]	WO	Clear DMA interrupt status flag When IISINT is set, setting IISDMACLR to 1 will clear IISDMAINT to 0. Writing to 0 is invalid.	0
IISDMAEN	[0]	RW	Enable IIS internal DMA You can use internal DMA in IIS after this bit field is ON. Internal DMA can issue 32-bit single read transaction for AHB and TXFIFO0 will hold data returned by DMA. Warning: If IISDMARLD is set, IISDMAEN bit will be automatically cleared when reload operation is in progress. After auto-reload operation is done, IISDMAEN bit will be	0

Name	Bit	Type	Description	Reset Value
			automatically set. When auto-reload operation is in progress, software intervention on this field will cause mal-function of internal DMA operations. To manipulate IISAHB register, software must verify that IISDMAEN is in stable state.	

35.9.1.10 IISSTR0

- Base Address: 0x0383_0000
- Address = Base Address + 0x0024, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
IISSTR	[31:0]	RW	Start address0 of IIS internal DMA operation. When DMAEN is ON, internal DMA in IIS will start DMA operation based on IISSTR0 address. Internal DMA can handle word-aligned address only but to get best performance, IISSTR0 should be 64 word-aligned address.	0x00

35.9.1.11 IISSIZE

- Base Address: 0x0383_0000
- Address = Base Address + 0x0028, Reset Value = 0x7FFF_0000

Name	Bit	Type	Description	Reset Value
TRNS_SIZE	[31:16]	RW	Transfer block size for IIS internal DMA When IIS internal DMA is enabled, IIS internal DMA transfers TRNS_SIZE word (s) data from memory before DMA done interrupt occurs. Valid ranges for TRNS_SIZE will be 0x0001 – 0xC000: DMEM and I$ at MAUDIO Sub-System (192Kbytes) 0x0001 – 0xFFFF: IRAM at MAUDIO Sub-System (256 Kbytes) This value can be dynamically changed while DMA is running. But before changing, read IISTRNCNT value and verify whether at least four words to transfer is remaining. If not, you cannot change this value at that time. Wait until IISTRNCNT is 0 and then change.	0x7FFF
RSVD	[15:0]	R	Reserved	0x0000

35.9.1.12 IISTRNCNT

- Base Address: 0x0383_0000
- Address = Base Address + 0x002C, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	R	Reserved	–
IISTRNCNT	[23:0]	R	Number of transferred data using IIS internal DMA. (word unit) User program to terminate IIS internal DMA operation by turning DMA_EN off. After DMA_EN is 0, user program reads IISTRNCNT value to know where IIS internal DMA stops.	–

35.9.1.13 IISLVL0ADDR

- Base Address: 0x0383_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
IISLVL0ADDR	[31:10]	R	AHB DMA level 0 interrupt address While IISLVL0EN in IISAHB register is set, AHB DMA is comparing this register to generated address in DMA. When two value matches, IISLVL0INT in IISAHB will be set.	0x00
RSVD	[9:1]	R	Reserved	0x00
IISLVL0STOP	[0]	RW	Enables Precise stop 0 = Do not stop DMA operation 1 = Stops DMA operation when DMA working address matches with IISLVL0ADDR. IISDMAEN in IISAHB will be turned off automatically.	0

35.9.1.14 IISLVL1ADDR

- Base Address: 0x0383_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
IISLVL1ADDR	[31:10]	R	AHB DMA level 1 interrupt address While IISLVL1EN in IISAHB register is set, AHB DMA is comparing this register to generated address in DMA. When two value matches, IISLVL1INT in IISAHB is set.	0x00
RSVD	[9:1]	R	Reserved	0x00
IISLVL1STOP	[0]	RW	Enables Precise stop 0 = Do not stop DMA operation 1 = Stops DMA operation when DMA working address is matched with IISLVL1ADDR. IISDMAEN in IISAHB will be turned off automatically.	0

35.9.1.15 IISLVL2ADDR

- Base Address: 0x0383_0000
- Address = Base Address + 0x0038, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
IISLVL2ADDR	[31:10]	R	AHB DMA level 2 interrupt address While IISLVL2EN in IISAHAB register is set, AHB DMA is comparing this register to generated address in DMA. When two values match, IISLVL2INT in IISAHB will be set.	0x00
RSVD	[9:1]	R	Reserved	0x00
IISLVL2STOP	[0]	RW	Enables Precise stop 0 = Do not stop DMA operation 1 = Stop DMA operation when DMA working address matches with IISLVL2ADDR. IISDMAEN in IISAHB will be turned off automatically.	0

35.9.1.16 IISLVL3ADDR

- Base Address: 0x0383_0000
- Address = Base Address + 0x003C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
IISLVL3ADDR	[31:10]	R	AHB DMA level 3 interrupt address While IISLVL3EN in IISAHB register is set, AHB DMA is comparing this register to generated address in DMA. When two value matches, IISLVL3INT in IISAHB is set.	0x00
RSVD	[9:1]	R	Reserved	0x00
IISLVL3STOP	[0]	RW	Enables Precise stop 0 = Do not stop DMA operation 1 = Stops DMA operation when DMA working address matches with IISLVL3ADDR. IISDMAEN in IISAHB will be turned off automatically.	0

35.9.1.17 IISSTR1

- Base Address: 0x0383_0000
- Address = Base Address + 0x0040, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
IISSTR1	[31:0]	RW	Start address1 of IIS internal DMA operation. When DMAEN is ON, internal DMA in IIS starts DMA operation based on IISSTR1 address. Internal DMA can handle word-aligned address only, but to achieve best performance, IISSTR1 should be 64 word-aligned address.	0x00

36 IIS-Bus Interface

36.1 Overview

Inter-IC Sound (IIS) is one of the popular digital audio interfaces.

The IIS bus handles only audio data and the other signals such as sub-coding and control are transferred seperately. Serial data, word select and clock signals are used to minimize the number of pins required and to keep wiring simple.

- Serial data signals (I2SSDO, I2SSDI)
- Word select signal (I2SLRCLK)
- Clock signals(I2SSCLK, I2SCDCLK)

IIS interface transmits or receives sound data from external stereo audio codec. To transmit and receive data, there're two 32x64 FIFOs (First-In-First-Out) data structures for transmitting and receiving data. DMA transfer mode can be supported to transmit and receive samples. From internal system clock controller through IIS clock divider or direct clock source.

36.2 Features

The features of IIS-Bus Interface are:

- 2-ports stereo(2 Channel) IIS-bus for audio interface with DMA-based operation
- Serial, 8/16/24-bit per channel data transfers
- Supports master/slave mode
- Supports IIS, MSB-justified, and LSB-justified data format

36.3 Block Diagram

Figure 36-1 illustrates IIS-Bus block diagram.

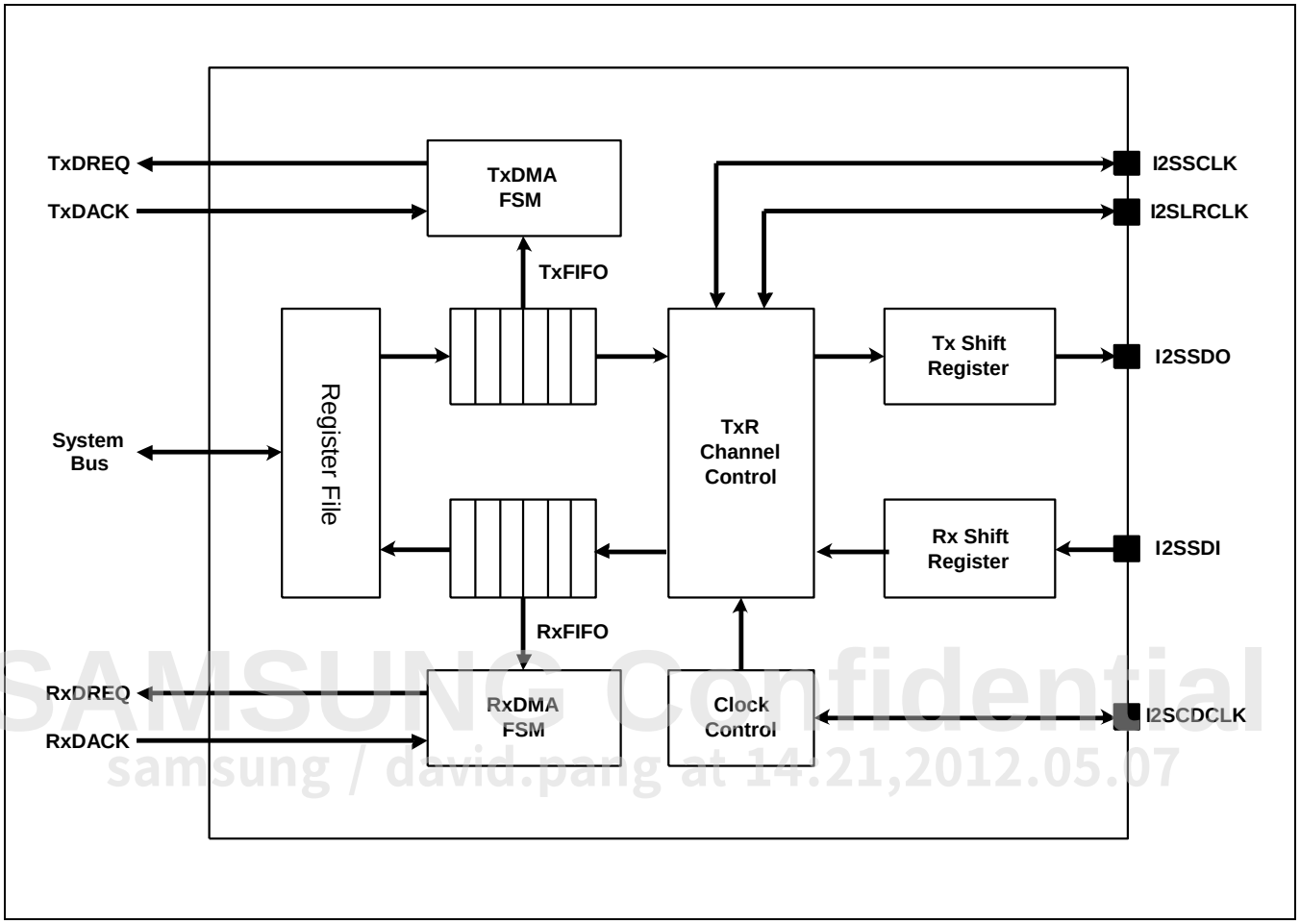


Figure 36-1 IIS-Bus Block Diagram

36.4 Functional Description

IIS interface consists of register bank, FIFOs, shift registers, clock control, DMA finite state machine, and channel control block as illustrated in *Figure 36-1*. Note that each FIFO has 32-bit width and 64 depth structure, which contains left/right channel data. Thus, FIFO access and data transfer are handled with left/right pair unit.

36.4.1 Master/Slave Mode

MSS bit of IISMOD register selects master or slave mode. Master generates I2SSCLK and I2SLRCLK that a root clock (RCLK) is needed for them. The root clock comes from either internal clock source or I2SCDCLK. Slave gets I2SSCLK and I2SLRCLK from Master.

IIS Tx or IIS Rx to generate I2SSCLK and I2SLRCLK is master while IIS Tx or IIS Rx to receive I2SSCLK and I2SLRCLK is slave.

For example, when IIS bus interface generates clock signals during transmitting data to IIS codec, the IIS bus interface becomes master. If IIS bus interface gets clock signals from IIS codec during transmitting data to IIS codec, the slave is IIS bus interface in this case.

Figure 36-2 illustrates the route of the root clock (RCLK) in master or slave mode setting in IIS clock control block and system controller.

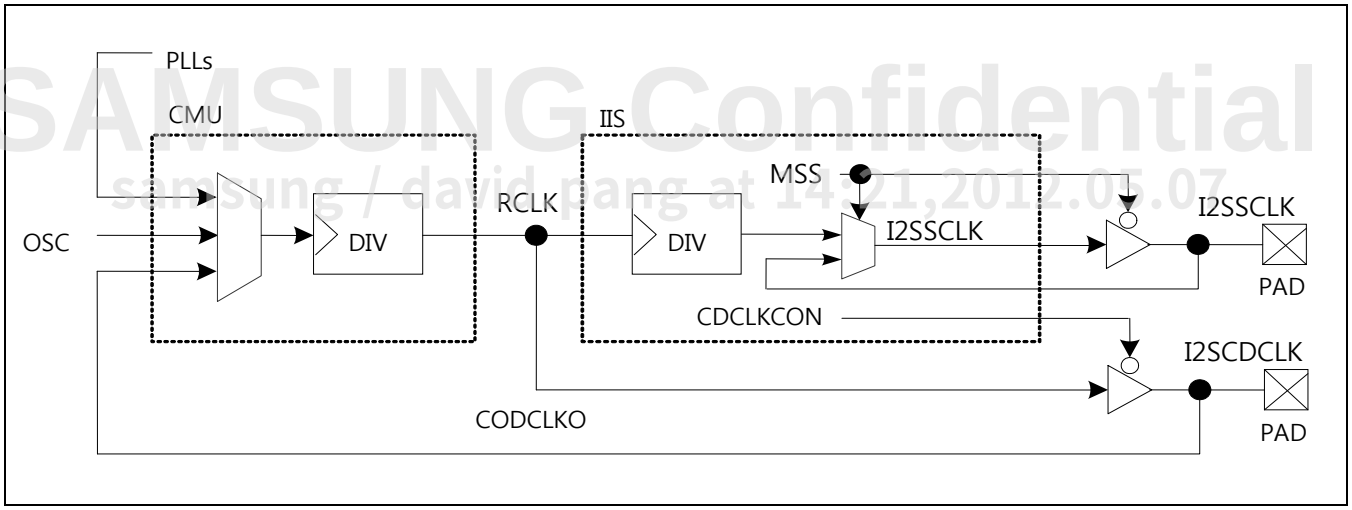


Figure 36-2 IIS Clock Control Block Diagram

36.4.2 DMA Transfer

In DMA transfer mode, use external DMA controller to access transmitter or receiver FIFO. The transmitter or receiver FIFO state activates the DMA service request internally. The FTXEMPT, FRXEMPT, FTXFULL, and FRXFULL bits of I2SCON register represents transmitter or receiver FIFO data state. Especially, FTXEMPT and FRXFULL bit are the ready flag for DMA service request. the DMA service request for transmitting is activated when TxFIFO is not empty and the DMA service request for receiving is activated when RxFIFO is not full.

The DMA transfer uses only handshaking method for single data. Note that during DMA acknowledge activation; the data read or write operation should be performed during DMA acknowledge activation.

NOTE: Reference: DMA request point

- Tx mode: (FIFO is not full) and (TXDMACTIVE is active)
- Rx mode: (FIFO is not empty) and (RXDMACTIVE is active)

36.4.3 Audio Serial Data Format

This section includes:

- IIS-bus Format
- MAB (Left) Justified
- LSB (Right) Justified
- Sampling Frequency and Master Clock

36.4.3.1 IIS-bus Format

The IIS bus has serial data input I2SSDI, serial data output I2SSDO, left/right channel select clock I2SLRCLK; master generates I2SLRCLK and I2SSCLK.

Serial data is transmitted in 2's complement with MSB first with a fixed position, whereas the position of LSB depends on word length. The transmitter sends MSB of next word at one clock period after the I2SLRCLK is changed. Serial data sent by transmitter can be synchronized either with trailing or with leading edge of clock signal. However, the serial data must be latched into receiver on the leading edge of serial clock signal. Therefore there are some restrictions when transmitting data that is synchronized with leading edge.

The LR channel select line indicates the channel being transmitted. I2SLRCLK may be changed either on a trailing or leading edge of serial clock, but it does not need to be symmetrical. In the slave, this signal is latched on the leading edge of the clock signal. The I2SLRCLK line changes one clock period before the MSB is transmitted. This allows slave transmitter to derive synchronous timing of serial data that will be set up for transmission. Furthermore, it enables receiver to store previous word and clear the input for next word.

36.4.3.2 MSB (Left) Justified

MSB-Justified (Left-Justified) format is similar to IIS bus format. In MSB-justified format, the transmitter always sends MSB of the next word simultaneously whenever the I2SLRCLK is changed.

36.4.3.3 LSB (Right) Justified

LSB-Justified (Right-Justified) format is opposite to MSB-justified format. In other word, the transferring serial data is aligned with ending point of I2SLRCLK transition.

Note that in this figure, the word length is 16-bit and I2SLRCLK makes transition every 24 cycle of I2SSCLK (BFS is 48 fs, where fs is sampling frequency; I2SLRCLK frequency).

Figure 36-3 illustrates the IIS audio serial data formats.

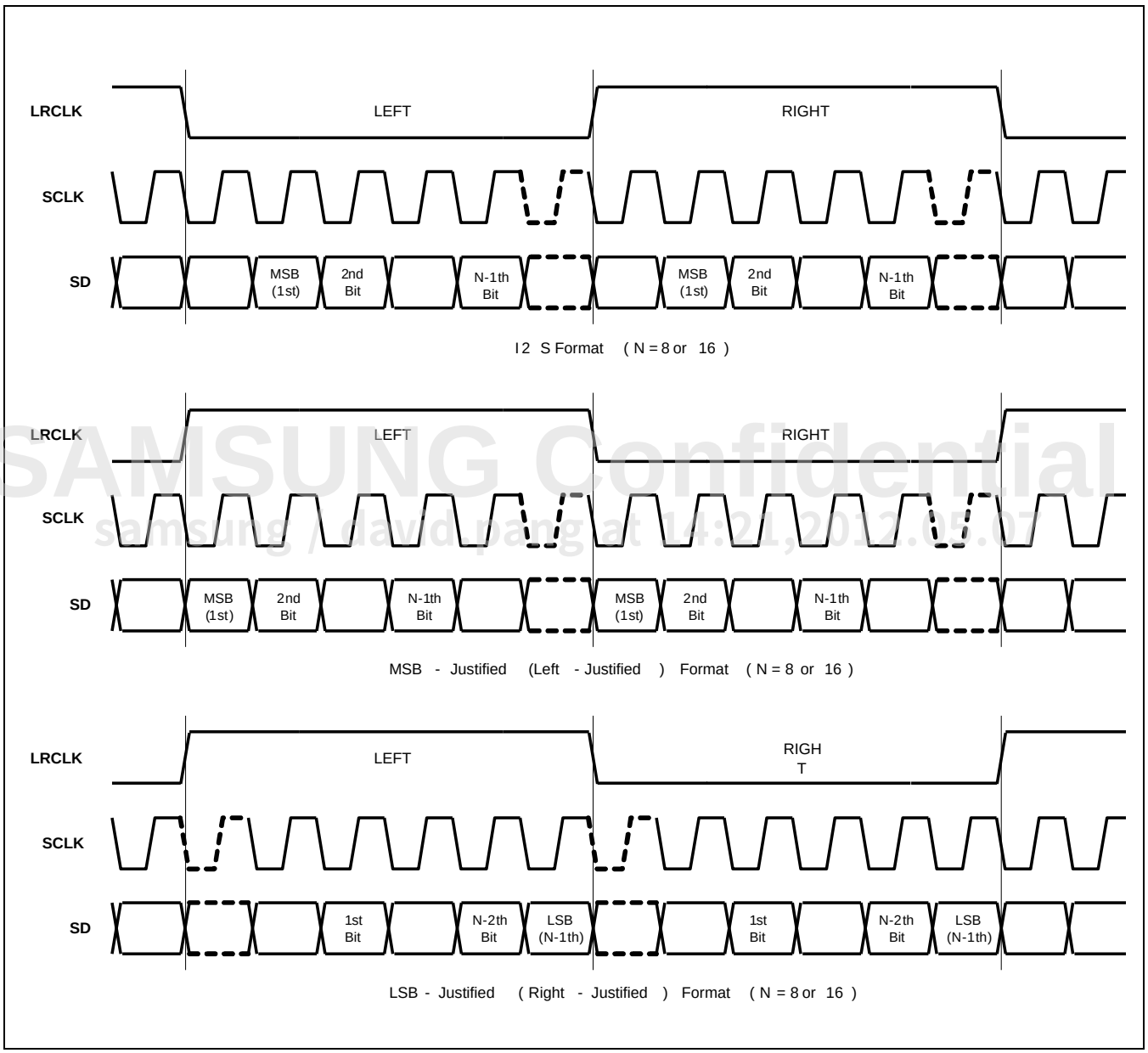


Figure 36-3 IIS Audio Serial Data Formats

36.4.3.4 Sampling Frequency and Master Clock

When IIS interface Controller operates as master, IIS interface Controller generates IISLRCLK and IISSCLK that are divided as Root Clock by RFS and BFS value. To decide Sampling Frequency – IISLRCLK –, BLC, BFS, and RFS are selected first. Optionally, IIS interface Controller clocks out Root clock as IISCDCLK for codec master clock (if source of root clock is not IISEXTCDCLK).

In slave mode, you should set the value of BLC, BFS, and RFS similar to master (ex: Codec). Because IIS interface controller requires these value for correct operation.

36.4.4 PCM Word Length and BFS Divider

PCM Word Length (BLC) setting should be preceded before setting BFS value.

Table 36-1 lists the BFS available value as BLC.

Table 36-1 Allowed BFS Value as BLC

PCM bit Length (BLC)	8-bit	16-bit	24-bit
Available BFS Value	16 fs, 24 fs, 32 fs, 48 fs	32 fs, 48 fs	48 fs

36.4.5 BFS Divider and RFS Divider

RFS value is selected as BFS selected.

Table 36-2 lists the RFS available value as BFS.

Table 36-2 Allowed RFS Value as BFS

BFS Divider	16 fs, 32 fs	24 fs, 48 fs
Available RFS Value	256 fs, 384 fs, 512 fs, 768 fs	384 fs, 768 fs

36.4.6 RFS Divider and Root Clock

Root Clock is made for sampling frequency proper RFS value as listed in *Table 36-3*. RCLK is clock divided by IIS pre-scaler (IISPSR) that is selected by MSS.

Table 36-3 lists the RFS value for Root Clock.

Table 36-3 Root Clock Table (MHz)

IISLRCK RFS	8.000 kHz	11.025 kHz	16.000 kHz	22.050 kHz	32.000 kHz	44.100 kHz	48.000 kHz	64.000 kHz	88.200 kHz	96.000 kHz
256fs	2.0480	2.8224	4.0960	5.6448	8.1920	11.2896	12.2880	16.3840	22.5792	24.5760
384fs	3.0720	4.2336	6.1440	8.4672	12.2880	16.9344	18.4320	24.5760	33.8688	36.8640
512fs	4.0960	5.6448	8.1920	11.2896	16.3840	22.5792	24.5760	32.7680	45.1584	49.1520
768fs	6.1440	8.4672	12.2880	16.9344	24.5760	33.8688	36.8640	49.1520	67.7376	73.7280

Root Clock Frequency = fs × (256, 384, 512 or 768)

36.5 Programming Guide

The IIS bus interface can be accessed either by the processor using programmed I/O instructions or by DMA controller.

36.5.1 Initialization

The procedure for initialization is:

1. Before you use IIS bus interface, you must configure GPIOs to IIS mode, that is, I2SSDI is input and I2SSDO is output. I2SLRCLK, I2SSCLK and I2SCDCLK is inout-type.
2. Select clock source. Exynos 4412 SCP has three clock sources. They are
   - PCLK
   - EPLL
   - External codec
     Refer to *Figure 36-2* for more information.

36.5.2 Play Mode (Tx Mode) with DMA

The procedure for Play Mode with DMA is:

1. TXFIFO is flushed before operation. If you do not distinguish Master/Slave mode from Tx/Rx mode, you should study Master/Slave mode and Tx/Rx mode. Refer to Master/Slave chapter for more information.
2. Configure I2SMOD register and I2SPSR (IIS pre-scaler register).
3. To operate system in stability, internal TxFIFO should be almost full before transmission. For TxFIFO to be almost full start DMA operation.
4. IIS bus does not support interrupt. Therefore, you can only verify state by polling through accessing SFR.
5. After TxFIFO is full, then I2SACTIVE must be asserted.

36.5.3 Recording Mode (Rx Mode) with DMA

The procedure for Recording Mode with DMA is:

1. RxFIFO is flushed before operation. Also, if you do not distinguish between Master/Slave mode and Tx/Rx mode, you should study Master/Slave mode and Tx/Rx mode. Refer to Master/Slave chapter for more information.
2. Configure I2SMOD register and I2SPSR (IIS pre-scaler register).
3. To operate system in stability, the internal RxFIFO should have at least one data before DMA operation. You should assert I2SACTIVE before DMA operation.
4. Verify RxFIFO state by polling through accessing SFR.
5. If RxFIFO is not empty, start RXDMACTIVE.

36.5.4 Example Code

This section includes:

- Tx Channel
- Rx Channel

36.5.4.1 Tx Channel

The I2S Tx channel provides a single stereo compliant output. The transmit channel can operate in master or slave mode. Data is transferred between processor and I2S controller through an APB access or a DMA access.

The processor writes words in multiples of two (that is for left and right audio sample).The words are serially shifted out timed with respect to audio serial bit clk, SCLK and word select clock, LRCLK.

Tx Channel has 64 × 32-bit wide FIFO where processor or DMA can write up to 16 left/right data samples after enabling channel for transmission.

An example sequence is:

Ensure PCLK and CDCLK are coming correctly to I2S controller and flush the Tx FIFO using TFLUSHH bit in the IISFIC register. Please ensure that I2S Controller is configured in one of the modes:

- Tx only mode
- Tx/Rx simultaneous mode

The Data is aligned in Tx FIFO for 8 bits/channel or 16 bits/channel BLC.

Figure 36-4 illustrates the Tx FIFO structure for BLC = 00 or BLC = 01.

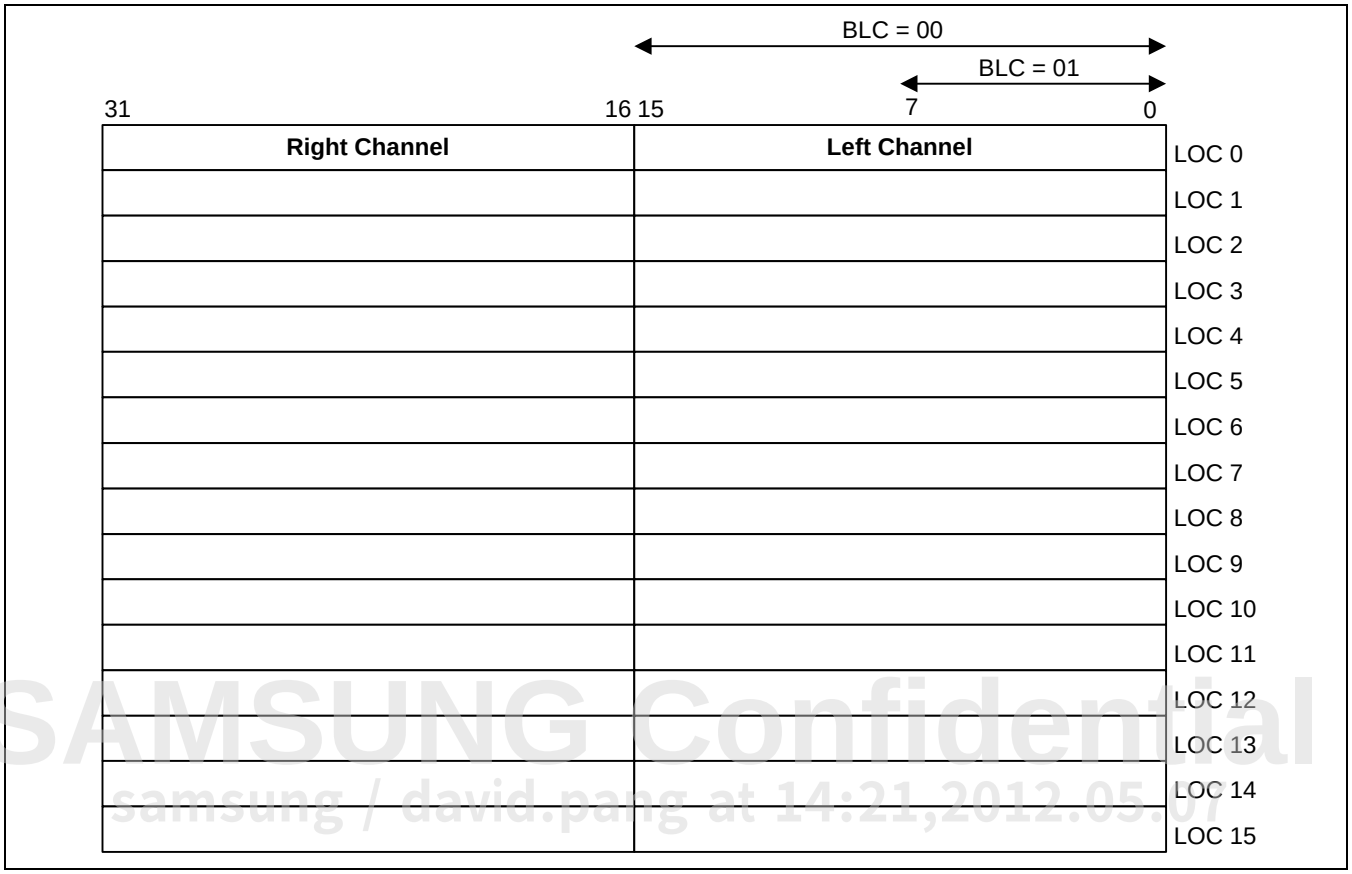


Figure 36-4 Tx FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the Tx FIFO for 24-bit/channel BLC.

Figure 36-5 illustrates the Tx FIFO structure for BLC = 10 (24-bit/channel).

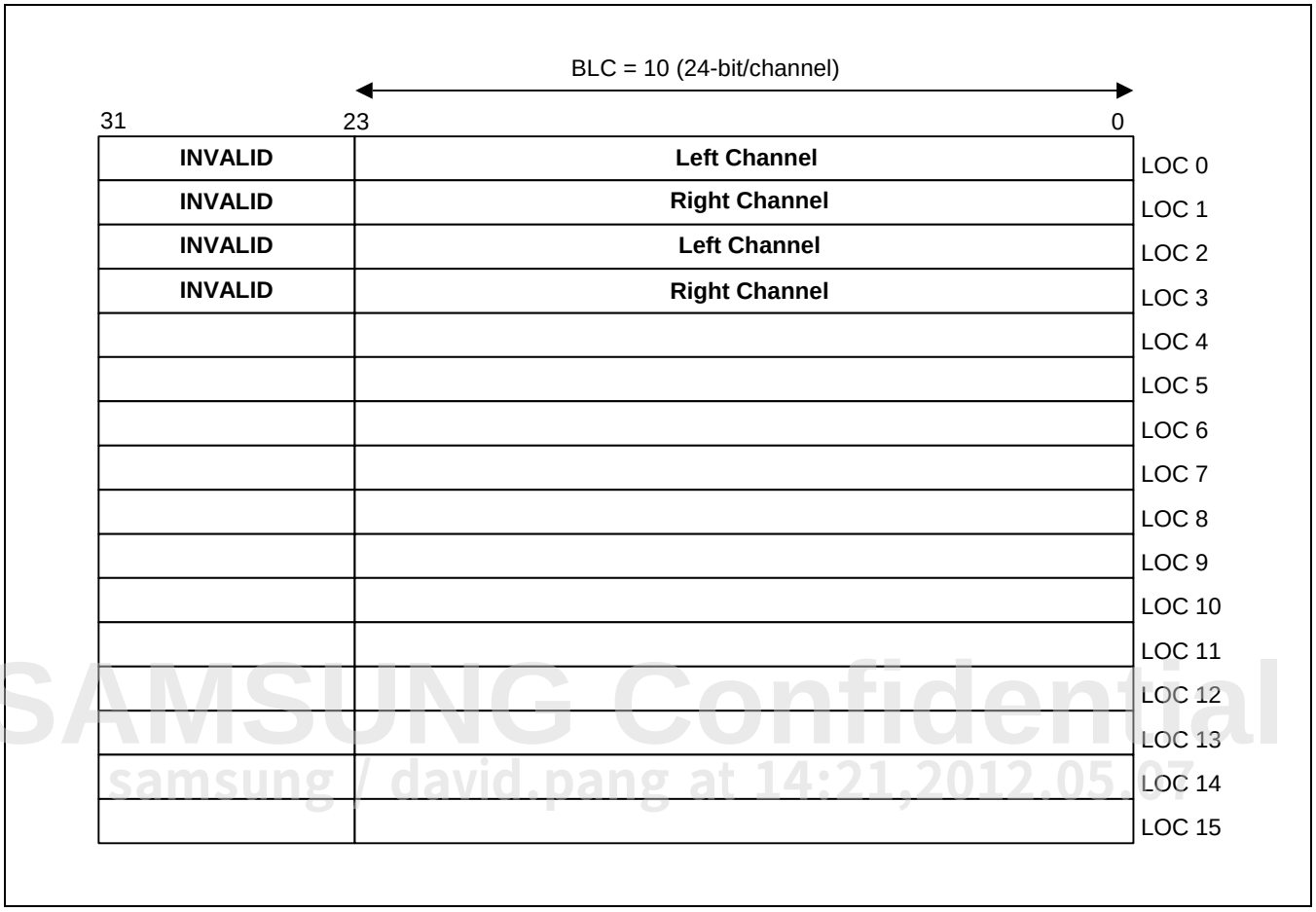


Figure 36-5 Tx FIF0 Structure for BLC = 10 (24-bit/Channel)

Once the data is written to the Tx FIFO the Tx channel can be made active by enabling the I2SACTIVE bit in I2SCON Register (I2S Control Register).

The data is then serially shifted out with respect to serial bit clock SCLK and word select clock LRCLK.

The TXCHPAUSE in I2SCON Register (I2S Control Register) can stop serial data transmission on I2SSDO.The transmission stops after the current Left/Right channel is transmitted.

If the control registers in the I2SCON Register (I2S Control Register) and I2SMOD Register (I2S Mode Register) then it is advisable to disable the Tx channel.

If the Tx channel is enabled while the FIFO is empty, no samples are read from the FIFO.

The Status of Tx FIFO can be checked by verifying the bits in the I2SFIC Register (I2S FIFO Control Register).

36.5.4.2 Rx Channel

The I2S Rx channel provides a single stereo compliant output. The receive channel can operate in master or slave mode. Data is received from the input line and transfers into Rx FIFO. The processor can then read this data through an APB read or a DMA access can access this data.

Rx Channel has a 64 × 32-bit wide Rx FIFO where the processor or DMA can read up to 16 left/right data samples after enabling the channel for reception.

An example sequence is:

Ensure PCLK and CDCLK are coming correctly to I2S controller and Flush the Rx FIFO using RFLUSH bit in I2SFIC Register (I2S FIFO Control Register). And the I2S controller is configured in any of the modes:

- Receive only.
- Receive/Transmit simultaneous mode

This is done by Programming TXR bit in I2SMOD Register (I2S Mode Register)

1. Then program these parameters according to the requirement:
   - MSS
   - SDF
   - BFS
   - BLC
   - LRP

   For programming, the above mentioned fields please refer to I2SMOD Register (I2S Mode Register)
2. After ensured that input clocks for I2S controller are up and running and step 1 and 2 have been completed user must put the I2SACTIVE high to enable any reception of data, the I2S Controller receives data on the LRCLK change.

Read data from Rx FIFO using the I2SRXD Register (I2S Rx FIFO Register) after looking at the Rx FIFO count in the I2SFIC Register (I2S FIFO Control Register). The count would only increment once the complete left channel and right have been received.

Figure 36-6 illustrates the Rx FIFO structure for BLC = 00 or BLC = 01.

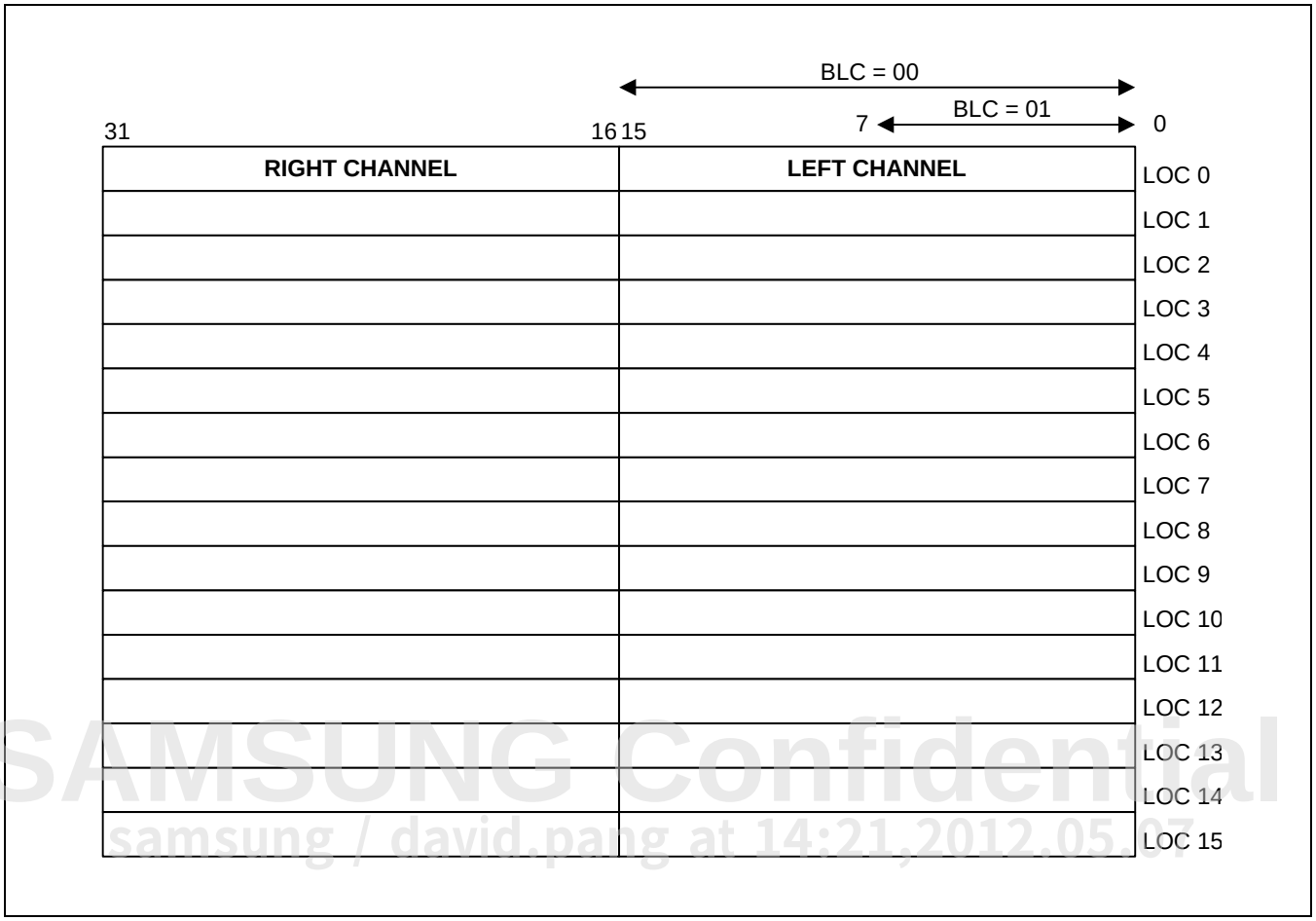


Figure 36-6 Rx FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the Rx FIFO for 24-bit/channel BLC.

Figure 36-7 illustrates the Rx FIFO structure for BLC = 10 (24-bit/channel).

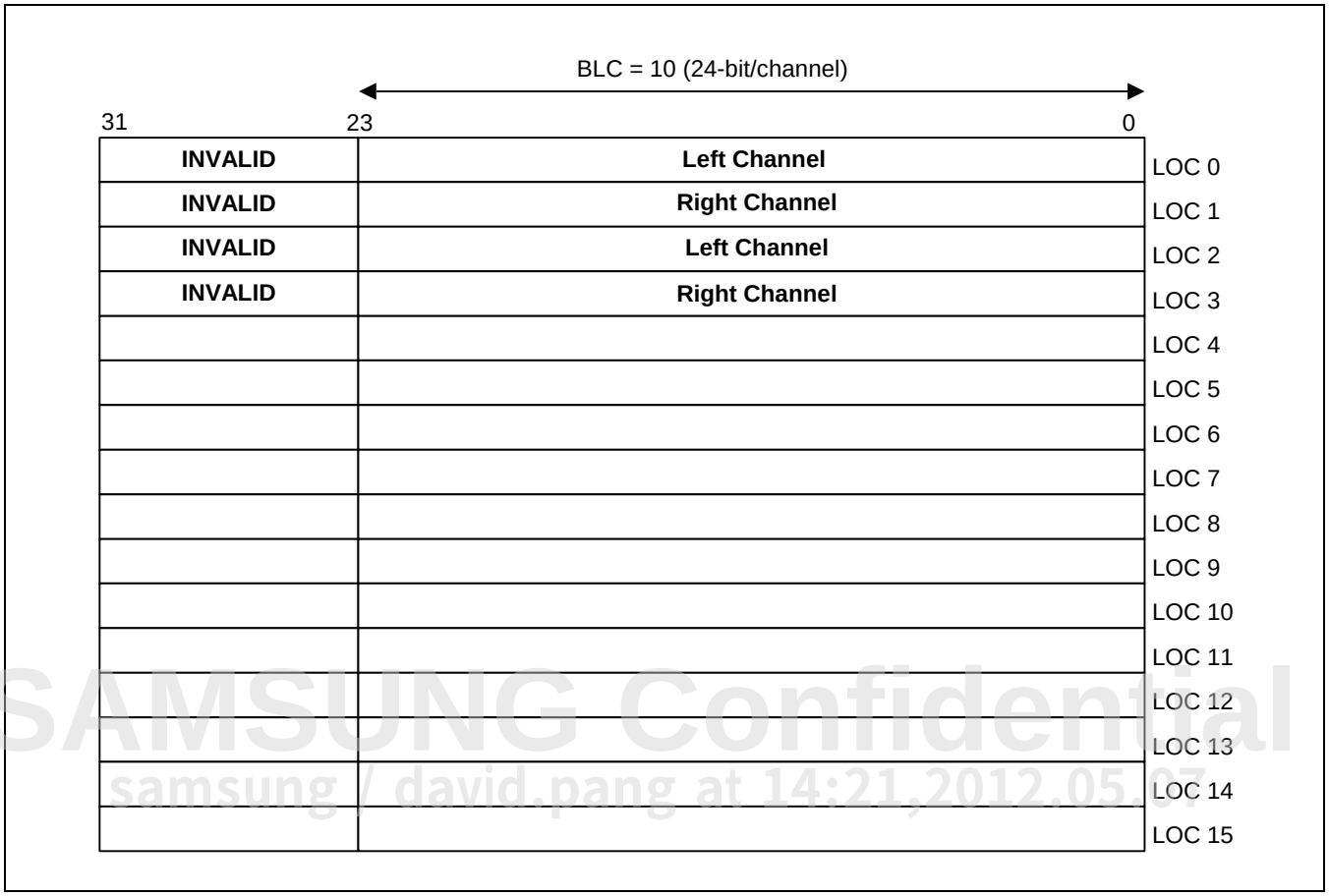


Figure 36-7 Rx FIF0 Structure for BLC = 10 (24-bit/Channel)

The RXCHPAUSE in I2SCON register can stop serial data reception on the I2SSDI.The reception is stopped once the current Left/Right channel is received.

If the control registers in I2SCON Register (I2S Control Register) and I2SMOD Register (I2S Mode Register) then it is advisable to disable the Rx channel.

Verify the status of Rx FIFO by verifying the bits in I2SFIC Register (I2S FIFO Control Register).

36.6 I/O Description

Each I2S (V3.2) external pads are shared with I2S and PCM. To use these pads for I2S, GPIO must be set before starting the I2S. Refer to GPIO chapter of this manual for proper GPIO setting.

Signal	I/O	Description	Pad	Type
I2S1_CDCLK, I2S2_CDCLK	I/O	I2S Codec clock input/output	Xi2s1CDCLK, Xpcm2EXTCLK	Dedicated
I2S1_SCLK, I2S2_SCLK	I/O	I2S Bit Clock input/output	Xi2s1SCLK, Xpcm2SCLK	Dedicated
I2S1_LRCK, I2S2_LRCK	I/O	I2S LR Channel clock input/output	Xi2s1LRCK, Xpcm2FSYNC	Dedicated
I2S1_SDI, I2S2_SDI	I	I2S Serial data input	Xi2s1SDI, Xpcm2SIN	Dedicated
I2S1_SDO, I2S2_SDO	O	I2S Serial data out	Xi2s1SDO, Xpcm2SOUT	Dedicated

36.7 Register Description

36.7.1 Register Map Summary

- Base Address: 0x1396_0000, 0x1397_0000

Register	Offset	Description	Reset Value
IISCON	0x0000	Specifies IIS interface control register	0xE00
IISMOD	0x0004	Specifies IIS interface mode register	0x0
IISFIC	0x0008	Specifies IIS interface FIFO control register	0x0
IISTXD	0x0010	Specifies IIS interface transmit data register	0x0
IISRXD	0x0014	Specifies IIS interface receive data register	0x0

NOTE: All registers of IIS interface are accessible by word unit with STR/LDR instructions.

36.7.1.1 IISCON

- Base Address: 0x1396_0000, 0x1397_0000
- Address = Base Address + 0x0000, Reset Value = 0xE00

Name	Bit	Type	Description	Reset Value
RSVD	[31:18]	–	Reserved (program to zero)	14'b0
FRXOFSTATUS	[17]	RW	Rx FIFO Overflow Interrupt Status. And this is used by interrupt clear bit. When this is high, you can do interrupt clear by writing "1". 0 = Interrupt not occurred. 1 = Interrupt occurred.	1'b0
FRXOFINTEN	[16]	RW	Rx FIFO Overflow Interrupt Enable 0 = Disables RxFIFO Under-run INT 1 = Enables RxFIFO Under-run INT	1'b0
RSVD	[15:12]	–	Reserved (Program to zero)	4'b0
LRI	[11]	R	Left/Right channel clock indication. Note that LRI meaning is dependent on the value of LRP bit of I2SMOD register. 0 = Left (when LRP bit is low) or right (when LRP bit is high) 1 = Right (when LRP bit is low) or left (when LRP bit is high)	1'b1
FTXEMPT	[10]	R	Tx FIFO empty status indication. 0 = FIFO is not empty (ready for transmit data to channel) 1 = FIFO is empty (not ready for transmit data to channel)	1'b1
FRXEMPT	[9]	R	Rx FIFO empty status indication. 0 = FIFO is not empty 1 = FIFO is empty	1'b1
FTXFULL	[8]	R	Tx FIFO full status indication. 0 = FIFO is not full 1 = FIFO is full	1'b0
FRXFULL	[7]	R	Rx FIFO full status indication. 0 = FIFO is not full (ready for receive data from channel) 1 = FIFO is full (not ready for receive data from channel)	1'b0
TXDMAPAUSE	[6]	RW	Tx DMA operation pause command. Note that when this bit is activated at any time, the DMA request halts after current on-going DMA transfer completes. 0 = No pause DMA operation 1 = Pause DMA operation	1'b0
RXDMAPAUSE	[5]	RW	Rx DMA operation pause command. Note that when this bit is activated at any time, the DMA request halts after current on-going DMA transfer completes. 0 = No pause DMA operation	1'b0

Name	Bit	Type	Description	Reset Value
			1 = Pause DMA operation	
TXCHPAUSE	[4]	RW	Tx Channel operation pause command. Note that when this bit is activated at any time, the channel operation halts after left-right channel data transfer completes. 0 = No pause operation 1 = Pause operation	1'b0
RXCHPAUSE	[3]	RW	Rx channel operation pause command. Note that when this bit is activated at any time, the channel operation halts after left-right channel data transfer completes. 0 = No pause operation 1 = Pause operation	1'b0
TXDMACTIVE	[2]	RW	Tx DMA active (start DMA request). Note that when this bit is set from high to low, the DMA operation will be forced to stop immediately. 0 = Inactive 1 = Active	1'b0
RXDMACTIVE	[1]	RW	Rx DMA active (start DMA request). Note that when this bit is set from high to low, the DMA operation will be forced to stop immediately. 0 = Inactive 1 = Active	1'b0
I2SACTIVE	[0]	RW	IIS interface active (start operation). 0 = Inactive 1 = Active	1'b0

36.7.1.2 IISMOD

- Base Address: 0x1396_0000, 0x1397_0000
- Address = Base Address + 0x0004, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:15]	–	Reserved (Program to zero)	17'b0
BLC	[14:13]	RW	Bit Length Control Bit which decides transmission of 8/16 bits per audio channel 00 = 16 bits per channel 01 = 8 bits per channel 10 = 24 bits per channel 11 = Reserved	2'b0
CDCLKCON	[12]	RW	Determine codec clock source 0 = Uses internal codec clock source 1 = Gets codec clock source from external codec chip NOTE: 0 means External CDCLK Input pad enable	1'b0
IMS	[11:10]	RW	IIS master (internal/external) or slave mode select. 00 = Master mode (divide mode, using PCLK) 01 = Master mode (bypass mode, using I2SCLK) 10 = Slave mode (divide mode, using PCLK) 11 = Slave mode (bypass mode, using I2SCLK)	2'b0
TXR	[9:8]	RW	Transmit or receive mode select. 00 = Transmit only mode 01 = Receive only mode 10 = Transmit and receive simultaneous mode 11 = Reserved	2'b0
LRP	[7]	RW	Left/Right channel clock polarity select. 0 = Low for left channel and high for right channel 1 = High for left channel and low for right channel	1'b0
SDF	[6:5]	RW	Serial data format. 00 = IIS format 01 = MSB-justified (left-justified) format 10 = LSB-justified (right-justified) format 11 = Reserved	2'b0
RFS	[4:3]	RW	IIS root clock (codec clock) frequency select. 00 = 256 fs, where fs is sampling frequency 01 = 512 fs 10 = 384 fs 11 = 768 fs	2'b0
BFS	[2:1]	RW	Bit clock frequency select. 00 = 32 fs, where fs is sampling frequency 01 = 48 fs 10 =16 fs 11 = 24 fs	2'b0
RSVD	[0]	–	Reserved (Program to zero)	1'b0

36.7.1.3 IISFIC

- Base Address: 0x1396_0000, 0x1397_0000
- Address = Base Address + 0x0008, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved (Program to zero)	16'b0
TFLUSH	[15]	RW	Tx FIFO Flush command. 0 = No Flush 1 = Flush	1'b0
RSVD	[14:13]	–	Reserved (Program to zero)	2'b0
FTX0CNT	[12:8]	R	Tx FIFO data count. FIFO has 64 dept, so value ranges from 0 to 64. N: Data Count N of FIFO	5'b0
RFLUSH	[7]	RW	Rx FIFO Flush command. 0 = No Flush 1 = Flush	1'b0
RSVD	[6:5]	–	Reserved (Program to zero)	2'b0
FRXCNT	[4:0]	R	Rx FIFO data count. FIFO has 64 dept, so value ranges from 0 to 64. N: Data Count N of FIFO	5'b0

36.7.1.4 IISTXD

- Base Address: 0x1396_0000, 0x1397_0000
- Address = Base Address + 0x0010, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
IISTXD	[31:0]	W	Tx FIFO write data. Note that the left/right channel data is allocated as the following bit fields. R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC	32'b0

36.7.1.5 IISRXD

- Base Address: 0x1396_0000, 0x1397_0000
- Address = Base Address + 0x0014, Reset Value = 0x0

Name	Bit	Type	Description	Reset Value
IISRXD	[31:0]	R	Rx FIFO read data. Note that the left/right channel data is allocated as the following bit fields. R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC	32'b0

37 AC97 Controller

This chapter describes the functions and usage of AC97 Controller in Exynos 4412 SCP RISC microprocessor. It also provides description for the programming model for the AC97 Controller Unit.

37.1 Overview

The AC97 Controller Unit in the Exynos 4412 SCP supports the features of AC97 revision 2.0.

AC97 Controller performs these tasks:

24. Uses audio controller link (AC-link) to communicate with AC97 Codec.
25. Sends the stereo PCM data to Codec. The external digital-to-analog converter (DAC) in the Codec converts the audio sample to an analog audio waveform.
26. Receives the stereo PCM data and the mono MIC data from Codec, and then stores in memories.

NOTE: The prerequisite in this chapter requires an understanding of the AC97 revision 2.0 specifications.

37.2 Features

Features of the AC97 Controller are:

- Independent channels for stereo PCM In, stereo PCM Out, and mono MIC In.
- DMA-based operation and interrupt-based operation.
- All the channels support only 16-bit samples.
- Variable sampling rate of AC97 Codec interface (less than or equal to 48 kHz). (48 kHz and below)
- 16-bit, 16 entry Fist In First Out (FIFO)s per channel
- Only primary Codec support

37.3 AC97 Controller Operation

This section describes the AC97 Controller operation, namely:

- AC-Link
- Power-down sequence
- Wake-up sequence

37.3.1 Block Diagram

Figure 37-1 illustrates the functional block diagram of the Exynos 4412 SCP AC97 Controller.

The AC97 signals from the AC-link are connected with external AC97 Codec device. The Exynos 4412 SCP AC97 Controller supports full-duplex data transfers. All digital audio streams and command/status information are communicated through the AC-link.

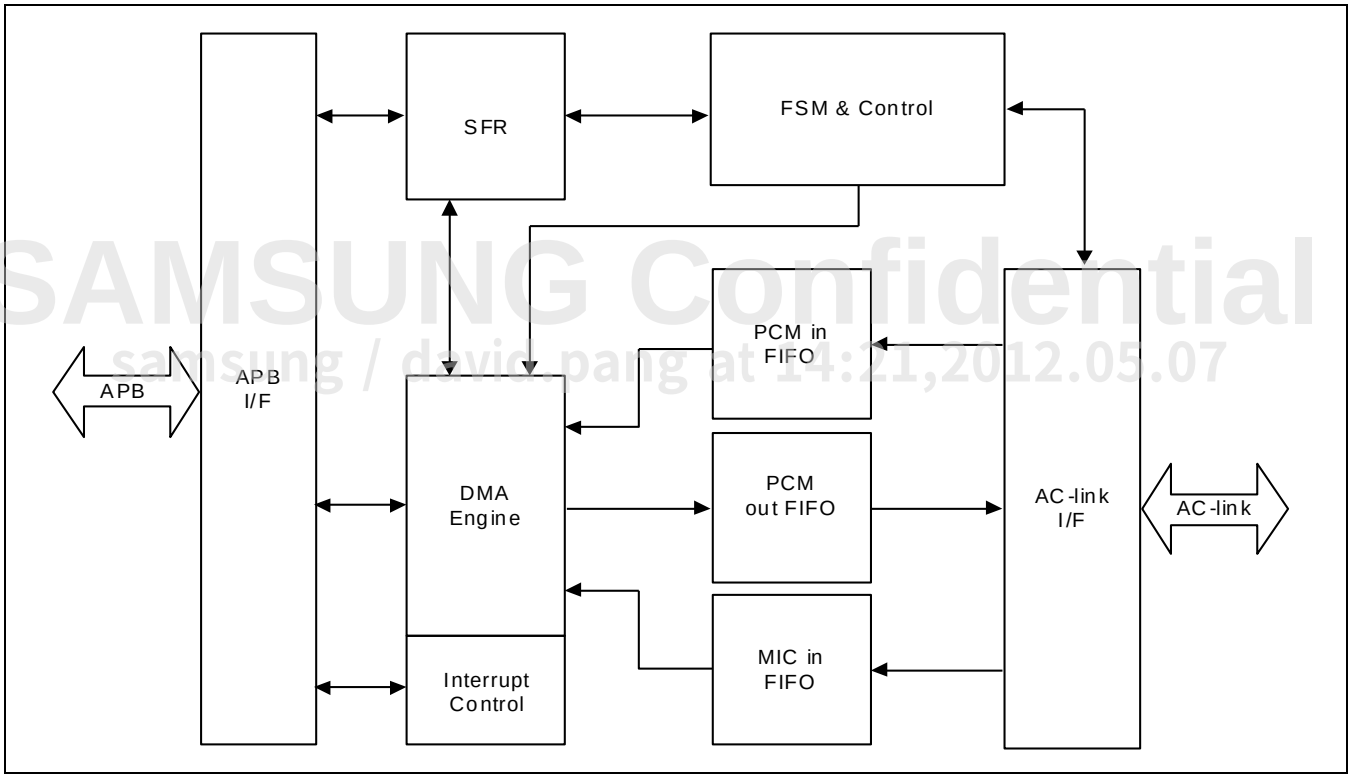


Figure 37-1 AC97 Block Diagram

37.3.2 Internal Data Path

Figure 37-2 illustrates internal data path of the Exynos 4412 SCP AC97 Controller.

Internal data path includes stereo PCM In, stereo PCM Out, and mono MICIn buffers, which include 16-bit and 16 entries buffer. It also has 20-bit I/O shift register through the AC-link.

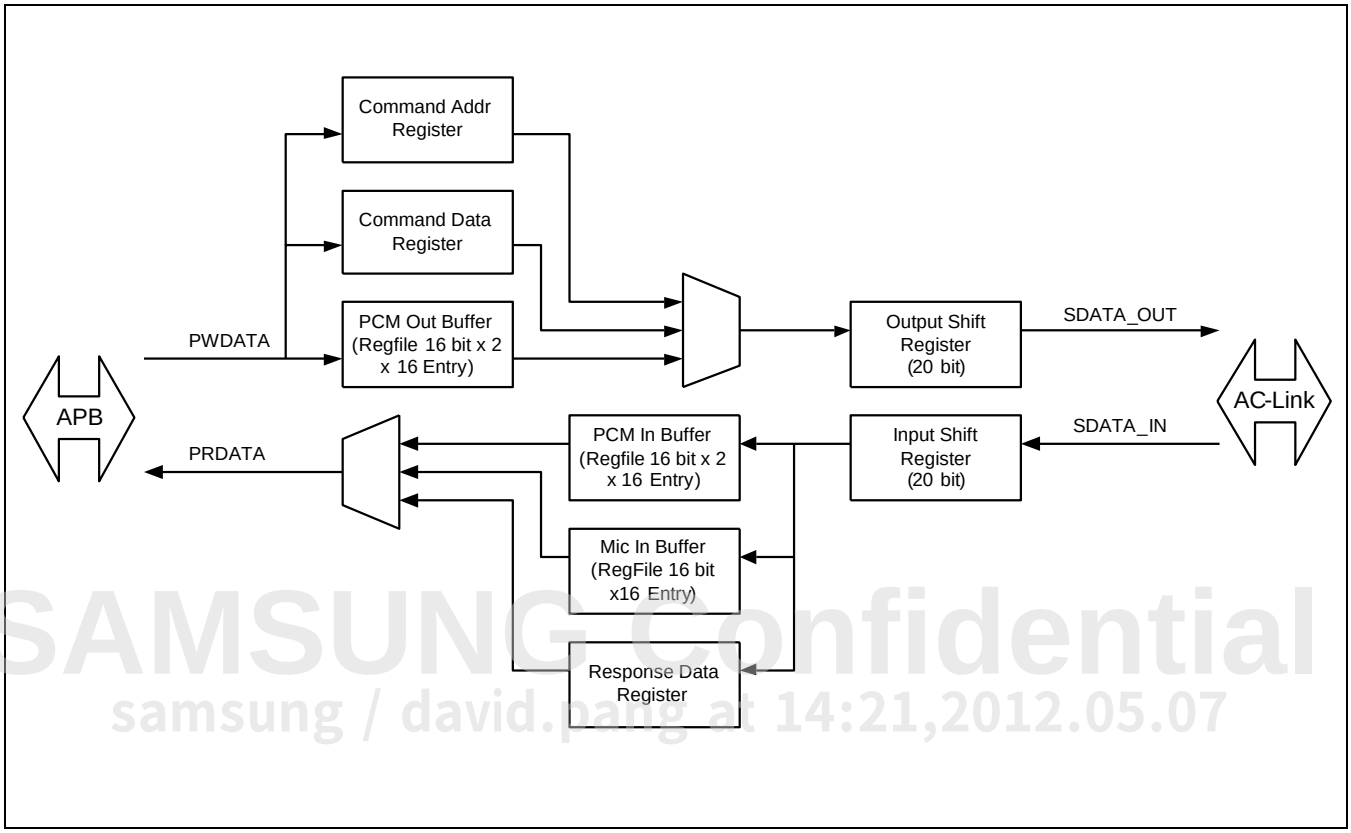


Figure 37-2 Internal Data Path

37.3.3 Operation Flow Chart

When you initialize the AC97 Controller, you must assert system reset or cold reset because the previous state of the external AC97 audio Codec is unknown. This assures that General Purpose Input Output (GPIO) is ready. Then, enable the Codec-ready interrupt. Verify the Codec-ready interrupt by polling or interrupt. When interrupt occurs, you must de-assert the Codec-ready interrupt. Use DMA or PIO (directly to Write data to register) to transmit data either from memory to register or from register to memory. If internal FIFOs (Tx FIFO or Rx FIFO) are not empty, then let data be transmitted.

Figure 37-3 illustrates the AC97 operation flow chart.

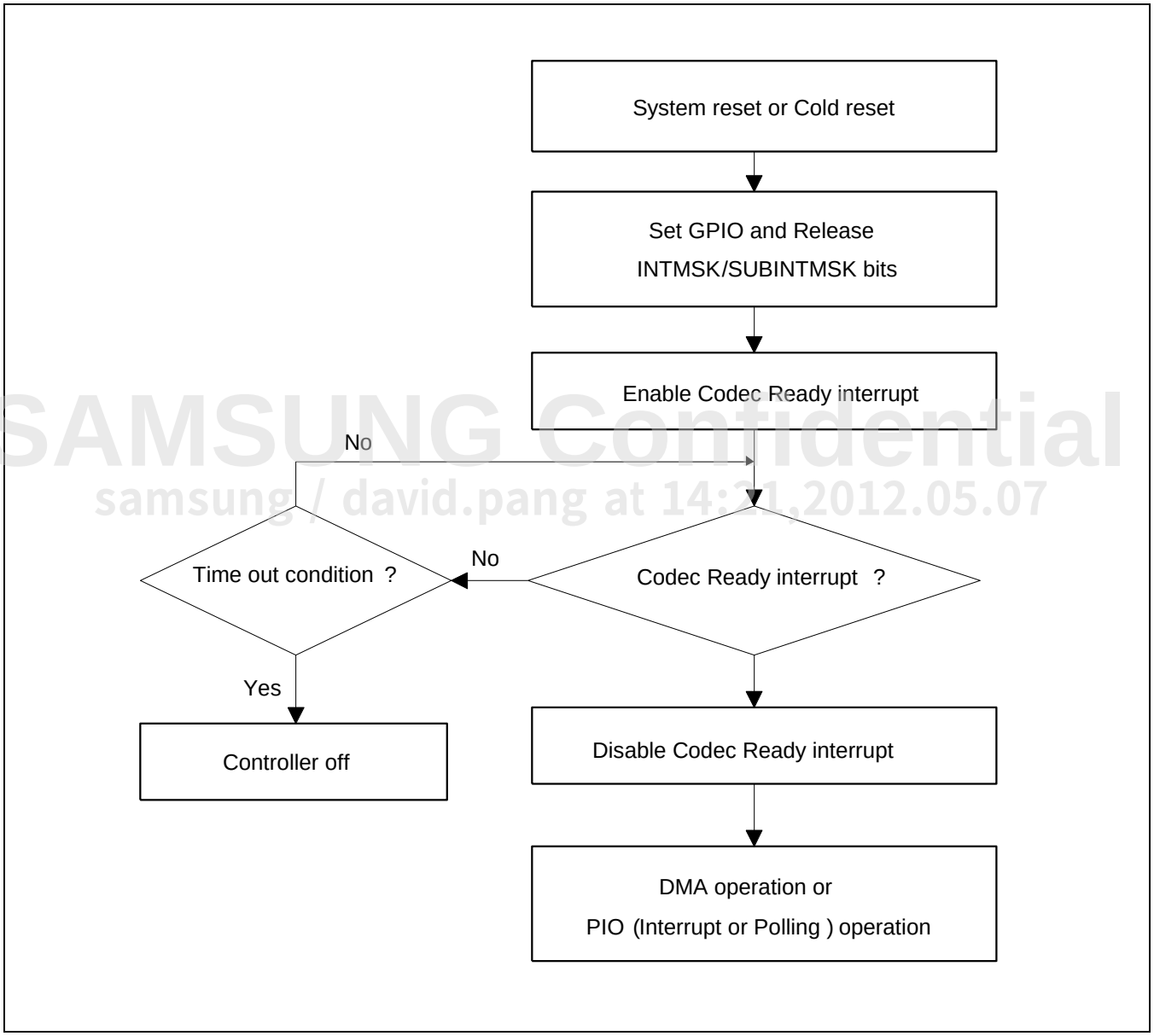


Figure 37-3 AC97 Operation Flow Chart

37.3.4 AC-link Digital Interface Protocol

Each AC97 Codec incorporates a 5-pin digital serial interface that links it to the Exynos 4412 SCP AC97 Controller. AC-link is a full-duplex, fixed-clock, and PCM digital stream. It uses a Time Division Multiplexed (TDM) scheme to manage control register accesses and multiple input and output audio streams.

The AC-link architecture divides each audio frame into 12 outgoing and 12 incoming data streams. Each stream has 20-bit sample resolution and requires a DAC and an ADC with a minimum 16-bit resolution.

Figure 37-4 illustrates the bi-directional AC-link frame with slot assignments.

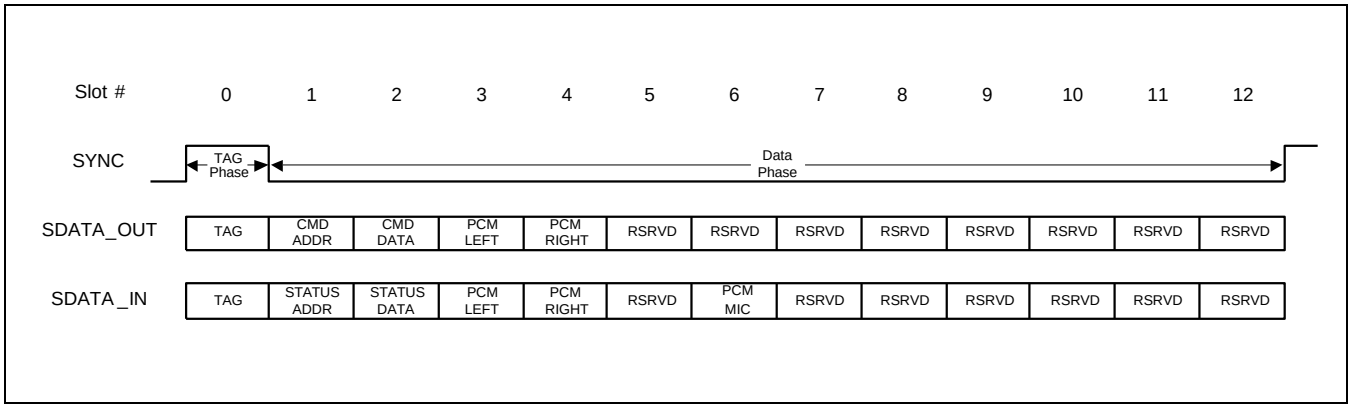


Figure 37-4 Bi-Directional AC-link Frame with Slot Assignments

Figure 37-4 shows the slot definitions that are supported by Exynos 4412 SCP AC97 Controller. The Exynos 4412 SCP AC97 Controller provides synchronization for all data transactions on the AC-link.

A data transaction is made up of 256 bits of information that are broken into groups of 13 time slots. and is called a frame. Time slot 0 is called the Tag Phase and it is 16 bits long. The other 12 time slots are called the Data Phase.

The Tag Phase contains 1-bit that identifies a valid frame and 12 bits that identify the time slots in the Data Phase. that contain valid data. Each time slot in the Data Phase is 20 bits long. A frame begins when SYNC is high. The amount of time that SYNC is high corresponds to the Tag Phase. AC97 frames occur at fixed 48 kHz intervals and are synchronous to the 12.288 MHz bit rate clock, BITCLK.

The Controller and the Codec use the SYNC and BITCLK to determine when to send transmit data and when to sample received data. A transmitter transfers the serial data stream on each rising edge of BITCLK and a receiver samples the serial data stream on falling edges of BITCLK. The transmitter must tag the valid slots in its serial data stream. The valid slots are tagged in slot 0. Serial data on the AC-link is ordered Most Significant Bit (MSB) to Least Significant Bit (LSB).

The first bit of Tag Phase is bit[15] and the first bit of each slot in Data Phase is bit[19]. The last bit in any slot is bit[0].

37.3.4.1 AC-link Output Frame (SDATA_OUT)

Slot 0: Tag Phase

In slot 0, the first bit is a bit (SDATA_OUT, bit[15]) that represents the validity of the entire frame. When bit[15] is 1, the current frame contains at least a valid time slot. The next 12-bit positions correspond each 12 time slot contains valid data. Bits 0 and 1 of slot 0 are used as Codec IO bits for I/O Reads and Writes to the Codec registers. Refer next section for more information. In this way, data streams of different sample rate are transmitted across AC-link at its fixed 48 kHz audio frame rate.

Slot 1: Command Address Port

In slot 1, it communicates control register address and WRITE/READ command information to the AC97 Controller. When software accesses the primary Codec, the hardware configures the frame as follows:

1. In slot 0, the valid bit for 1, 2 slots are set.
2. In slot 1, bit[19] is set (Read) or clear (write). Bits 18-12 (of slot 1) are configured to specify the index to the Codec register. Others are filled with 0's (reserved).
3. In slot 2, it configures with the data, that is, for writing because of output frame.

Slot 2: Command Data Port

In slot 2, this is the Write data with 16-bit resolution ([19:4] is valid data)

Slot 3: PCM Playback Left Channel

Slot 3 is audio output frame. This is the composite digital audio left stream. When a sample has a resolution that is less than 16 bits, the AC97 Controller fills entire training non-valid bit positions in the slot with zeroes.

Slot 4: PCM Playback Right Channel

Slot 4, which is audio output frame, is the composite digital audio right stream. When a sample has a resolution that is less than 16 bits, the AC97 controller fills entire training non-valid bit positions in the slot with zeroes.

Figure 37-5 illustrates the AC-link output frame.

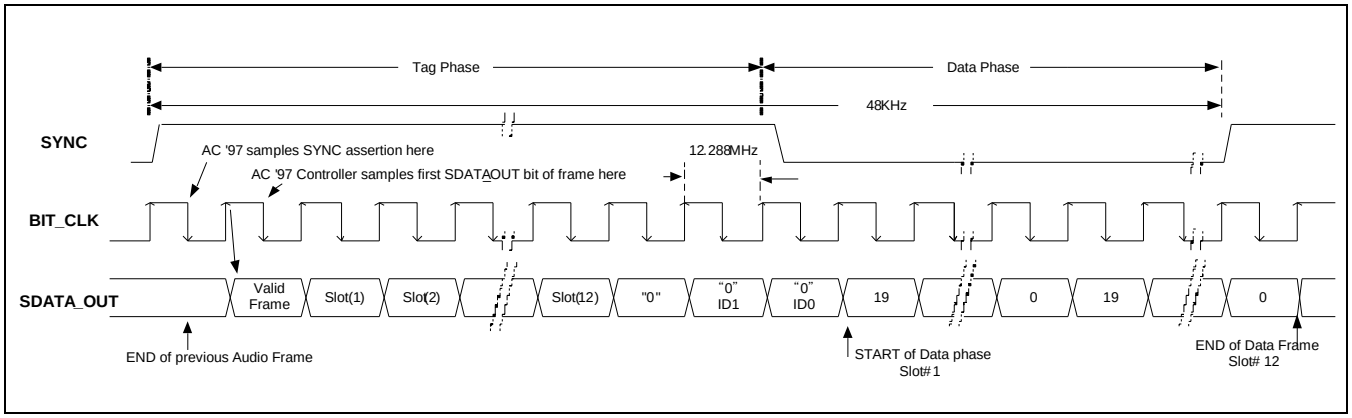


Figure 37-5 AC-Link Output Frame

37.3.5 AC-link Input Frame (SDATA_IN)

Slot 0: Tag Phase

In slot 0, the first bit (SDATA_OUT, bit[15]) indicates whether the AC97 Controller is in the Codec-ready state. If the Codec-Ready bit is set to 0, it means that the AC97 Controller is not ready for normal operation. This condition is normal after it deasserts the power on reset and the AC97 Controller voltage references are settling.

Slot 1: Status Address Port/SLOTREQ Bits

The status port monitors the status of the AC97 Controller functions. It is not limited to mixer settings and power management. Audio input frame slot 1s stream echoes the control register index for the data to be returned in slot 2, if the controller tags slots 1 and 2 as valid during slot 0. The Controller only accepts status data if the accompanying status address matches the last valid command address issued during the most recent READ command. For multiple sample rate output, the Codec examines its sample-rate control registers, its FIFOs' states, and the incoming SDATA_OUT tag bits at the beginning of each audio output frame. This is to determine which SLOTREQ bits to set active (low).

SLOTREQ bits asserted during the current audio input frame indicate which output slots require data from the Controller in the next audio output frame. For fixed 48 kHz operation, the SLOTREQ bits are set active (low), and transfers a sample in each frame. For multiple sample-rate input, the "tag" bit for each input slot indicates whether valid data is present.

Table 37-1 describes the input slot 1-bit definitions.

Table 37-1 Input Slot 1-bit Definitions

Bit	Description
[19]	Reserved (Filled with zero)
[18:12]	Control Register index (Filled with zeroes if AC97 tags is invalid)
[11]	Slot 3 Request: PCM left channel
[10]	Slot 4 Request: PCM right channel
[9]	Slot 5 Request: NA
[8]	Slot 6 Request: MIC channel
[7]	Slot 7 Request: NA
[6]	Slot 8 Request: NA
[5]	Slot 9 Request: NA
[4]	Slot 10 Request: NA
[3]	Slot 11 Request: NA
[2]	Slot 12 Request: NA
[1:0]	Reserved (Filled with zero)

Slot 2: Status Data Port

In slot 2, this is the status data with 16-bit resolution ([19:4] is valid data)

Slot 3: PCM Record Left Channel

Slot 3, which is audio input frame, is the left channel audio output of the AC97 Codec. When a sample has a resolution that is less than 16 bits, the AC97 Codec fills all training non-valid bit positions in the slot with zeroes.

Slot 4: PCM Record Right Channel

Slot 4, which is audio input frame, is the right channel audio output of the AC97 Codec. When a sample has a resolution that is less than 16 bits, the AC97 Codec fills all training non-valid bit positions in the slot with zeroes.

Slot 6: Microphone Record Data

The AC97 Controller supports 16-bit resolution for the MIC In channel.

Figure 37-6 illustrates the AC-link input frame.

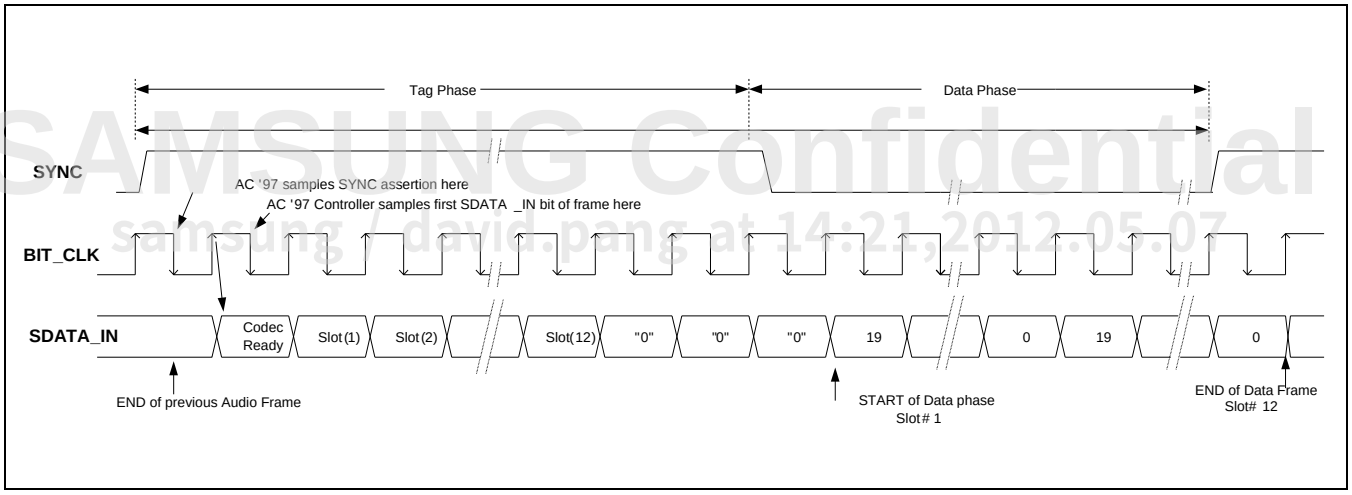


Figure 37-6 AC-Link Input Frame

37.3.6 AC97 Power-Down

Figure 37-7 illustrates the AC97 power-down timing.

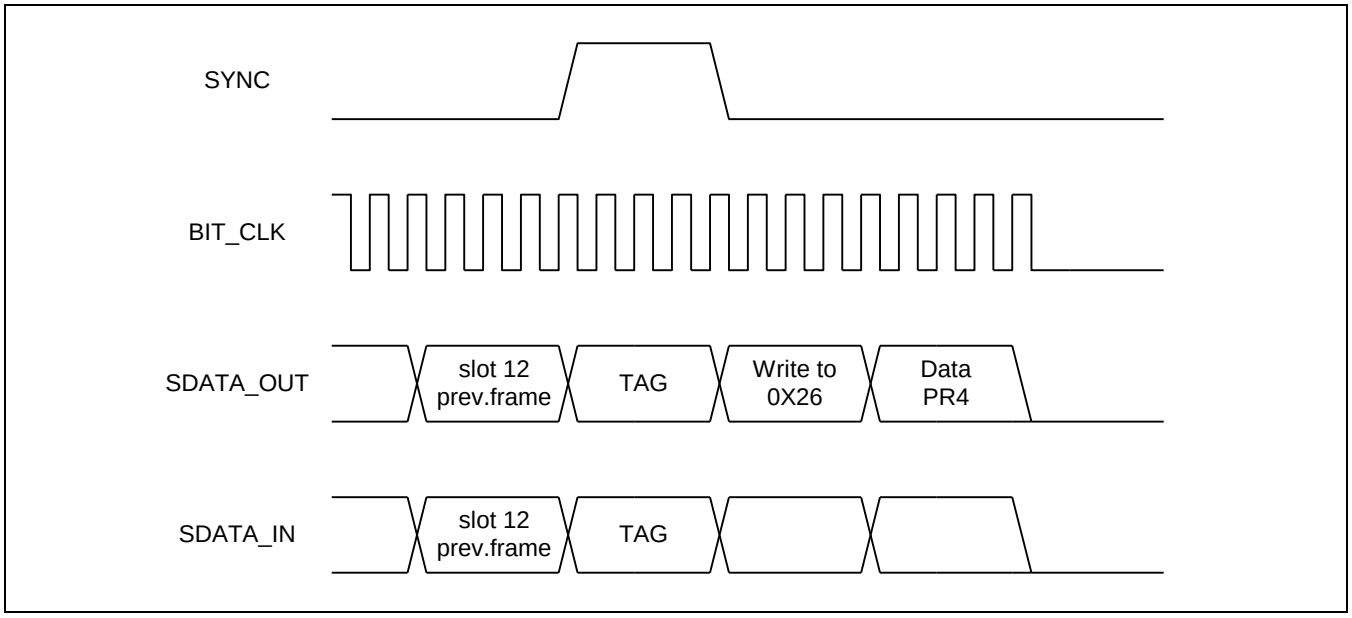


Figure 37-7 AC97 Power-Down Timing

37.3.6.1 Powering Down the AC-Link

The AC-link signals enter a low-power mode when the AC97 Codec power-down register (0x26) bit PR4 is set to 1 (by writing 0x1000). The primary Codec then drives both BITCLK and SDATA_IN to a logic low-voltage level. The sequence follows the timing diagram as shown in *Figure 37-7*.

The AC97 Controller transmits the Write to power-down register (0x26) through AC-link. Set up the AC97 Controller so that it does not transmit data to slots 3-12 when it Writes to the Power-down register bit PR4 (data 0x1000). It does not require the Codec to process other data when it receives a power-down request. When Codec processes the request, it immediately transfers BITCLK and SDATA_IN to a logic low level.

The AC97 Controller drives SYNC and SDATA_OUT to a logic low level after programming the AC_GLBCTRL register.

37.3.6.2 Waking Up the AC-Link-Wake Up Triggered by the AC97 Controller

AC-link protocol provides a cold AC97 reset and a warm AC97 reset. The current power-down state ultimately dictates which AC97 reset is used. Registers must stay in the same state during all power-down modes unless a cold AC97 reset is performed. In a cold AC97 reset, the AC97 registers are initialized to their default values. After a power down, the AC-link must wait for a minimum of four audio frame times after the frame in which the power down occurred. before it can be reactivated by reasserting the SYNC signal. When AC-link powers up, Codec ready bit (input slot 0, bit[15]) indicates that AC-link is ready for operation.

Figure 37-8 illustrates the AC97 power down/power up flow.

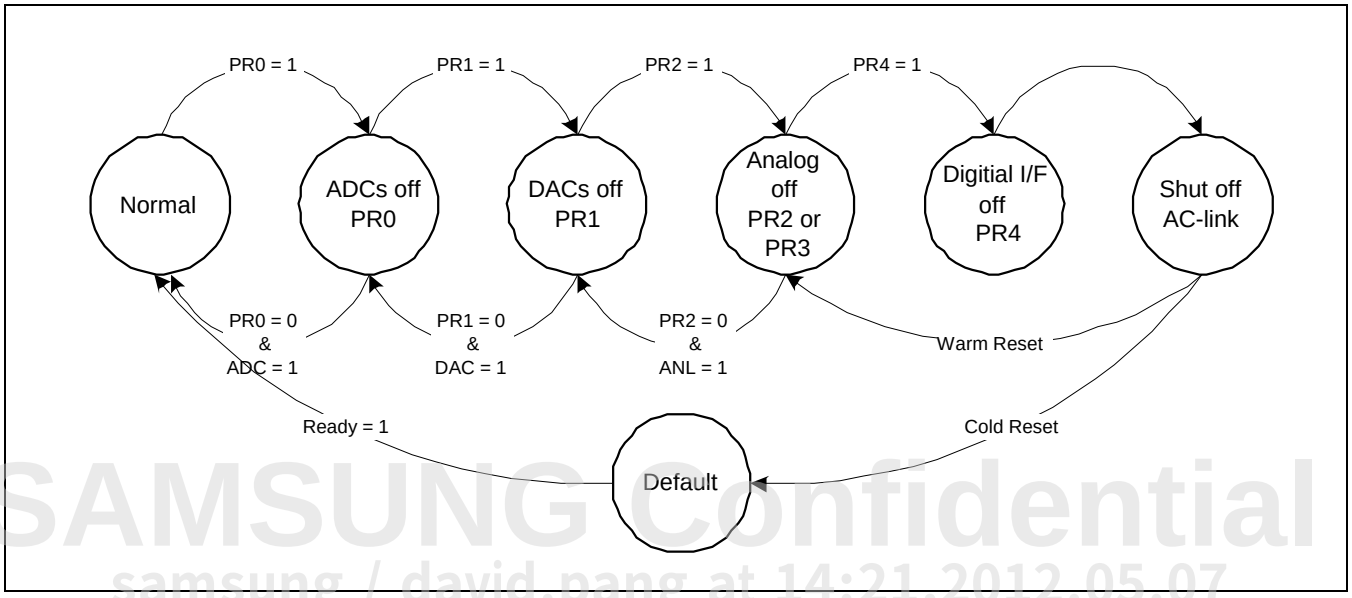


Figure 37-8 AC97 Power Down/Power Up Flow

37.3.6.3 Cold AC97 Reset

A cold reset is generated when the nRESET pin is asserted through the AC_GLBCTRL. Asserting and deasserting nRESET activates BITCLK and SDATA_OUT. A cold reset initializes all of AC97 control registers. nRESET is an asynchronous AC97 input.

37.3.6.4 Warm AC97 Reset

A warm AC97 reset reactivates the AC-link without altering the current AC97 register values. A warm reset is generated when BITCLK is absent and SYNC is driven high. In normal audio frames, SYNC is a synchronous AC97 input. When BITCLK is absent, SYNC is treated as an asynchronous input that is used to generate a warm reset to AC97.The AC97 Controller must not activate BITCLK until it samples SYNC low again. This prevents a new audio frame being falsely detected.

37.3.6.5 AC97 State Diagram

Figure 37-9 illustrates the AC97 state diagram.

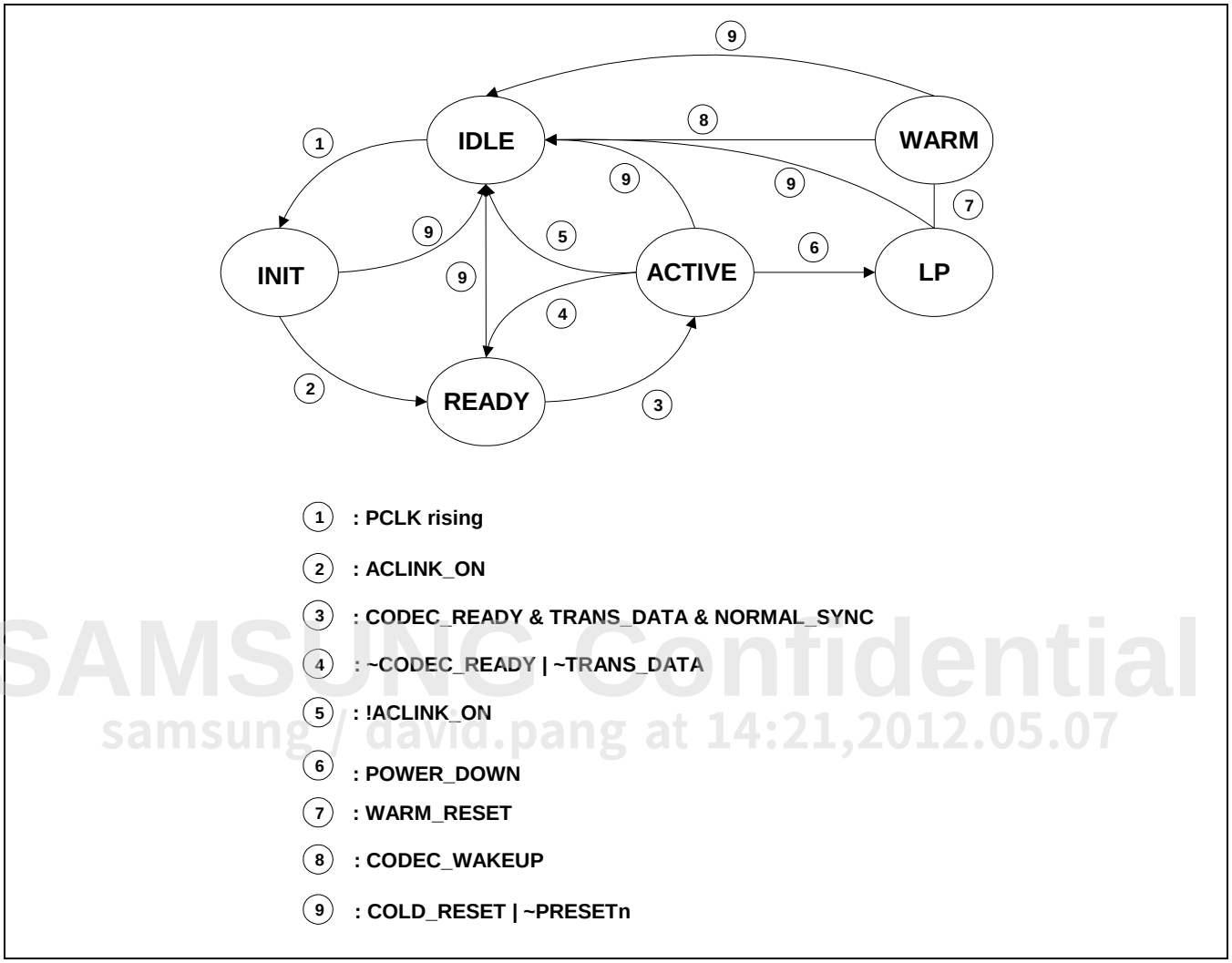


Figure 37-9 AC97 State Diagram

Figure 37-9 shows the state diagram of AC97 Controller. It is useful to verify AC97 Controller state machine. It shows that peripheral clock (PCLK) synchronizes state machine. Use AC_GLBSTAT register to monitor state.

37.4 I/O Description

AC97 external pads are shared with I2S. To use these pads for AC97, you must set GPIO before the AC97 starts. Refer to the GPIO chapter of this manual for more information.

Signal	I/O	Description	Pad	Type
AC97RESETn	O	Active-low Codec reset	Xi2s1CDCLK	muxed
AC97BITCLK	I	12.288 MHz bit-rate clock	Xi2s1SCLK	muxed
AC97SYNC	O	48 kHz frame indicator and synchronizer	Xi2s1LRCK	muxed
AC97SDI	O	Serial audio output data	Xi2s1SDO	muxed
AC97SDO	I	Serial audio input data	Xi2s1SDI	muxed

37.5 Register Description

37.5.1 Register Map Summary

- Base Address: 0x139A_0000

Register	Offset	Description	Reset Value
AC_GLBCTRL	0x0000	Specifies the AC97 global control register	0x0000_0000
AC_GLBSTAT	0x0004	Specifies the AC97 global status register	0x0000_0001
AC_CODEC_CMD	0x0008	Specifies the AC97 codec command register	0x0000_0000
AC_CODEC_STAT	0x000C	Specifies the AC97 codec status register	0x0000_0000
AC_PCMADDR	0x0010	Specifies the AC97 PCM out/in channel FIFO address register	0x0000_0000
AC_MICADDR	0x0014	Specifies the AC97 MIC in channel FIFO address register	0x0000_0000
AC_PCMDATA	0x0018	Specifies the AC97 PCM out/in channel FIFO data register	0x0000_0000
AC_MICDATA	0x001C	Specifies the AC97 MIC in channel FIFO data register	0x0000_0000

37.5.1.1 AC_GLBCTRL

- Base Address: 0x139A_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
Codec ready interrupt clear	[30]	RW	1 = Interrupt clear (Write only)	0
PCM out channel underrun interrupt clear	[29]	RW	1 = Interrupt clear (Write only)	0
PCM in channel overrun interrupt clear	[28]	RW	1 = Interrupt clear (Write only)	0
MIC in channel overrun interrupt clear	[27]	RW	1 = Interrupt clear (Write only)	0
PCM out channel threshold interrupt clear	[26]	RW	1 = Interrupt clear (Write only)	0
PCM in channel threshold interrupt clear	[25]	RW	1 = Interrupt clear (Write only)	0
MIC in channel threshold interrupt clear	[24]	RW	1 = Interrupt clear (Write only)	0
RSVD	[23]	–	Reserved	0
Codec ready interrupt enable	[22]	RW	0 = Disables 1 = Enables	0
PCM out channel underrun interrupt enable	[21]	RW	0 = Disables 1 = Enables (FIFO is empty)	0
PCM in channel overrun interrupt enable	[20]	RW	0 = Disables 1 = Enables (FIFO is full)	0
Mic in channel overrun interrupt enable	[19]	RW	0 = Disables 1 = Enables (FIFO is full)	0
PCM out channel threshold interrupt enable	[18]	RW	0 = Disables 1 = Enables (FIFO is half empty)	0
PCM in channel threshold interrupt enable	[17]	RW	0 = Disables 1 = Enables (FIFO is half full)	0
MIC in channel threshold interrupt enable	[16]	RW	0 = Disables 1 = Enables (FIFO is half full)	0
RSVD	[15:14]	–	Reserved	00
PCM out channel transfer mode	[13:12]	RW	00 = Off 01 = PIO 10 = DMA 11 = Reserved	00
PCM in channel transfer mode	[11:10]	RW	00 = Off 01 = PIO 10 = DMA 11 = Reserved	00
MIC in channel transfer	[9:8]	RW	00 = Off	00

Name	Bit	Type	Description	Reset Value
mode			01 = PIO 10 = DMA 11 = Reserved	
RSVD	[7:4]	–	Reserved	0000
Transfer data enable using AC-link	[3]	RW	0 = Disables 1 = Enables	0
AC-Link on	[2]	RW	0 = Off 1 = SYNC signal transfer to Codec	0
Warm reset	[1]	RW	0 = Normal 1 = Wake up Codec from power down	0
Cold reset	[0]	S	0 = Normal 1 = Resets Codec and Controller logic NOTE: 1. During Cold reset, does not affect Writing to any AC97 registers 2. When recovering from Cold reset, does not affect writing to any AC97 registers Example: For consecutive Cold reset and Warm reset, first set AC_GLBCTRL = 0x1 then set AC_GLBCTRL = 0x0. After recovering from Cold reset set AC_GLBCTRL = 0x2 then AC_GLBCTRL = 0x0.	0

NOTE: This is the global register of the AC97 Controller. There are Interrupt Control registers, DMA Control registers, AC-Link Control register, Data Transmission Control register, and Related Reset Control register.

37.5.1.2 AC_GLBSTAT

- Base Address: 0x139A_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:23]	–	Reserved	0x00
Codec ready interrupt	[22]	R	0 = Not requested 1 = Requested	0
PCM out channel underrun interrupt	[21]	R	0 = Not requested 1 = Requested	0
PCM in channel overrun interrupt	[20]	R	0 = Not requested 1 = Requested	0
MIC in channel overrun interrupt	[19]	R	0 = Not requested 1 = Requested	0
PCM out channel threshold interrupt	[18]	R	0 = Not requested 1 = Requested	0
PCM in channel threshold interrupt	[17]	R	0 = Not requested 1 = Requested	0
MIC in channel threshold interrupt	[16]	R	0 = Not requested 1 = Requested	0
RSVD	[15:3]	–	Reserved	0x000
Controller main state	[2:0]	R	000 = Idle 001 = Init 010 = Ready 011 = Active 100 = LP 101 = Warm	001

NOTE: This is the status register. When the interrupt occurs, you can verify the source of interrupt.

37.5.1.3 AC_CODEC_CMD

- Base Address: 0x139A_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x00
Read enable	[23]	RW	0 = Command Write (NOTE) 1 = Status Read	0
Address	[22:16]	RW	Codec command address	0x00
Data	[15:0]	RW	Codec command data	0x0000

When you control Writing or Reading, you must set the Read-enable bit. If you want to Write data to the AC97 Codec, you must set the index (or address) of the AC97 Codec and data.

NOTE: When the commands are written on the AC_CODEC_CMD register, consider to have more than 1/48 kHz delay time between the command and the next command.

37.5.1.4 AC_CODEC_STAT

- Base Address: 0x139A_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:23]	–	Reserved	0x00
Address	[22:16]	R	Codec status address	0x00
Data	[15:0]	R	Codec status data	0x0000

NOTE:

1. When the Read-enable bit is 1 and Codec command address is valid, Codec status data is also valid.
2. Steps to Read data from AC97 Codec register through the AC_CODEC_STAT register:
   - Write command address and data on the AC_CODEC_CMD register with Bit[23] = 1.
   - Set a proper delay time. Proper delay time depends on Codec type.
   - Read command address and data from AC_CODEC_STAT register.

37.5.1.5 AC_PCMADDR

- Base Address: 0x139A_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0000
Out read address	[27:24]	R	PCM out channel FIFO read address	0000
RSVD	[23:20]	–	Reserved	0000
In read address	[19:16]	R	PCM in channel FIFO read address	0000
RSVD	[15:12]	–	Reserved	0000
Out write address	[11:8]	R	PCM out channel FIFO write address	0000
RSVD	[7:4]	–	Reserved	0000
In write address	[3:0]	R	PCM in channel FIFO write address	0000

To index the internal PCM FIFOs address.

37.5.1.6 AC_MICADDR

- Base Address: 0x139A_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	0000
Read address	[19:16]	R	MIC In channel FIFO read address	0000
RSVD	[15:4]	–	Reserved	0x000
Write address	[3:0]	R	MIC In channel FIFO write address	0000

To index the internal MIC-in FIFO address.

37.5.1.7 AC_PCMDATA

- Base Address: 0x139A_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Right data	[31:16]	RW	PCM out/in right channel FIFO data Read = PCM In right channel Write = PCM Out right channel	0x0000
Left data	[15:0]	RW	PCM out/in left channel FIFO data Read = PCM In left channel Write = PCM Out left channel	0x0000

NOTE: This is PCM out/in channel FIFO data register.

37.5.1.8 AC_MICDATA

- Base Address: 0x139A_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0x0000
Mono data	[15:0]	RW	MIC In mono channel FIFO data	0x0000

NOTE: This is MIC In channel FIFO data register.

38 PCM Audio Interface

38.1 Overview

PCM Audio Interface module provides PCM bi-directional serial interface to an external Codec.

38.2 Features

Features of PCM Audio Interface are:

- 16-bit PCM, and three ports audio interface
- Supports only master mode
- All PCM serial timings and strobes including the main shift clock, are based on an PCM_EXTCLK
- OSC, EPLL_FOUT, or AUDIO_SCLK can be used as PCM_EXTCLK source clock
- Optional timing based on the internal APB PCLK
- Input (16-bit × 32 depth) and output (16-bit × 32 depth) FIFOs to buffer data
- Optional DMA interface for Tx, or Rx, or for both.

38.3 PCM Audio Interface

The PCM Audio Interface provides a serial interface to an external Codec. PCM module receives an input PCMCODEC_CLK to generate the serial shift timing. The PCM interface outputs a serial data out, a serial shift clock, and a sync signal. It receives data from the external Codec over a serial input line. It synchronizes serial data in, serial data out, and sync signal to the serial shift clock.

The serial shift clock, PCMSCLK, is generated from a programmable divide of the input PCMCODEC_CLK. The sync signal, PCMSYNC, is generated based on a programmable number of serial clocks and is one serial clock wide.

PCM data words are 16-bit wide and it serially shifts out 1-bit per PCMSCLK. Only one 16-bit word is shifted out for each PCMSYNC. The PCMSCLK will continue to toggle even after all 16-bit have been shifted out. The PCMSOUT data is not valid after the 16-bit word is complete. The next PCMSYNC will signal the start of the next PCM data word.

The Tx FIFO provides the 16-bit data word to be serially shifted out. This data is serially shifted out MSB first, one bit per PCMSCLK. The PCM serial output data, PCMSOUT, is clocked out using the rising edge of the PCMSCLK. The MSB bit position relative to the PCMSYNC is programmable to either match the PCMSYNC or one PCMCLK later. After all 16-bit have been shifted out, to indicate the end of the transfer you can generate an interrupt.

Simultaneously when the data is being shifted out, it uses PCMSIN input to serially shift data from the external codec. The data is received MSB first and is clocked in on the falling edge of PCMSCLK. The position of the first bit is programmable to be coincident with the PCMSYNC or one PCMSCLK later.

The first 16-bit are serially shifted into the PCM_DATAIN register which is then loaded into the Rx FIFO. It ignores subsequent bits until the next PCMSYNC.

Various Interrupts are available to indicate the status of the Rx and Tx FIFO. Each FIFO has a programmable flag to indicate when the CPU should to service the FIFO. In the Rx FIFO, there is an interrupt, which will be raised when the FIFO exceeds a certain programmable ALMOST_FULL depth. Similarly there is a programmable ALMOST_EMPTY interrupt for the Tx FIFO.

38.4 PCM Timing

Figure 38-1 illustrates the timing relationship for the PCM transfers.

In all cases, the PCM shift timing is derived by dividing the input clock, PCMCODEC_CLK. While the timing is based on the PCMCODEC_CLK, there is no attempt to realign the rising edge of the output PCMSCLK with the original PCMCODEC_CLK input clock. It will skew these edges by internal delay through the pads as well as the divider logic. This does not represent a problem because it synchronizes the actual shift clock, PCMSCLK, with the data. Furthermore, even if you do not use the PCMSCLK output, the skew will be significantly less than the period of the PCMCODEC_CLK and this does not represent a problem since most PCM interfaces capture data on the falling edge of the clock.

Figure 38-1 illustrates a PCM transfer with the MSB configured to be coincident with the PCMSYNC. This MSB positioning corresponds to setting the TX_MSB_POS and RX_MSB_POS bits in PCMCTL register to be 0.

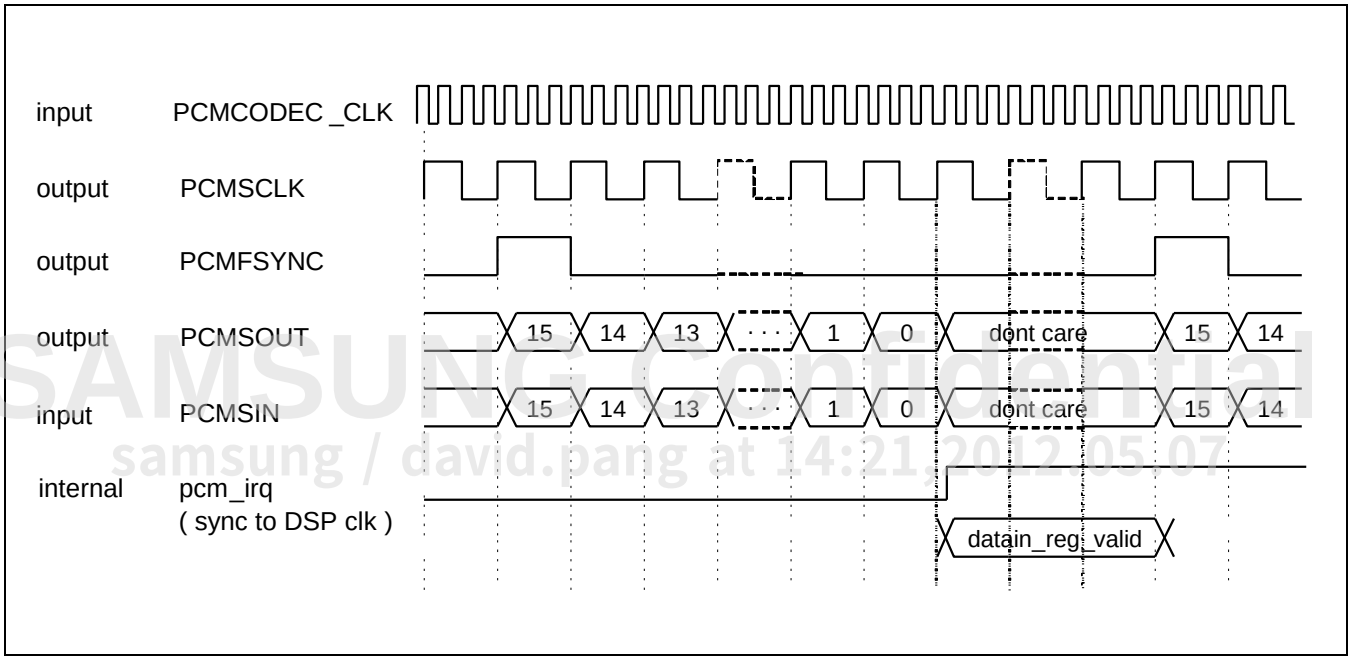


Figure 38-1 PCM Timing, POS_MSB_WR/RD = 0

Figure 38-2 illustrates a PCM transfer with the MSB configured one shift clock after the PCMSYNC. This MSB positioning corresponds to setting the TX_MSB_POS and RX_MSB_POS bits in PCMCTL register to be 1.

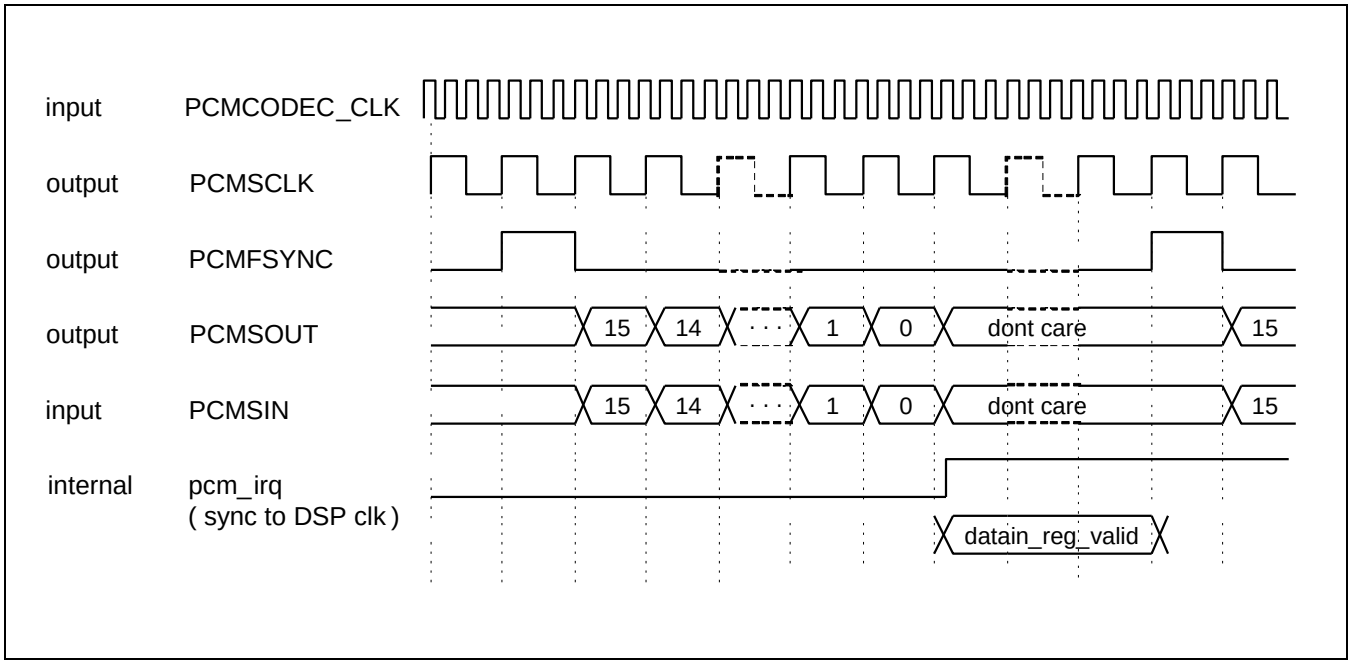


Figure 38-2 PCM timing, POS_MSB_WR/RD = 1

Figure 38-3 illustrates the input clock diagram for PCM.

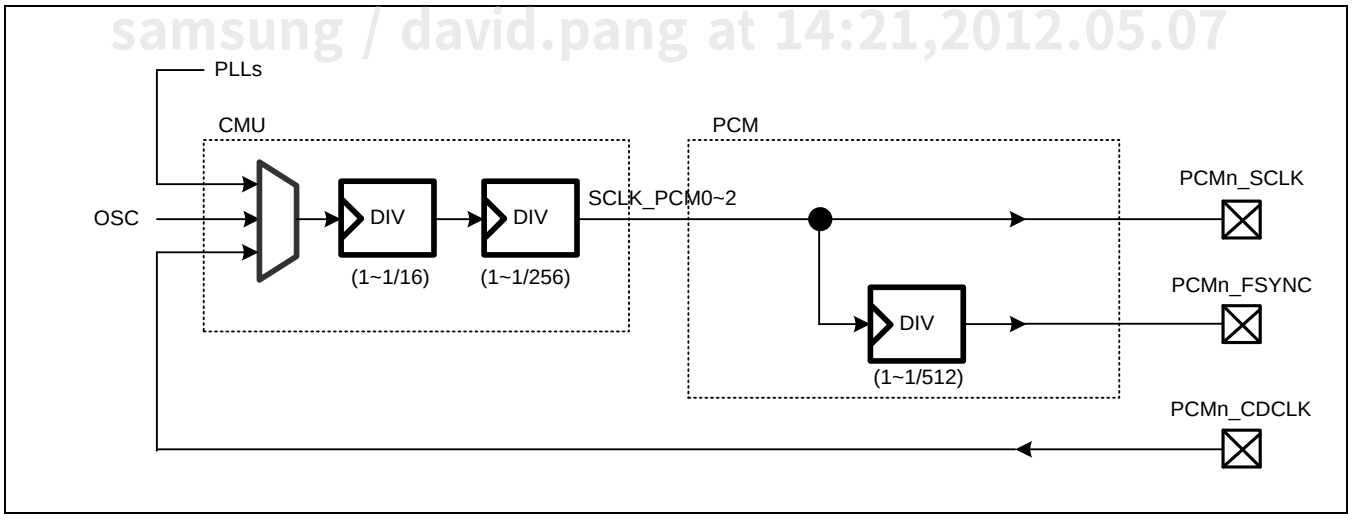


Figure 38-3 Input Clock Diagram for PCM

Exynos 4412 SCP PCM can use only SCLK_PCM from the SYSCON part. SCLK_PCMs generated by dividing PLL, OSC or External Clock. Refer to Figure 38-3 for more information. Refer to CMU chapter, for more information on enabling clock gating.

38.5 I/O Description

Each PCM external pads are shared with I2S, AC97, SPDIF, and PCM. To use these pads for PCM, GPIO must be set before starting the PCM. Refer to GPIO chapter of this manual for more information on proper GPIO setting.

Signal	I/O	Description	Pad	Type
PCM0_EXTCLK, PCM1_EXTCLK, PCM2_EXTCLK	I/O	PCM Codec clock input/output	Xpcm0EXTCLK, Xi2s0CDCLK, Xi2s1CDCLK,	Dedicated
PCM0_SCLK, PCM1_SCLK, PCM2_SCLK	I/O	PCM Bit clock input/output	Xpcm0SCLK, Xi2s0SCLK, Xi2s1SCLK	Dedicated
PCM0_FSYNC, PCM1_FSYNC, PCM2_FSYNC	I/O	PCM FSYNC channel clock input/output	Xpcm0FSYNC, Xi2s0LRCK, Xi2s1LRCK	Dedicated
PCM0_SIN, PCM1_SIN, PCM2_SIN	I	PCM serial data input	Xpcm0SIN, Xi2s0SDI, Xi2s1SDI	Dedicated
PCM0_SOUT, PCM1_SOUT, PCM2_SOUT	O	PCM serial data out	Xpcm0SOUT, Xi2s0SDO, Xi2s1SDO	Dedicated

38.6 Register Description

38.6.1 Register Map Summary

- Base Address: 0x1398_0000, 0x1399_0000

Register	Offset	Description	Reset Value
PCM_CTL	0x0000	Specifies the PCM main control	0x0000_0000
PCM_CLKCTL	0x0004	Specifies the PCM clock and shift control	0x0000_0000
PCM_TXFIFO	0x0008	Specifies the PCM TxFIFO write port	0x0001_0000
PCM_RXFIFO	0x000C	Specifies the PCM RxFIFO read port	0x0001_0000
PCM_IRQ_CTL	0x0010	Specifies the PCM interrupt control	0x0000_0000
PCM_IRQ_STAT	0x0014	Specifies the PCM interrupt status	0x0000_0000
PCM_FIFO_STAT	0x0018	Specifies the PCM FIFO status	0x0000_0000
PCM_CLRINT	0x0020	Specifies the PCM interrupt clear	Undefined

38.6.1.1 PCM_CTL

- Base Address: 0x1398_0000, 0x1399_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:19]	–	Reserved	–
TXFIFO_DIPSTICK	[18:13]	RW	Determines when the ALMOST_FULL and ALMOST_EMPTY flags go active for the TxFIFO. • ALMOST_EMPTY: fifo_depth < fifo_dipstick • ALMOST_FULL: fifo_depth > (32 – fifo_dipstick) NOTE: 1. If fifo_dipstick == 0 ALMOST_EMPTY and ALMOST_FULL are invalid 2. For DMA loading of Tx FIFO, Txfifo_dipstick >= 2 It requires this since the PCM_TXDMA uses ALMOST_FULL as the DMA request (keep requesting data until the FIFO is almost full), In some circumstances, the DMA writes one more word after the DMA_req fall to low. Thus, the ALMOST_FULL flag go active with a minimum space for one extra word in the FIFO.	0
RXFIFO_DIPSTICK	[12:7]	RW	Determines when the ALMOST_FULL and ALMOST_EMPTY flags go active for the RxFIFO. • ALMOST_EMPTY: fifo_depth < fifo_dipstick • ALMOST_FULL: fifo_depth > (32 – fifo_dipstick) NOTE: 1. If fifo_dipstick == 0 ALMOST_EMPTY and ALMOST_FULL are invalid 2. For DMA, RXFIFO_DIPSTICK is a don't care DMA unloading of RX fifo uses the rx_fifo_empty flag as the DMA request 3. Non-DMA IRQ/polling RXFIFO_DIPSTICK should be 0x20 This will have the effect of rx_fifo_ALMOST_FULL acting as an rx_fifo_not_empty flag.	0
PCM_TX_DMA_EN	[6]	RW	Enables the DMA interface for the TxFIFO. DMA_TX request will occur whenever the TxFIFO is not almost full.	0
PCM_RX_DMA_EN	[5]	RW	Enables the DMA interface for the RxFIFO. DMA_RX request will occur whenever the RxFIFO is not empty.	0
TX_MSB_POS	[4]	RW	Controls the position of the MSB bit in the serial output stream relative to the PCMSYNC signal. 0 = MSB sent during the same clock that PCMSYNC is high	0

Name	Bit	Type	Description	Reset Value
			1 = MSB sent on the next PCMSCLK cycle after PCMSYNC is high.	
RX_MSB _POS	[3]	RW	Controls the position of the MSB bit in the serial input stream relative to the PCMSYNC signal. 0 = MSB is captured on the falling edge of PCMSCLK during the same cycle that PCMSYNC is high 1 = MSB is captured on the falling edge of PCMSCLK during the cycle after the PCMSYNC is high	0
PCM_TXFIFO _EN	[2]	RW	Enables the TxFIFO When the enable is low, the internal FIFOs will clear and reinitialize.	0
PCM_RXFIFO _EN	[1]	RW	Enables the RxFIFO When the enable is low, the internal FIFOs will clear and reinitialize.	0
PCM_PCM _ENABLE	[0]	RW	PCM enable signal. Enables the serial shift state machines. The enable must be set high for the PCM to operate. When the enable is low, the PCM outputs will not toggle (PCMSCLK, PCMSYNC, and PCMSOUT). Additionally, when the enable is low, the internal divider-counters are held in reset.	0

The PCM_CTL register is used to control the various aspects of the PCM module. It also provides a status bit for polling control instead of interrupt based control.

38.6.1.2 PCM_CLKCTL

- Base Address: 0x1398_0000, 0x1399_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	0
SYNC_DIV	[8:0]	RW	Controls the frequency of the PCMSYNC signal based on the PCMSCLK.	000

38.6.1.3 PCM_TXFIFO

- Base Address: 0x1398_0000, 0x1399_0000
- Address = Base Address + 0x0008, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0
TXFIFO_ DVALID	[16]	RW	TxFIFO data is valid. Write: Not valid Read: TxFIFO read data valid 0 = Invalid (probably read an empty FIFO) 1 = Valid	1
TXFIFO_ DATA	[15:0]	RW	TxFIFO DATA Write: TXFIFO_DATA is written into the TxFIFO Read: TxFIFO is read using the APB interface NOTE: Reading the TxFIFO is meant to support debugging. Online the TxFIFO is read by the PCM serial shift engine, not the APB.	0

38.6.1.4 PCM_RXFIFO

- Base Address: 0x1398_0000, 0x1399_0000
- Address = Base Address + 0x000C, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0
RXFIFO_ DVALID	[16]	RW	RxFIFO data is valid. Write: Not Valid Read: TxFIFO read data valid 0 = Invalid (probably read an empty FIFO) 1 = Valid	1
RXFIFO_ DATA	[15:0]	RW	RxFIFO DATA Write: RXFIFO_DATA is written into the RxFIFO. NOTE: Writing the RxFIFO is meant to support debugging. Online the RxFIFO is written by the PCM serial shift engine, not the APB. Read: TxFIFO is read using the APB interface.	0

38.6.1.5 PCM_IRQ_CTL

- Base Address: 0x1398_0000, 0x1399_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:15]	–	Reserved	0
EN_IRQ_TO_ARM	[14]	RW	Controls whether or not the PCM interrupt is sent to the ARM. 0 = Does not forward PCM IRQ to the ARM subsystem 1 = Forwards PCM IRQ to the ARM subsystem	0
RSVD	[13]	–	Reserved	0
TRANSFER_DONE	[12]	RW	Generates interrupt every time the serial shift for a word completes. 0 = Disables IRQ source 1 = Enables IRQ source	0
TXFIFO_EMPTY	[11]	RW	Interrupt is generated whenever the TxFIFO is empty. 0 = Disables IRQ source 1 = Enables IRQ source enabled	0
TXFIFO_ALMOST_EMPTY	[10]	RW	Generates interrupt whenever the TxFIFO is ALMOST empty. Almost empty is defined as TX_FIFO_DEPTH < TX_FIFO_DIPSTICK 0 = Disables IRQ source 1 = Enables IRQ source enabled	0
TXFIFO_FULL	[9]	RW	Generates interrupt whenever the TxFIFO is full. 0 = Disables IRQ source 1 = Enables IRQ source	0
TXFIFO_ALMOST_FULL	[8]	RW	Generates interrupt whenever the TxFIFO is ALMOST full. Almost full is defined as TX_FIFO_DEPTH > (32– TX_FIFO_DIPSTICK) 0 = Disables IRQ source 1 = Enables IRQ source	0
TXFIFO_ERROR_STARVE	[7]	RW	Generates interrupt for TxFIFO starve error. This occurs whenever the TxFIFO is read when it is still empty. It considers this as an error and will have unexpected results 0 = Disables IRQ source 1 = Enables IRQ source	0
TXFIFO_ERROR_OVERFLOW	[6]	RW	Generates interrupt for TxFIFO overflow error. This occurs whenever it writes TxFIFO when it is already full. It considers this as an error and will have unexpected results.	0

Name	Bit	Type	Description	Reset Value
			0 = Disables IRQ source 1 = Enables IRQ source	
RXFIFO_EMPTY	[5]	RW	Generates interrupt whenever the RxFIFO is empty. 0 = Disables IRQ source 1 = Enables IRQ source	0
RXFIFO_ALMOST_EMPTY	[4]	RW	Generates interrupt whenever the RxFIFO is ALMOST empty. Almost empty is defined as RX_FIFO_DEPTH < RX_FIFO_DIPSTICK 0 = Disables IRQ source 1 = Enables IRQ source	0
RX_FIFO_FULL	[3]	RW	Generates interrupt whenever the RxFIFO is full. 0 = Disables IRQ source 1 = Enables IRQ source	0
RX_FIFO_ALMOST_FULL	[2]	RW	Generates interrupt whenever the RxFIFO is ALMOST full. Almost full is defined as RX_FIFO_DEPTH > (32– RX_FIFO_DIPSTICK) 0 = Disables IRQ source 1 = Enables IRQ source	0
RXFIFO_ERROR_STARVE	[1]	RW	Generates interrupt for RxFIFO starve error. This occurs whenever it reads RxFIFO when it is still empty. It considers this as an error and will have unexpected results. 0 = Disables IRQ source 1 = Enables IRQ source	0
RXFIFO_ERROR_OVERFLOW	[0]	RW	Generates interrupt for RxFIFO overflow error. This occurs whenever it writes RxFIFO when it is already full. It considers this as an error and will have unexpected results. 0 = Disables IRQ source 1 = Enables IRQ source	0

NOTE: The PCM_IRQ_CTL register is used to control the various aspects of the PCM interrupts.

38.6.1.6 PCM_IRQ_STAT

- Base Address: 0x1398_0000, 0x1399_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	–	Reserved	0
IRQ_PENDING	[13]	R	Monitoring PCM IRQ 0 = PCM IRQ does not occur 1 = PCM IRQ occurs	0
TRANSFER_DONE	[12]	R	Generates interrupt every time the serial shift for a word completes. 0 = IRQ does not occur 1 = IRQ occurs.	0
TXFIFO_EMPTY	[11]	R	Generates interrupt whenever the Tx FIFO is empty. 0 = IRQ does not occur 1 = IRQ occurs	0
TXFIFO_ALMOST_EMPTY	[10]	R	Generates interrupt whenever the Tx FIFO is ALMOST empty. 0 = IRQ does not occur 1 = IRQ occurs	0
TXFIFO_FULL	[9]	R	Generates interrupt whenever the Tx FIFO is full. 0 = IRQ occurs 1 = IRQ does not occur	0
TXFIFO_ALMOST_FULL	[8]	R	Generates interrupt whenever the Tx FIFO is ALMOST full. 0 = IRQ does not occur 1 = IRQ occurs	0
TXFIFO_ERROR_STARVE	[7]	R	Generates interrupt for Tx FIFO starve error. This occurs whenever it reads Tx FIFO when it is still empty. It considers this as an error and will have unexpected results. 0 = IRQ does not occur 1 = IRQ occurs	0
TXFIFO_ERROR_OVERFLOW	[6]	R	Generates interrupt for Tx FIFO overflow error. This occurs whenever it writes Tx FIFO when it is already full. It considers this as an error and will have unexpected results. 0 = IRQ does not occur 1 = IRQ occurs	0
RXFIFO_EMPTY	[5]	R	Generates interrupt whenever the Rx FIFO is empty. 0 = IRQ does not occur 1 = IRQ occurs	0
RXFIFO_ALMOST	[4]	R	Generates interrupt whenever the Rx FIFO is	0

Name	Bit	Type	Description	Reset Value
_EMPTY			ALMOST empty. 0 = IRQ does not occur 1 = IRQ occurs	
RX_FIFO_FULL	[3]	R	Generates interrupt whenever the Rx FIFO is full. 0 = IRQ does not occur 1 = IRQ occurs	0
RX_FIFO _ALMOST_FULL	[2]	R	Generates interrupt whenever the Rx FIFO is ALMOST full. 0 = IRQ does not occur 1 = IRQ occurs	0
RXFIFO_ERROR _STARVE	[1]	R	Generates interrupt for Rx FIFO starve error. This occurs whenever it reads Rx FIFO when it is still empty. It considers this as an error and will have unexpected results. 0 = IRQ does not occur 1 = IRQ occurs	0
RXFIFO_ERROR _OVERFLOW	[0]	R	Generates interrupt for Rx FIFO overflow error. This occurs whenever it writes Rx FIFO when it is already full. It considers this as an error and will have unexpected results. 0 = IRQ does not occur 1 = IRQ occurs	0

NOTE: The PCM_IRQ_STAT register is used to report IRQ status.

38.6.1.7 PCM_FIFO_STAT

- Base Address: 0x1398_0000, 0x1399_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:20]	–	Reserved	0
TXFIFO_COUNT	[19:14]	R	TxFIFO data count (0 to 32).	0
TXFIFO_EMPTY	[13]	R	To indicate whether TxFIFO is empty.	0
TXFIFO_ALMOST_EMPTY	[12]	R	To indicate whether TxFIFO is almost empty.	0
TXFIFO_FULL	[11]	R	To indicate whether TxFIFO is full.	0
TXFIFO_ALMOST_FULL	[10]	R	To indicate whether TxFIFO is almost full.	0
RXFIFO_COUNT	[9:4]	R	RxFIFO data count (0 to 32).	0
RXFIFO_EMPTY	[3]	R	To indicate whether RxFIFO is empty.	0
RXFIFO_ALMOST_EMPTY	[2]	R	To indicate whether RxFIFO is almost empty.	0
RX_FIFO_FULL	[1]	R	To indicate whether RxFIFO is full.	0
RX_FIFO_ALMOST_FULL	[0]	R	To indicate whether RxFIFO is almost full.	0

NOTE: The PCM_FIFO_STAT register is used to report FIFO status.

38.6.1.8 PCM_CLRINT

- Base Address: 0x1398_0000, 0x1399_0000
- Address = Base Address + 0x0020, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
PCM_CLRINT	[31:0]	W	The PCM_CLRINT register is used to clear the interrupt. Interrupt service routine is responsible for clearing interrupt asserted. Writing any values on this register clears interrupts for both ARM and DSP. Reading this register is not allowed. Clearing interrupt must be prior to resolving the interrupt condition; else, it may ignore another interrupt that would occur after this interrupt.	–

.

39 SPDIF Transmitter

39.1 Overview

The Sony Philips Digital Interconnect Format (SPDIF) transmitter is based on IEC60958.

This chapter describes a serial, un-directional, self-clocking interface to interconnect digital audio equipment in consumer and professional applications.

When you use a consumer digital processing environment in SPDIF standard, the SPDIF interface is primarily intended to carry stereophonic programs. With a resolution of up to 20 bits per sample, an extension to 24 bits per sample being possible.

When you use SPDIF in a broadcasting studio environment, the interface is primarily intended to carry monophonic or stereophonic programs. At a 48 kHz sampling frequency and with a resolution of up to 24 bits per sample; it can carry one or two signals sampled at 32 kHz.

In both cases, the clock references and auxiliary information are transmitted with the program. IEC60958 enables the interface to carry software-related data.

39.2 Features

- SPDIFOUT module only supports the consumer application in Exynos 4412 SCP
- Supports linear Pulse Code Modulation (PCM) up to 24-bit per sample
- Supports non-linear PCM formats such as AC3, MPEG1, and MPEG2
- 2 × 24-bit buffers which is alternately filled with data

39.3 Block Diagram

Figure 39-1 illustrates the block diagram of SPDIF Transmitter.

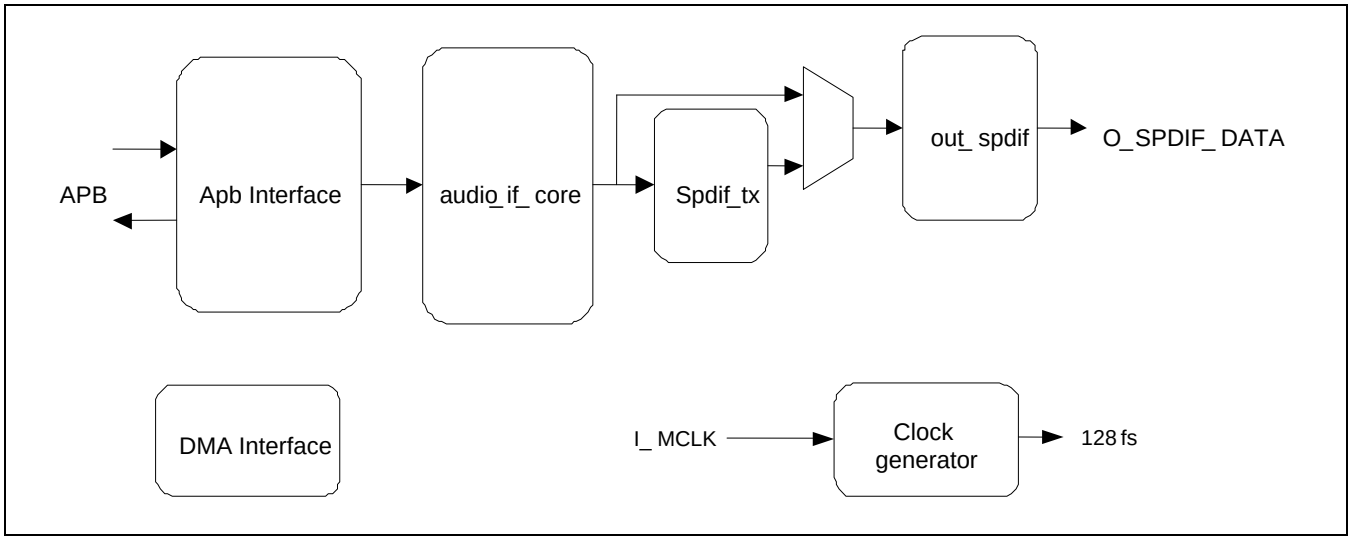


Figure 39-1 Block Diagram of SPDIFOUT

Components in SPDIF Transmitter are:

- APB interface block: This block defines register banks to control the driving of SPDIFOUT module and data buffers to store linear or non-linear PCM data.
- DMA interface block: This block requests DMA service to IODMA. This depends on the status of data buffer in APB interface block
- Clock Generator block: This block generates 128 fs (sampling frequency) clock. used in out_spdif block from system audio clock (MCLK)
- Audio_if_core block: This block acts as interface block between data buffer and out_spdif block. Finite-state machine controls the flow of PCM data.
- Spdif_tx block: This block inserts burst preamble and executes zero-stuffing (filling by zero) in the non-linear PCM stream. spdif_tx module bypasses the linear PCM data.
- Out_spdif block: This block generates SPDIF format. It inserts these into the appropriate position of 32-bit word:
  - 4-bit preamble
  - 16- or 20- or 24-bit data
  - User-data bit,
  - Validity bit
  - Channel status bit
  - Parity bit

Out_spdif block modulates each bit to bi-phase format.

39.4 Functional Description

This section includes:

- Data Format of SPDIF
- Channel Coding
- Preamble
- Non-Linear PCM Encoded Source (IEC 61937)
- SPDIF Operation
- Shadowed Register

39.4.1 Data Format of SPDIF

This section includes:

- Frame Format
- Sub-frame Format (IEC 60958)

39.4.1.1 Frame Format

A frame consists of two sub-frames. The transmission rate of frames exactly corresponds to the source sampling frequency.

In the 2-channel operation mode, the time multiplexing transmits samples taken from both channels in consecutive sub-frames.

Sub-frames related to channel 1 (left or "A" channel in stereophonic operation and primary channel in monophonic operation) normally use preamble M. However, the preamble is changed to preamble B once in every 192 frames. This unit defines the block structure. You can use this block structure to organize the channel status information.

Sub-frames of channel 2 (right or "B" in stereophonic operation and secondary channel in monophonic operation) always use preamble W.

In broadcasting studio environment of the single channel operation mode, the frame format is identical to the 2-channel mode. It carries the data only in channel 1. In the sub-frames that are allocated to channel 2, time slot 28 (validity flag) should be set to logical "1" ("1" refers to not valid).

Figure 39-2 illustrates the SPDIF frame format.

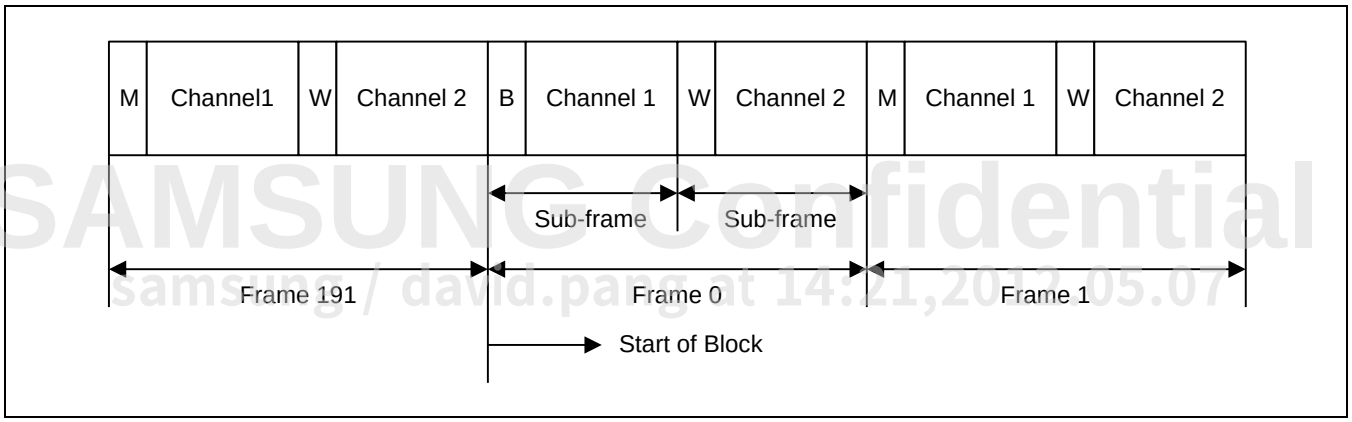


Figure 39-2 SPDIF Frame Format

39.4.1.2 Sub-frame Format (IEC 60958)

Each sub-frame is divided into 32 time slot, which are numbered from 0 to 31. Time slots 0 to 3 carry one of the three permitted preambles. These are used to affect synchronization of sub-frames, frames, and blocks. Time slots 4 to 27 carry the audio sample word in linear 2's complement representation. Time slot 27 carries the most significant bit. When a 24-bit coding range is used, the least significant bit is in time slot 4. When a 20-bit coding range is sufficient, the least significant bit is in time slot 8. And time slot 4 to 7 may be used for other application. Under these circumstances, the bits in the time slot 4 to 7 are designated auxiliary sample bits.

When the source provides fewer bits than the interface allows (24 or 20), it sets the unused least significant bits to a logical "0". This procedure supports to connect equipment by using different numbers of bits.

Time slot 28 carries the validity flag that is associated with the audio sample word. This flag is set to logical "0" when the audio sample is reliable.

Time slot 29 carries one bit of the user data that is associated with the audio channel. Transmitted in the same sub-frame. The default value of the user bit is logical "0".

Time slot 30 carries one bit of the channel status words associated with the audio channel. Transmitted in the same sub-frame.

Time slot 31 carries a parity bit so that the time slots, inclusive of 4 to 31, carry an even number of ones and zeroes.

Figure 39-3 illustrates the SPDIF sub-frame format.

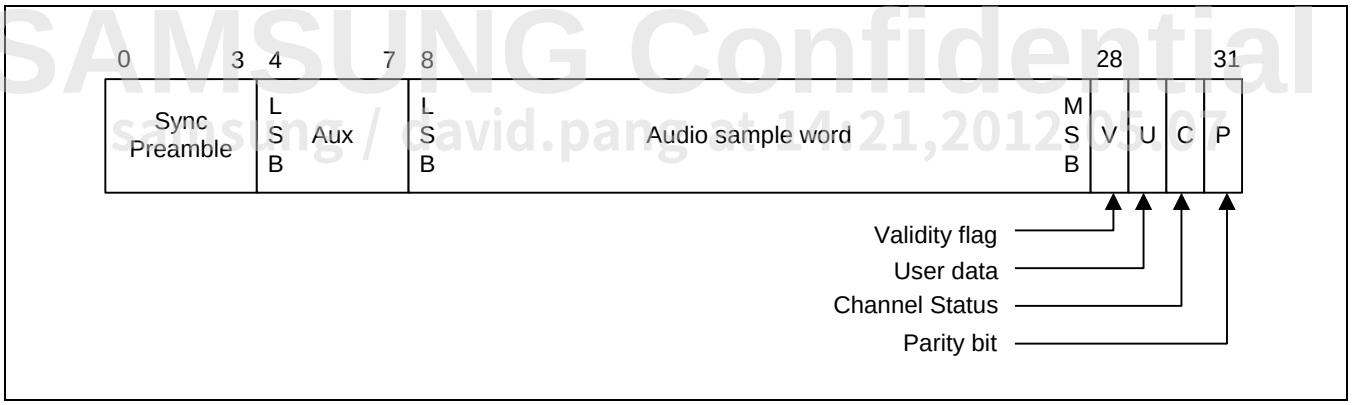


Figure 39-3 SPDIF Sub-Frame Format

39.4.2 Channel Coding

It encodes time slots, 4 to 31, in bi-phase-mark to perform these tasks:

- Minimize the Dynamic Current (DC) component on the transmission line
- Facilitate clock recovery from the data stream
- Make the interface insensitive to the polarity of connections

A symbol comprising two consecutive binary states represent each bit to be transmitted. The first state of a symbol is always different from the second state of the previous symbol. The second state of the symbol is identical to the first when the bit to be transmitted is logical "0". The second bit is different from the first when the bit is logical "1".

Figure 39-4 illustrates Channel Coding.

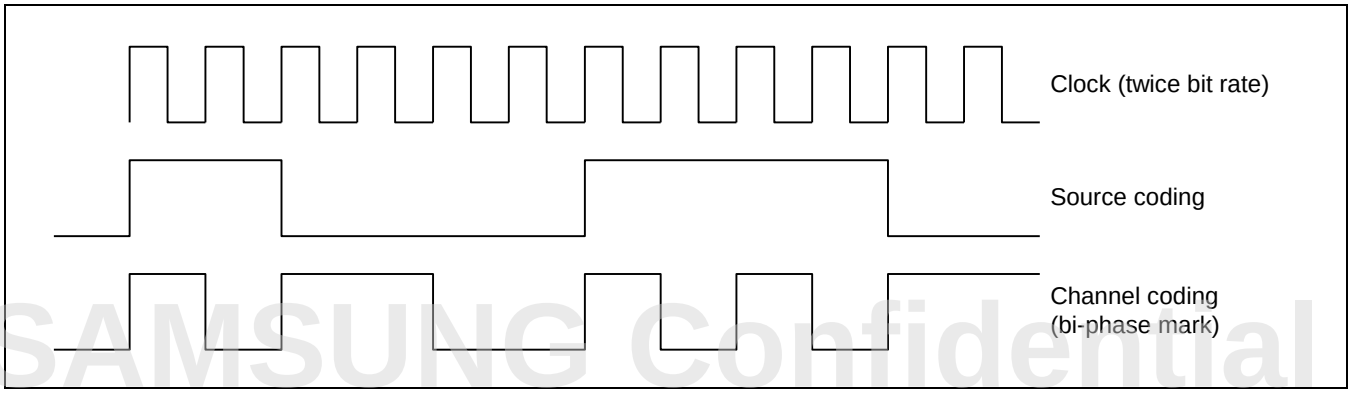


Figure 39-4 Channel Coding

39.4.3 Preamble

Preambles are specific patterns that provide synchronization and identification of the sub-frames and blocks. A set of three preambles (M, B and W) is used. These preambles are transmitted in the time allocated to four time slots (time slots 0 to 3) and are represented by eight successive states. The first state of the preamble is always different from the second state of the previous symbol.

Similar to bi-phase code, these preambles are DC free and provide clock recovery. They differ in minimum two states from any valid bi-phase sequence.

39.4.4 Non-Linear PCM-Encoded Source (IEC 61937)

The non-linear PCM encoded audio bit stream is transferred using the basic 16-bit data area of the IEC 60958 sub frames, that is, in time slots 12 to 27. Each IEC 60958 frame transfers 32 bits of the non-PCM data in consumer application mode.

When the SPDIF bit stream conveys linear PCM audio, the symbol frequency is 64 times that of the sampling frequency of PCM (32 time slots per PCM, and each Channel is sampled with 2 PCM.). When a non-linear PCM-encoded audio bit stream is transmitted by the interface, the symbol frequency is 64 times that of the sampling rate of the encoded audio within that bit stream.

If a non-linear PCM encoded audio bit stream is transmitted by the interface containing audio with low sampling frequency, the symbol frequency shall be 128 times the sampling rate of the encoded audio within that bit stream.

Each data burst contains a burst-preamble consisting of four 16-bit words, namely:

- Pa
- Pb
- Pc
- Pd

Followed by the burst-payload which contains data of an encoded audio frame.

The burst-preamble consists of four mandatory fields, namely:

- Pa and Pb represent a synchronization word.
- Pc provides information about the type of data and some information/control for the receiver.
- Pd provides the length of the burst-payload, limited to 216 (= 65,535) bits.

Two sequential SPDIF frames contain the four preamble words:

- The frame that begins the data-burst contains Pa and Pb in sub-frame 1 and sub-frame 2, respectively.
- The next frame contains Pc and Pd in sub-frame 1 and sub-frame 2, respectively.

When placed into a SPDIF sub-frame, the MSB of a 16-bit burst-preamble is placed into time slot 27 and the LSB is placed into time slot 12.

Figure 39-5 illustrates the format of burst-payload.

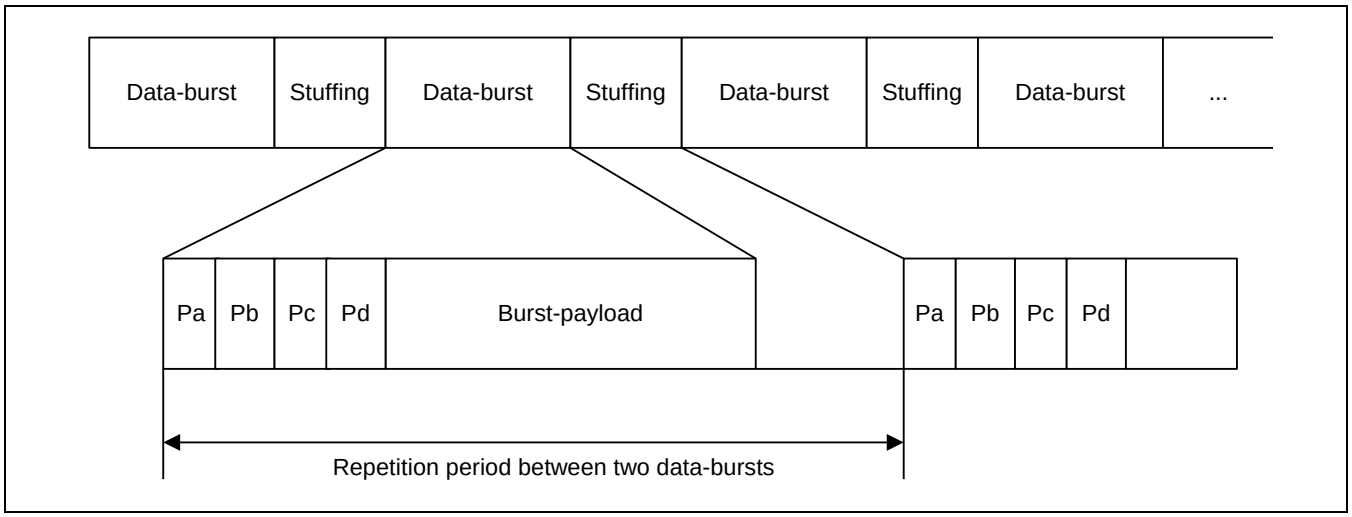


Figure 39-5 Format of Burst-Payload

Table 39-1 lists the burst-preamble words.

Table 39-1 Burst Preamble Words

Preamble Word	Length of Field	Contents	Value MSB. LSB
Pa	16 bits	Sync word 1	0xF872
Pb	16 bits	Sync word 2	0x4E1F
Pc	16 bits	Burst-info	Refer to section *39.6.1.7 SPDBSTAS_SHD*[15:0] for more information
Pd	16 bits	Length-code	Refer to *39.6.1.7 SPDBSTAS_SHD*[31:16] for more information

39.4.5 SPDIF Operation

As the bit frequency of SPDIF is 128 fs (fs: sampling frequency), divide audio main clock (MCLK) depending on its frequency to make the main clock of SPDIF.

You can divide MCLK by:

- 2 in case of 256 fs
- 3 in case of 384 fs
- 4 in case of 512 fs

SPDIF module in Exynos 4412 SCP changes the audio sample data format to SPDIF.

To change the format SPDIF module:

4. Inserts preamble data, channel status data, user data, error check bit and parity bit into the appropriate time slots.
5. Fixes the preamble data in the module and inserts depending on sub-frame counter.
6. Sets channel status data in the SPDCSTAS register and used by one bit per frame.

User data always have zero values.

For non-linear PCM data, insert burst-preamble (Pa, Pb, Pc, and Pd) before burst-payload. It pads zero from the end of burst-payload to the repetition count. It fixes Pa (= 16'hF872) and Pb (= 16'h4E1F) in the module and it sets Pc and Pd in the SPDBSTAS register.

To stuff zero, the end of burst-payload is calculated from Pd value and repetition count. This depends on data type in the preamble Pc that is acquired from SPDCNT register.

Audio data are justified to the LSB. 16-, 20- or 24-bit PCM data and 16-bit stream data are supported. The unoccupied upper bits of 32-bit word are ignored.

Data are fetched via DMA request. When one of two data buffers is empty, DMA service is requested. Audio data stored in the data buffers are transformed into SPDIF format and output to the port. For non-linear PCM data, interrupt is generated after audio data are output up to the value specified in the SPDCNT register. Interrupt sets registers such as SPDBSTAS and SPDCNT to new values when data type of new bitstream is different from the previous one.

39.4.6 Shadowed Register

Both SPDBSTAS_SHD and SPDCNT_SHD registers are shadowed registers that are related to SPDBSTAS and SPDCNT registers, respectively. They are updated from related registers at every stream end interrupt signal. The usage of shadowed register is as follows.

7. Set the burst status and repetition count information to their respective registers.
8. Turn on SPDIF module, and stream end interrupt is asserted immediately.
9. With stream end interrupt, shadowed registers are updated from their related registers and SPDIF begins to transfer data. Now next stream information (burst status and repetition count) can be written to SPDBSTAS and SPDCNT register because previous information is copied to their respective shadowed registers.
10. Set next stream information to SPDBSTAS and SPDCNT registers.
11. Wait for stream end interrupt that signals the end of the first stream.
    With stream end interrupt, the 2nd stream data will start to transfer.
12. Set the third stream information to registers.

The usage of user bit registers is similar to stream information registers. However, that they are not related to SPDIF end interrupt but to user data interrupt.

As soon as SPDIF is on, shadowed user bit registers are updated from their related registers. And user bit starts to be shifted out with user data interrupt asserted. User can write the next user data to registers with this interrupt.

After entire 96 user bits are shifted out, user data interrupt will be asserted again and 3rd user bits can be written to registers with 2nd user bits going out.

39.5 I/O Description

SPDIF external pads are shared with I2S and PCM. To use these pads for SPDIF, GPIO must be set before the SPDIF starts. Refer to the GPIO chapter for more information.

Table 39-2 describes List of I/O Description.

Table 39-2 I/O Description

Signal	I/O	Description	Pad	Type
SPDIF_EXTCLK	I	Global audio main clock (External MCLK)	Xpcm0EXTCLK	Muxed
SPDIF_0_OUT	O	SPDIFOUT data output	Xpcm0SCLK	Muxed

39.6 Register Description

39.6.1 Register Map Summary

- Base Address: 0x139B_0000

Register	Offset	Description	Reset Value
SPDCLKCON	0x0000	Specifies the clock control register	0x0000_0002
SPDCON	0x0004	Specifies the control register	0x0000_0000
SPDBSTAS	0x0008	Specifies the burst status register	0x0000_0000
SPDCSTAS	0x000C	Specifies the channel status register	0x0000_0000
SPDDAT	0x0010	Specifies the SPDIFOUT data buffer	0x0000_0000
SPDCNT	0x0014	Specifies the repetition count register	0x0000_0000
SPDBSTAS_SHD	0x0018	Specifies the shadowed burst status register	0x0000_0000
SPDCNT_SHD	0x001C	Specifies the shadowed repetition count register	0x0000_0000
USERBIT1	0x0020	Specifies the sub-code Q1 to Q32	0x0000_0000
USERBIT2	0x0024	Specifies the sub-code Q33 to Q64	0x0000_0000
USERBIT3	0x0028	Specifies the sub-code Q65 to Q96	0x0000_0000
USERBIT1_SHD	0x002C	Specifies the shadowed register userbit1	0x0000_0000
USERBIT2_SHD	0x0030	Specifies the shadowed register userbit2	0x0000_0000
USERBIT3_SHD	0x0034	Specifies the shadowed register userbit3	0x0000_0000
VERSION_INFO	0x0038	Specifies the RTL version information	0x0000_000D

39.6.1.1 SPDCLKCON

- Base Address: 0x139B_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0002

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0
MCLK SEL	[3:2]	RW	Main audio clock selection 0 = Selects internal clock (I_MCLK_INT) 1 = Selects external clock (I_MCLK_EXT0)	0
SPDIFOUT Clock Down Ready (Read only)	[1]	R	0 = Clock-Down not Ready 1 = Clock-Down Ready	0
SPDIFOUT power on	[0]	RW	0 = Power Off 1 = Power On	0

39.6.1.2 SPDCON

- Base Address: 0x139B_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0
FIFO Level	[26:22]	R	First In First Out (FIFO) level monitoring (read only) Depth of FIFO is 16 0 = FIFO level is empty 16 = FIFO level is full	00000
FIFO Level Threshold	[21:19]	RW	FIFO Threshold Level is controllable 000 = 0-FIFO Level 001 = 1-FIFO Level 010 = 4-FIFO Level 011 = 6-FIFO Level 100 = 10-FIFO Level 101 = 12-FIFO Level 110 = 14-FIFO Level 111 = 15-FIFO Level	000
FIFO transfer mode	[18:17]	RW	00 = DMA transfer mode 01 = Polling mode 10 = Interrupt mode 11 = Reserved	00
FIFO_level Interrupt Status	[16]	RWX	Read Operation 0 = No interrupt pending. 1 = Interrupt pending. Write Operation 0 = No effect 1 = Clears this flag	0
FIFO_level Interrupt Enable	[15]	RW	0 = Interrupt masked 1 = Enables interrupt	0
endian format	[14:13]	RW	00 = big endian - o_data = {in_data[23:0]} 01 = 4 byte swap - o_data = {in_data[15:8], in_data[23:16], in_data[31:24]} 10 = 3 byte swap - o_data = {in_data[7:0], in_data[15:8], in_data[23:16]} 11 = 2 byte swap - o_data = {0x00, in_data[7:0], in_data[15:8]} NOTE: in_data: BUS → in port of SPDIF o_data: in port of SPDIF → Logic	0
user_data_attach	[12]	RW	0 = Stores user data in USERBIT register User data of sub-frame is out from USERBIT1, 2,	0

Name	Bit	Type	Description	Reset Value
			3 (96-bit) 1 = Stores user data in 23rd bit of audio data User data is out in PCM 23th bit of data.	
User Data Interrupt Status	[11]	RW	Read Operation 0 = No interrupt pending 1 = Interrupt pending when 96-bit of user data is out Write Operation 0 = No effect 1 = Clears this flag	0
User Data Interrupt Enable	[10]	RW	0 = Interrupt masked 1 = Enables interrupt	0
Buffer Empty Interrupt Status	[9]	RW	Read Operation 0 = No interrupt pending 1 = Interrupt pending Write Operation 0 = No effect 1 = Clears this flag	0
Buffer Empty Interrupt Enable	[8]	RW	0 = Interrupt masked 1 = Enables interrupt	0
Stream End Interrupt Status	[7]	RWX	Read Operation 0 = No interrupt pending 1 = Interrupt pending when the number of output audio data reaches repetition count in SPDCNT register. Write Operation 0 = No effect 1 = Clears this flag	0
Stream End Interrupt Enable	[6]	RW	0 = Interrupt masked 1 = Enables interrupt	0
software reset	[5]	S	0 = Normal operation 1 = Software reset Software reset is 1-cycle pulse (auto clear) Enables I_MCLK before software reset assertion because SPDIF uses synchronous reset	0
Main Audio Clock Frequency	[4:3]	RW	00 = 256 fs 01 = 384 fs 10 = 512 fs 11 = Reserved When you want to use SPDIF on HDMI, you should select 512 fs because HDMI in Exynos 4412 SCP accepts only 512 fs or more frequency.	0
PCM Data Size	[2:1]	RW	00 = 16-bit 01 = 20-bit 10 = 24-bit	0

Name	Bit	Type	Description	Reset Value
			11 = Reserved	
PCM or Stream	[0]	RW	0 = Stream 1 = PCM	0

39.6.1.3 SPDBSTAS

- Base Address: 0x139B_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Burst data length bit	[31:16]	RW	ES size in bits (Burst Preamble Pd) ES size: Elementary Stream size This indicates burst-payload length	0
Bitstream number	[15:13]	RW	Bit_stream_number. It is set to 0	0
Data type dependent info	[12:8]	RW	Data type dependent information	0
Error flag	[7]	RW	0 = Error flag indicates a valid burst-payload 1 = Error flag indicates that the burst-payload may contain errors	0
RSVD	[6:5]	–	Reserved	0
Compressed data type	[4:0]	RW	00000 = Null Data 00001 = AC-3 00010 = Reserved 00011 = Pause 00100 = MPEG1 (layer1) 00101 = MPEG1 (layer2, 3), MPEG2-bc 00110 = MPEG2-extension 00111 = Reserved 01000 = MPEG2 (layer1-lsf) 01001 = MPEG2 (layer2, layer3-lsf) Others = Reserved	0

39.6.1.4 SPDCSTAS

- Base Address: 0x139B_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
Clock accuracy	[29:28]	RW	00 = Level II, ± 1000 ppm 01 = Level III, variable pitch shifted 10 = Level I, ± 50 ppm	0
Sampling frequency	[27:24]	RW	0000 = 44.1 kHz 0010 = 48 kHz 0011 = 32 kHz 1010 = 96 kHz	0
Channel number	[23:20]	RW	Bit[20] is LSB	0
Source number	[19:16]	RW	Bit[16] is LSB	0
Category code	[15:8]	RW	Equipment type CD player = 0000_0001 DAT player = L000_0011 DCC player = L100_0011 Mini disc = L100_1001 (L: Information about generation status of the material)	0
Channel status mode	[7:6]	RW	00 = Mode 0 Others = Reserved	0
Emphasis	[5:3]	RW	When bit[1] = 0, 000 = 2 Audio channels without pre-emphasis 001 = 2 Audio channels with 50 µs/15 µs pre-emphasis When bit[1] = 1, 000 = Default state	0
Copyright assertion	[2]	RW	0 = Copyright 1 = No copyright	0
Audio sample word	[1]	RW	0 = Linear PCM 1 = Non-linear PCM	0
Channel status block	[0]	RW	0 = Consumer format 1 = Professional format	0

39.6.1.5 SPDDAT

- Base Address: 0x139B_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0
SPDIFOUT data	[23:0]	W	PCM or stream data	0

39.6.1.6 SPDCNT

- Base Address: 0x139B_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	0
Stream repetition count	[12:0]	W	Repetition count according to data type. This bit is valid only for stream data.	0

39.6.1.7 SPDBSTAS_SHD

- Base Address: 0x139B_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Burst data length bit	[31:16]	R	ES size in bits (Burst Preamble Pd) ES size: Elementary Stream size This indicates Burst-payload length	0
Bitstream number	[15:13]	R	Bit_stream_number. It is set to 0	0
Data Type dependent info	[12:8]	R	Data type dependent information	0
Error flag	[7]	R	0 = Error flag indicates a valid burst-payload 1 = Error flag indicates that the burst payload may contain errors	0
RSVD	[6:5]	–	Reserved	0
Compressed data type	[4:0]	R	00000 = Null Data 00001 = AC–3 00010 = Reserved 00011 = Pause 00100 = MPEG1 (layer1) 00101 = MPEG1 (layer2, 3), MPEG-bc 00110 = MPEG2-extension 00111 = Reserved 01000 = MPEG2 (layer1-lsf) 01001 = MPEG2 (layer2, layer3-lsf) Others = Reserved	0

39.6.1.8 SPDCNT_SHD

- Base Address: 0x139B_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	0
Stream repetition count	[12:0]	R	Repetition count according to data type. This bit is valid only for stream data.	0

39.6.1.9 USERBITn (n = 1 to 3)

- Base Address: 0x139B_0000
- Address = Base Address + 0x0020, + 0x0024, + 0x0028, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
User data bit (Subcode Q for CD)	[31:0]	RW	USERBIT1 = Q1 to Q32 USERBIT2 = Q33 to Q64 USERBIT3 = Q65 to Q96 User data bit has the Digital Audio Track information (Track NO, Play Time and so on.). 1176 bits of these being taken out in a row.	0

39.6.1.10 USERBIT_SHDn (n = 1 to 3)

- Base Address: 0x139B_0000
- Address = Base Address + 0x002C, + 0x0030, + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
User data bit	[31:0]	R	USERBIT1_SHD = Q1 to Q32 USERBIT2_SHD = Q33 to Q64 USERBIT3_SHD = Q65 to Q96 User data bit has the Digital Audio Track information (Track NO, Play Time and so on.). 1176 bits of these being taken out in a row.	0

39.6.1.11 VERSION_INFO

- Base Address: 0x139B_0000
- Address = Base Address + 0x0038, Reset Value = 0x0000_000D

Name	Bit	Type	Description	Reset Value
Version information	[31:0]	R	RTL Version Information	0x0000_000D

40 High-Speed Synchronous Serial Interface (HSI)

40.1 Overview

HSI is an interface intended to connect an application processor to a cellular modem controller in cellular handsets, but also can be used in other applications. HSI provides a low-latency communication channel over a chip-to-chip link by dividing the physical link to logical channels at the hardware level. HSI enables versatile, easy and simple serial link realizations. HSI can reduce time to market and design cost of mobile handsets by simplifying serial chip-to-chip physical layer implementations.

40.2 Features

HSI Controller has the following features:

- 32-bit AHB interface to SOC interconnect
- It supports PIO and SDMA modes
- 8 SDMAs are supported
- Data Time Out feature supported for receiving operation
- Compliant to HSI specification version 1.0
- Full-Duplex High Speed Serial Interface
- Supports 8-logical channels for both transmitting and receiving operations
- Each channel is configurable and can be enabled or disabled
- Bandwidth for each channel is programmable
- Supports both Stream Mode and Frame Mode for data transmission and reception
- Maximum bandwidth of 200 Mbps in both transmitting and receiving directions
- Configurable Transmit and Receive FIFOs
- Programmable Transmission bit rate
- Supports both round-robin and priority-based arbitration for transmitting operation
- Run-time configurability of channel id bits.

40.3 Functional Description

40.3.1 Block Diagram

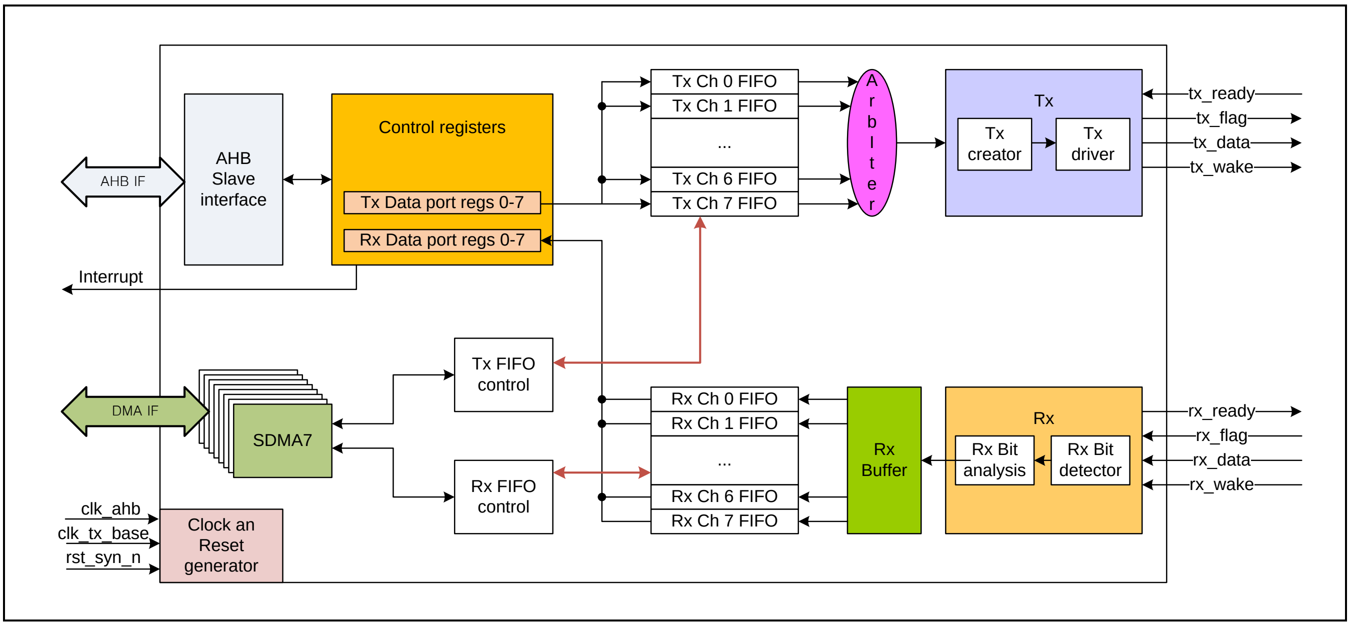


Figure 40-1 Functional Block Diagram of HSI Controller

- AHB Slave Interface
  - The HSI Controller is connected to the SOC AHB interconnect using the AHB Slave Interface. The interface is 32 bits. The firmware can access the HSI controller through AHB IF. The firmware can program the control registers of the HSI through this interface. It can send the data to Tx FIFO and read the data from Rx FIFO through this interface.
- Control Registers
  - This block has all the control registers of HSI Controller. For details about the registers, Refer to "HSI Registers".
- Tx FIFO
  - The Tx FIFO size is configurable and you can partition among eight channels. For example, assuming a FIFO size of 4096 Bytes, Ch0 can be allocated to the entire FIFO by programming the register "TX Ch0 FIFO size" to b1010 (i.e. Ch0 has a size of 4096 Bytes and the others have 0 Bytes). These eight Tx FIFO's provide data storage for eight channels. You can enable or disable each channel based on the application's need.
  - Writing to the Tx FIFOs happens on ahb_clk and reading from FIFOs happens on derived clk_tx. Tx Ch0 FIFO is used as control channel. You can send configuration commands to Rx of other die by writing to Tx Ch0 FIFO.
- Arbiter
  - The Arbiter arbitrates the requests from all eight channels. It can be programmed to be fixed or Round Robin priority. Based on priority, the Arbiter arbitrates the request from each channel and data from respective channel gets serialized and transmitted over HSI serial interface.

- Tx
  - Tx module consists of Tx frame creator and Tx frame driver blocks. The Tx frame creator generates a frame by appending the channel ID bits to the data from the Tx FIFO. The Tx frame driver block serializes the frame generated to the HSI IF.
- Rx
  - The Rx block consists of Rx bit detector and Rx bit analysis blocks. The Rx bit detector de-serializes the serial data got from the HSI IF and the Rx bit analysis writes the data into the Rx buffer. The receive clock is derived from the "RX_DATA" and "RX_FLAG", which is used to sample the serial data ("RX_DATA").
- Rx Buffer
  - The output of the Rx block is stored in the Rx buffer. This temporary buffer is used to avoid the head of line blocking in HSI during receive mode. The buffer size is configurable.
- Rx Fifo
  - The Rx FIFO size is configurable and you can partition among eight channels. For example, assuming a FIFO size of 4096 Bytes, Ch0 can be allocated to the entire FIFO by programming the register "Rx Ch0 FIFO size" to b1010 (i.e. Ch0 has a size of 4096 Bytes, and the others have 0 Bytes). These eight Rx FIFOs provide data storage for eight channels. You can enable or disable each channel based on the application's need
  - The HSI Controller generates an interrupt when data is available to read from the Rx FIFO. You can read the data by accessing "Rx FIFO Data port registers" of Control registers block.
- Clock Generator
  - The Clock Generator module generates the "clk_tx" from "clk_tx_base". The divisor value is programmed using the "Clock Control Register".
- SDMA
  - The HSI Controller supports Slave DMA (SDMA) feature through AHB Slave Interface. The HSI Controller supports eight SDMA channels. The DMA transfer size, Direction and DMA burst size are programmable.
- Clocks and Reset Generator
  - All clocks are asynchronous to each other.

40.3.2 Interface Port Description

Signals	I/O	Description	Pad Name
hsi_tx_wake	Out	Assertion of this signal shows that HSI TX wants to transfer some data. This signal is used to wake up HSI Rx on the opposite side of HSI IF.	Xm0ADDR[12]
hsi_tx_data	Out	Serial data out	Xm0ADDR[13]
hsi_tx_flag	Out	Flag signal. If data on tx_data is toggling, then flag stays constant and if data is constant, then flag should toggle	Xm0ADDR[14]
hsi_tx_ready	In	Link data flow control signal. It indicates HSI RX of other die is ready to receive the data on tx_data port	Xm0ADDR[15]
hsi_rx_wake	In	Link Control signal used to wake-up the receiver so that transmitter starts a transmission	Xm0ADDR[8]
hsi_rx_data	In	Serial data input	Xm0ADDR[9]
hsi_rx_flag	In	Flag signal. If data on rx_data is toggling then flag stays constant and if data is constant then flag should toggle. It can be used with rx_data to derive receive clock at HSI Rx	Xm0ADDR[10]
hsi_rx_ready	Out	Link data flow control signal. It indicates HSI Rx is ready to receive the data on rx_data port	Xm0ADDR[11]

40.3.3 Programming Basic Sequence Guide

This section defines the basic sequence flow chart divided into several sub sequences.

40.3.3.1 Transmit or Receive Control Flow in PIO Mode

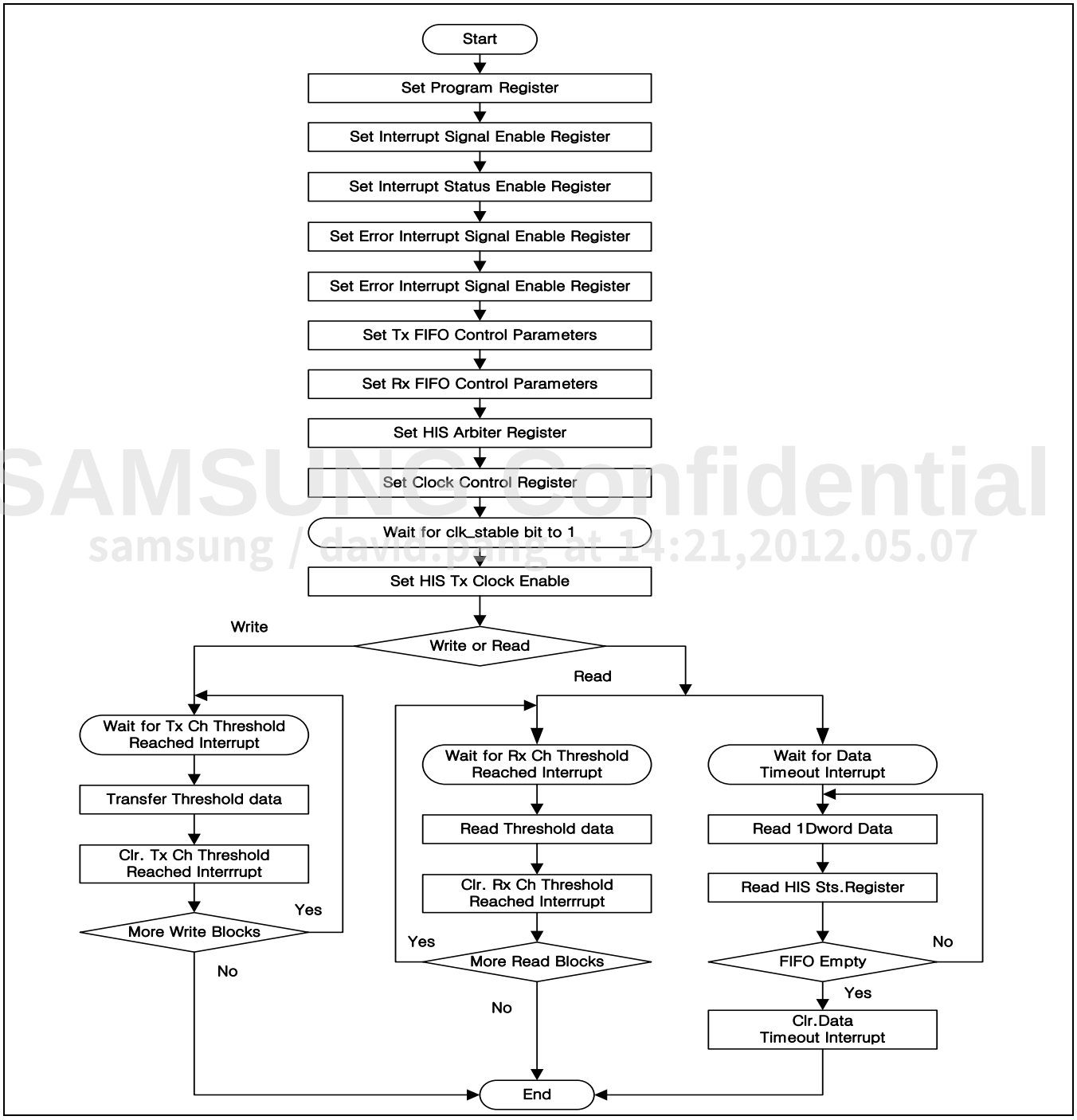
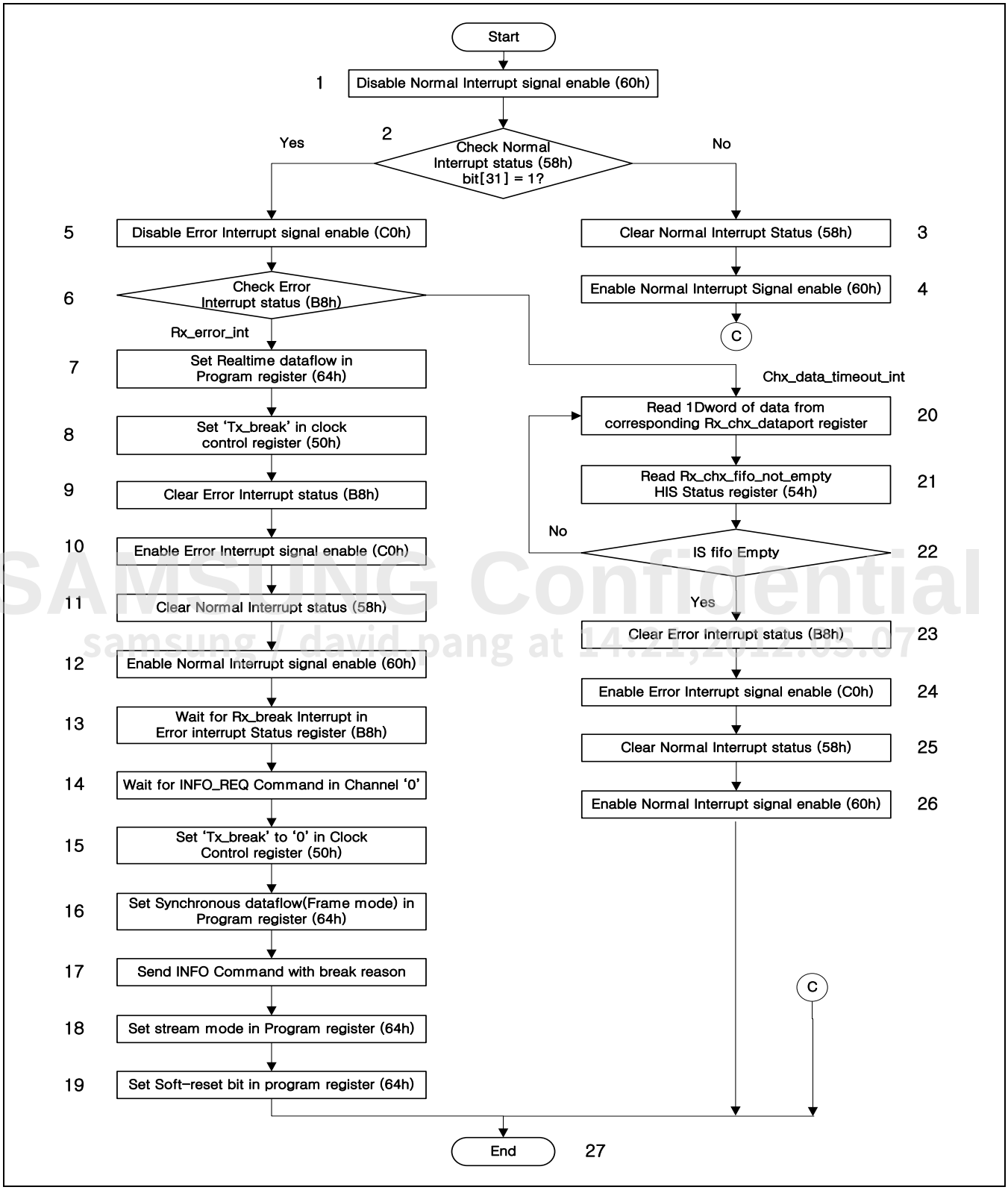


Figure 40-2 Transmit or Receive Control Flow in PIO Mode

Step Number	Description
1	Program the filed in "Program" register.
2	Program the "Interrupt Signal Enable" register
3	Program the "Interrupt Status Enable" register
4	Program the "Error Interrupt Signal Enable" register
5	Program the "Error Interrupt Status Enable" register
6	Program the Tx FIFO size and threshold in "Tx FIFO control" registers.
7	Program the Rx FIFO size and threshold in "Rx FIFO control" registers.
8	Program the priority of the channels in "HSI Arbiter Priority" register. Program the bandwidth for the channels in "HSI Arbiter Bandwidth1" and "HSI Bandwidth2" registers
9	Program a divisor value in "Clock Control" register.
10	Wait until "clock stable" bit of "Clock Control" register becomes one.
11	Set "HSI Tx clk enable" bit "Clock Control" register to one. Then HSI controller starts suppling the clk_tx to the Tx related logic.
12	For a write operation, go to step 13-W. For a read operation, if the data to be received is greater than or equal to Rx Ch threshold then go to step 13-R-a otherwise go to step 13-R-b.
13-W	Wait for "Tx Ch Threshold Reached" interrupt.
14-W	Write the threshold data to "Tx Data Port" register.
15-W	Clear "Tx Ch Threshold Reached" interrupt by writing one to the "Tx Ch Threshold Reached" field of HSI "Interrupt status register".
16-W	Repeat until all the blocks are written.
13-R-a	Wait for "Rx Ch Threshold Reached" interrupt.
14-R-a	Read threshold data from "Rx Data Port" register.
15-R-a	Clear "Rx Ch Threshold Reached" interrupt by writing one to the "Rx Ch Threshold Reached" field of HSI "Interrupt status register"
16-R-a	Repeat until all the data is read
13-R-b	Wait for "Data Timeout" interrupt.
14-R-b	Read 1Dword from "Rx Data Port" register.
15-R-b	Read "HSI Status" register to check Rx FIFO is empty.
16-R-b	Repeat steps 14-R-b and 15-R-b until the Rx FIFO is empty.
17-R-b	Clear "Data Timeout" interrupt by writing one to the "Data Timeout interrupt" field of HSI "Interrupt status register"

40.3.3.2 Resynchronization Sequence for Bus Contention



Step Number	Description
1	After reception of interrupt from HSI controller, disable all "Normal Interrupt Signal Enable" bits in Normal Interrupt Signal Enable Register (60h)
2	Read Normal Interrupt Status Register (58h). If "Error Interrupt" (bit 31) is set then goto step(5), otherwise goto step(3)
3	When Tx_chx_threshold_reached_int or Rx_chx_threshold_reached_int bit is set in Normal interrupt status Write/Read Threshold of data to/from Tx/Rx data_port_register respectvely. where x ranges from 0 to7. Write 1 to clear Normal Interrupt Status Register (58h). When Transfer complete int bit is set write 1 to clear in Normal Interrupt Status Register (58h)
4	Enable all "Normal Interrupt Signal Enable" bits in Normal Interrupt Signal Enable Register (60h), then goto step(27)
5	Disable all "Error Interrupt Signal Enable" bits in Error Interrupt Signal Enable Register (C0h)
6	Read Error Interrupt Status Register (B8h). If "Rx_error_interrupt" (bit 0) is set to "1", then goto step (7), otherwise if "data_timeout_intterrupt_for_chx" (bits[9:2]) is set to "1", then goto step (20). where x ranges from 0 to7
7	Set realtime dataflow (bits[10:8]) (frame mode) Program Register (64h).
8	Set "Tx_break" (bit 15) to "1" in Clock Control Register (50h) to send continuos zeros
9	Write "1" to clear Error Interrupt Status Register (B8h)
10	Enable all "Error Interrupt Signal Enable" bits in Error Interrupt Signal Enable Register (C0h)
11	Write "1" to clear Normal Interrupt Status Register (58h)
12	Enable all "Normal Interrupt Signal Enable" bits in Normal Interrupt Signal Enable Register (60h)
13	Wait for "Rx_break_int" (bit 1) in Error Interrupt Status Register (B8h)
14	Wait for "INFO_REQ" command with break reason class reception through Rx_channel0 (Rx_ch0_threshold_reached_interrupt will be received, then read Rx Ch0 Dataport Register (98h))
15	Set "Tx_break" (bit 15) to "0" in Clock Control Register (50h)
16	Set Program Register (64h) to frame mode and synchronous dataflow (bits[10:8])
17	Transmit "INFO" command with break reason through Tx_channel0 (wait for Tx_ch0_threshold_reached interrupt, then write INFO command to Tx_ch0_dataport Register (78h))
18	Set synchronous dataflow (bits[10:8]) stream mode in Program Register (64h).
19	Perform Soft Reset by setting "Software Reset" (bit 0) to "1" in Program Register (64h). After automatic clear of "software reset" bit goto step (27)
20	Read 4 Bytes of data from corresponding Rx Chx Dataport Register (98h-B4h) for which Data timeout interrupt is received
21	Read "Rx Channelx FIFO not empty" bit in HSI Status Register (54h).where x ranges from 0 to7
22	If 'Rx Channelx FIFO not empty' bit is "1", then goto step (20) otherwise goto step (23). where x ranges from 0 to7
23	Write "1" to clear Error Interrupt Status Register (B8h)

Step Number	Description
24	Enable all "Error Interrupt Signal Enable" bits in Error Interrupt Signal Enable Register (C0h).
25	Write "1" to clear Normal Interrupt Status Register (58h).
26	Enable all 'Normal Interrupt Signal Enable' bits in Normal Interrupt Signal Enable Register (60h).
27	End

40.3.3.3 Transmit or Receive Control Flow in Slave DMA Mode

In Slave DMA mode, Data read or write sequence is same like PIO mode. In addition, the firmware has to program "Slave DMA Configuration" register to define Transfer Direction, Transfer Size and DMA burst size.

Based on the burst size programmed in "Slave DMA Configuration" register, the HSI Controller will assert the DMA requests. The DMA transfer count value will decrement for every 4 Bytes transfers. Once the count reaches zero, the HSI Controller stops asserting DMA request and asserts "Transfer Complete Interrupt", so that CPU can program the external DMA for next data transfers.

For write operation, the DMA transfer count value denotes the number of 4 Bytes data available to transfer, where as for Read Operation, this value denotes the number of 4 Bytes buffer available.

40.4 Register Description

40.4.1 Register Map Summary

- Base Address: 0x1256_0000

Register	Offset	Description	Reset Value
SDMA0_CONF	0x0000	SDMA0 configuration register	0x0000_0000
SDMA1_CONF	0x0004	SDMA1 configuration register	0x0000_0000
SDMA2_CONF	0x0008	SDMA2 configuration register	0x0000_0000
SDMA3_CONF	0x000C	SDMA3 configuration register	0x0000_0000
SDMA4_CONF	0x0010	SDMA4 configuration register	0x0000_0000
SDMA5_CONF	0x0014	SDMA5 configuration register	0x0000_0000
SDMA6_CONF	0x0018	SDMA6 configuration register	0x0000_0000
SDMA7_CONF	0x001C	SDMA7 configuration register	0x0000_0000
RSVD	0x0020 to 0x003C	Reserved for future use.	0x0000_0000
Tx FIFO_CTL_1	0x0040	Tx FIFO control1 register	0x0000_0000
Tx FIFO_CTL_2	0x0044	Tx FIFO control2 register	0x0000_0000
Rx FIFO_CTL_1	0x0048	Rx FIFO control1 register	0x0000_0000
Rx FIFO_CTL_2	0x004C	Rx FIFO control2 register	0x0000_0000
CLK_CTRL	0x0050	Clock control register	0x1800_0000
STATUS	0x0054	HSI status register	0xFF00_0000
INT_STAT	0x0058	Interrupt status register	0x0000_0000
INT_STAT_EN	0x005C	Interrupt status enable register	0x0000_0000
INT_SIG_EN	0x0060	Interrupt signal enable register	0x0000_0000
PROGRAM	0x0064	Program register	0x0FFF_F000
ARBITER_PROG	0x0068	HSI arbiter priority register	0x0000_0000
ARBITER_BW1	0x006C	HSI arbiter bandwidth1	0x0000_0000
ARBITER_BW2	0x0070	HSI arbiter bandwidth2	0x0000_0000
CAPA	0x0074	Capability register	0x0000_07FF
Tx_CH0_DATA	0x0078	Tx channel0 data port register	0x0000_0000
Tx_CH1_DATA	0x007C	Tx channel1 data port register	0x0000_0000
Tx_CH2_DATA	0x0080	Tx channel2 data port register	0x0000_0000
Tx_CH3_DATA	0x0084	Tx channel3 data port register	0x0000_0000
Tx_CH4_DATA	0x0088	Tx channel4 data port register	0x0000_0000
Tx_CH5_DATA	0x008C	Tx channel5 data port register	0x0000_0000
Tx_CH6_DATA	0x0090	Tx channel6 data port register	0x0000_0000
Tx_CH7_DATA	0x0094	Tx channel7 data port register	0x0000_0000
Rx_CH0_DATA	0x0098	Rx channel0 data port register	0x0000_0000

Register	Offset	Description	Reset Value
Rx_CH1_DATA	0x009C	Rx channel1 data port register	0x0000_0000
Rx_CH2_DATA	0x00A0	Rx channel2 data port register	0x0000_0000
Rx_CH3_DATA	0x00A4	Rx channel3 data port register	0x0000_0000
Rx_CH4_DATA	0x00A8	Rx channel4 data port register	0x0000_0000
Rx_CH5_DATA	0x00AC	Rx channel5 data port register	0x0000_0000
Rx_CH6_DATA	0x00B0	Rx channel6 data port register	0x0000_0000
Rx_CH7_DATA	0x00B4	Rx channel7 data port register	0x0000_0000
ERR_INT_STAT	0x00B8	Error interrupt status register	0x0000_0000
ERR_INT_STAT_EN	0x00BC	Error interrupt status enable register	0x0000_0000
ERR_INT_SIG_EN	0x00C0	Error interrupt signal enable register	0x0000_0000
STATUS1	0x00C4	HSI status1 register	0x0000_01FF
PROGRAM1	0x00C8	Program1 register	0x0000_0000
RSVD	0x00CC to 0x00F8	Reserved for future use	–
VER_INFO	0x00FC	Version register	0x0000_1014

40.4.1.1 SDMA0_CONF

- Base Address: 0x1256_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SDMA enable	[31]	RW	This bit enables SDMA0.	0
RSVD	[30:26]	R	Reserved	0000(Must)
DMA burst size	[25:24]	RW	This field denotes the DMA burst size of a single data transfer. b00 = 16 Bytes b01 = 32 Bytes b10 = 64 Bytes b11 = 128 Bytes If the data inside a FIFO is less than the programmed DMA burst size then HSI controller will assert single transfer request. If the request type of PDMA is set to burst transfer then the DMA burst size should be equal to the burst size of PDMA. But if the DMA burst size is 128 Bytes then the request type of PDMA should be set to single transfer.	00
DMA Transfer Count	[23:4]	RW	Total DMA Transfer Count. Every 4 Bytes data transfer will decrease this count. The Slave DMA will initiate the data transaction only if this field is non-zero. The unit is 4 Bytes. h1 = 4 Bytes to transfer h2 = 8 Bytes to transfer … hfffff = 4,194,300 Bytes to transfer.	0x0000
Channel number	[3:1]	RW	This bit reflects which channel is assigned to Slave DMA0. b000 = Channel0 is assigned for Slave DMA0 b001 = Channel1 is assigned for Slave DMA0 b010 = Channel2 is assigned for Slave DMA0 … b111 = Channel7 is assigned for Slave DMA0	000
Sdma_tx_rx	[0]	RW	0 = Slave DMA0 is assigned for Transmit FIFO. Data transfer takes place from external memory to internal Tx FIFO. 1 = Slave DMA0 is assigned for Receive FIFO. Data transfer takes place from internal Rx FIFO to external memory.	0

40.4.1.2 SDMA1_CONF

- Base Address: 0x1256_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SDMA enable	[31]	RW	This bit enables SDMA1.	0
RSVD	[30:27]	R	Reserved	0000
DMA burst size	[26:24]	RW	This field denotes the DMA burst size of a single data transfer. b000 = 16 Bytes b001 = 32 Bytes b010 = 64 Bytes b011 = 128 Bytes b100 to b111 = Reserved for future use If the data inside a FIFO is less than the programmed DMA burst size then HSI controller will assert single transfer request. If the request type of PDMA is set to burst transfer then the DMA burst size should be equal to the burst size of PDMA. But if the DMA burst size is 128 Bytes then the request type of PDMA should be set to single transfer.	000
DMA Transfer Count	[23:4]	RW	Total DMA Transfer Count. Every 4 Bytes data transfer will decrease this count. The Slave DMA will initiate the data transaction only if this field is non-zero. The unit is 4 Bytes. h1 = 4 Bytes to transfer h2 = 8 Bytes to transfer … hfffff = 4,194,300 Bytes to transfer.	0x0000
Channel number	[3:1]	RW	This bit reflects which channel is assigned to Slave DMA1. b000 = Channel0 is assigned for Slave DMA1 b001 = Channel1 is assigned for Slave DMA1 b010 = Channel2 is assigned for Slave DMA1 … b111 = Channel7 is assigned for Slave DMA1	000
Sdma_tx_rx	[0]	RW	0 = Slave DMA1 is assigned for Transmit FIFO. Data transfer takes place from external memory to internal Tx FIFO. 1 = Slave DMA1 is assigned for Receive FIFO. Data transfer takes place from internal Rx FIFO to external memory.	0

40.4.1.3 SDMA2_CONF

- Base Address: 0x1256_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SDMA enable	[31]	RW	This bit enables SDMA2.	0
RSVD	[30:27]	R	Reserved	0000
DMA burst size	[26:24]	RW	This field denotes the DMA burst size of a single data transfer. b000 = 16 Bytes b001 = 32 Bytes b010 = 64 Bytes b011 = 128 Bytes b100 to b111 = Reserved for future use If the data inside a FIFO is less than the programmed DMA burst size then HSI controller will assert single transfer request. If the request type of PDMA is set to burst transfer then the DMA burst size should be equal to the burst size of PDMA. But if the DMA burst size is 128 Bytes then the request type of PDMA should be set to single transfer.	000
DMA Transfer Count	[23:4]	RW	Total DMA Transfer Count. Every 4 Bytes data transfer will decrease this count. The Slave DMA will initiate the data transaction only if this field is non-zero. The unit is 4 Bytes. h1 = 4 Bytes to transfer h2 = 8 Bytes to transfer … hfffff = 4,194,300 Bytes to transfer.	0x0000
Channel number	[3:1]	RW	This bit reflects which channel is assigned to Slave DMA2. b000 = Channel0 is assigned for Slave DMA2 b001 = Channel1 is assigned for Slave DMA2 b010 = Channel2 is assigned for Slave DMA2 … b111 = Channel7 is assigned for Slave DMA2	000
Sdma_tx_rx	[0]	RW	0 = Slave DMA2 is assigned for Transmit FIFO. Data transfer takes place from external memory to internal Tx FIFO. 1 = Slave DMA2 is assigned for Receive FIFO. Data transfer takes place from internal Rx FIFO to external memory.	0

40.4.1.4 SDMA3_CONF

- Base Address: 0x1256_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SDMA enable	[31]	RW	This bit enables SDMA3.	0
RSVD	[30:27]	R	Reserved	0000
DMA burst size	[26:24]	RW	This field denotes the DMA burst size of a single data transfer. b000 = 16 Bytes b001 = 32 Bytes b010 = 64 Bytes b011 = 128 Bytes b100 - b111 = Reserved for future use If the data inside a FIFO is less than the programmed DMA burst size then HSI controller will assert single transfer request. If the request type of PDMA is set to burst transfer then the DMA burst size should be equal to the burst size of PDMA. But if the DMA burst size is 128 Bytes then the request type of PDMA should be set to single transfer.	000
DMA Transfer Count	[23:4]	RW	Total DMA Transfer Count. Every 4 Bytes data transfer will decrease this count. The Slave DMA will initiate the data transaction only if this field is non-zero. The unit is 4 Bytes. 'h1 = 4 Bytes to transfer 'h2 = 8 Bytes to transfer ..... 'hfffff = 4,194,300 Bytes to transfer.	0x0000
Channel number	[3:1]	RW	This bit reflects which channel is assigned to Slave DMA3. b000 = Channel0 is assigned for Slave DMA3 b001 = Channel1 is assigned for Slave DMA3 b010 = Channel2 is assigned for Slave DMA3 .... b111 = Channel7 is assigned for Slave DMA3	000
Sdma_tx_rx	[0]	RW	0 = Slave DMA3 is assigned for Transmit FIFO. Data transfer takes place from external memory to internal Tx FIFO. 1 = Slave DMA3 is assigned for Receive FIFO. Data transfer takes place from internal Rx FIFO to external memory.	0

40.4.1.5 SDMA4_CONF

- Base Address: 0x1256_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SDMA enable	[31]	RW	This bit enables SDMA4.	0
RSVD	[30:27]	R	Reserved	0000
DMA burst size	[26:24]	RW	This field denotes the DMA burst size of a single data transfer. b000 = 16 Bytes b001 = 32 Bytes b010 = 64 Bytes b011 = 128 Bytes b100 - b111 = Reserved for future use If the data inside a FIFO is less than the programmed DMA burst size then HSI controller will assert single transfer request. If the request type of PDMA is set to burst transfer then the DMA burst size should be equal to the burst size of PDMA. But if the DMA burst size is 128 Bytes then the request type of PDMA should be set to single transfer.	000
DMA Transfer Count	[23:4]	RW	Total DMA Transfer Count. Every 4 Bytes data transfer will decrease this count. The Slave DMA will initiate the data transaction only if this field is non-zero. The unit is 4 Bytes. 'h1 = 4 Bytes to transfer 'h2 = 8 Bytes to transfer ..... 'hfffff = 4,194,300 Bytes to transfer.	0x0000
Channel number	[3:1]	RW	This bit reflects which channel is assigned to Slave DMA4. b000 = Channel0 is assigned for Slave DMA4 b001 = Channel1 is assigned for Slave DMA4 b010 = Channel2 is assigned for Slave DMA4 .... b111 = Channel7 is assigned for Slave DMA4	000
Sdma_tx_rx	[0]	RW	0 = Slave DMA4 is assigned for Transmit FIFO. Data transfer takes place from external memory to internal Tx FIFO. 1 = Slave DMA4 is assigned for Receive FIFO. Data transfer takes place from internal Rx FIFO to external memory.	0

40.4.1.6 SDMA5_CONF

- Base Address: 0x1256_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SDMA enable	[31]	RW	This bit enables SDMA5.	0
RSVD	[30:27]	R	Reserved	0000
DMA burst size	[26:24]	RW	This field denotes the DMA burst size of a single data transfer. b000 = 16 Bytes b001 = 32 Bytes b010 = 64 Bytes b011 = 128 Bytes b100 - b111 = Reserved for future use If the data inside a FIFO is less than the programmed DMA burst size then HSI controller will assert single transfer request. If the request type of PDMA is set to burst transfer then the DMA burst size should be equal to the burst size of PDMA. But if the DMA burst size is 128 Bytes then the request type of PDMA should be set to single transfer.	000
DMA Transfer Count	[23:4]	RW	Total DMA Transfer Count. Every 4 Bytes data transfer will decrease this count. The Slave DMA will initiate the data transaction only if this field is non-zero. The unit is 4 Bytes. 'h1 = 4 Bytes to transfer 'h2 = 8 Bytes to transfer ..... 'hfffff = 4,194,300 Bytes to transfer.	0x0000
Channel number	[3:1]	RW	This bit reflects which channel is assigned to Slave DMA5. b000 = Channel0 is assigned for Slave DMA5 b001 = Channel1 is assigned for Slave DMA5 b010 = Channel2 is assigned for Slave DMA5 .... b111 = Channel7 is assigned for Slave DMA5	000
Sdma_tx_rx	[0]	RW	0 = Slave DMA5 is assigned for Transmit FIFO. Data transfer takes place from external memory to internal Tx FIFO. 1 = Slave DMA5 is assigned for Receive FIFO. Data transfer takes place from internal Rx FIFO to external memory.	0

40.4.1.7 SDMA6_CONF

- Base Address: 0x1256_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SDMA enable	[31]	RW	This bit enables SDMA6.	0
RSVD	[30:27]	R	Reserved	0000
DMA burst size	[26:24]	RW	This field denotes the DMA burst size of a single data transfer. b000 = 16 Bytes b001 = 32 Bytes b010 = 64 Bytes b011 = 128 Bytes b100 - b111 = Reserved for future use If the data inside a FIFO is less than the programmed DMA burst size then HSI controller will assert single transfer request. If the request type of PDMA is set to burst transfer then the DMA burst size should be equal to the burst size of PDMA. But if the DMA burst size is 128 Bytes then the request type of PDMA should be set to single transfer.	000
DMA Transfer Count	[23:4]	RW	Total DMA Transfer Count. Every 4 Bytes data transfer will decrease this count. The Slave DMA will initiate the data transaction only if this field is non-zero. The unit is 4 Bytes. 'h1 = 4 Bytes to transfer 'h2 = 8 Bytes to transfer ..... 'hfffff = 4,194,300 Bytes to transfer.	0x0000
Channel number	[3:1]	RW	This bit reflects which channel is assigned to Slave DMA. b000 = Channel0 is assigned for Slave DMA6 b001 = Channel1 is assigned for Slave DMA6 b010 = Channel2 is assigned for Slave DMA6 .... b111 = Channel7 is assigned for Slave DMA6	000
Sdma_tx_rx	[0]	RW	0 = Slave DMA6 is assigned for Transmit FIFO. Data transfer takes place from external memory to internal Tx FIFO. 1 = Slave DMA6 is assigned for Receive FIFO. Data transfer takes place from internal Rx FIFO to external memory.	0

40.4.1.8 SDMA7_CONF

- Base Address: 0x1256_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SDMA enable	[31]	RW	This bit enables SDMA7.	0
RSVD	[30:27]	R	Reserved	0000
DMA burst size	[26:24]	RW	This field denotes the DMA burst size of a single data transfer. b000 = 16 Bytes b001 = 32 Bytes b010 = 64 Bytes b011 = 128 Bytes b100 - b111 = Reserved for future use If the data inside a FIFO is less than the programmed DMA burst size then HSI controller will assert single transfer request. If the request type of PDMA is set to burst transfer then the DMA burst size should be equal to the burst size of PDMA. But if the DMA burst size is 128 Bytes then the request type of PDMA should be set to single transfer.	000
DMA Transfer Count	[23:4]	RW	Total DMA Transfer Count. Every 4 Bytes data transfer will decrease this count. The Slave DMA will initiate the data transaction only if this field is non-zero. The unit is 4 Bytes. 'h1 = 4 Bytes to transfer 'h2 = 8 Bytes to transfer ..... 'hfffff = 4,194,300 Bytes to transfer.	0x0000
Channel number	[3:1]	RW	This bit reflects which channel is assigned to Slave DMA7. b000 = Channel0 is assigned for Slave DMA7 b001 = Channel1 is assigned for Slave DMA7 b010 = Channel2 is assigned for Slave DMA7 .... b111 = Channel7 is assigned for Slave DMA7	000
Sdma_tx_rx	[0]	RW	0 = Slave DMA7 is assigned for Transmit FIFO. Data transfer takes place from external memory to internal Tx FIFO. 1 = Slave DMA7 is assigned for Receive FIFO. Data transfer takes place from internal Rx FIFO to external memory.	0

40.4.1.9 TxFIFO_CTL_1

- Base Address: 0x1256_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Tx Channel7 FIFO size	[31:28]	RW	This field is used to set the buffer size for channel 7. All the allowed combinations of bit setting are listed here. b0000 = Ch7 buffer size is 4 Bytes b0001 = Ch7 buffer size is 8 Bytes b0010 = Ch7 buffer size is 16 Bytes b0011 = Ch7 buffer size is 32 Bytes b0100 = Ch7 buffer size is 64 Bytes b0101 = Ch7 buffer size is 128 Bytes b0110 = Ch7 buffer size is 256 Bytes b0111 = Ch7 buffer size is 512 Bytes b1000 = Ch7 buffer size is 1024 Bytes b1001 = Ch7 buffer size is 2048 Bytes b1010 = Ch7 buffer size is 4096 Bytes b1111 – b1011 = Reserved	0x0
Tx Channel6 FIFO size	[27:24]	RW	This field is used to set the buffer size for channel 6. All the allowed combinations of bit setting are listed here. b0000 = Ch6 buffer size is 4 Bytes b0001 = Ch6 buffer size is 8 Bytes b0010 = Ch6 buffer size is 16 Bytes b0011 = Ch6 buffer size is 32 Bytes b0100 = Ch6 buffer size is 64 Bytes b0101 = Ch6 buffer size is 128 Bytes b0110 = Ch6 buffer size is 256 Bytes b0111 = Ch6 buffer size is 512 Bytes b1000 = Ch6 buffer size is 1024 Bytes b1001 = Ch6 buffer size is 2048 Bytes b1010 = Ch6 buffer size is 4096 Bytes b1111 – b1011 = Reserved	0x0
Tx Channel5 FIFO size	[23:20]	RW	This field is used to set the buffer size for channel 5. All the allowed combinations of bit setting are listed here. b0000 = Ch5 buffer size is 4 Bytes b0001 = Ch5 buffer size is 8 Bytes b0010 = Ch5 buffer size is 16 Bytes b0011 = Ch5 buffer size is 32 Bytes b0100 = Ch5 buffer size is 64 Bytes b0101 = Ch5 buffer size is 128 Bytes b0110 = Ch5 buffer size is 256 Bytes b0111 = Ch5 buffer size is 512 Bytes b1000 = Ch5 buffer size is 1024 Bytes b1001 = Ch5 buffer size is 2048 Bytes b1010 = Ch5 buffer size is 4096 Bytes b1111 – b1011 = Reserved	0x0
Tx Channel4 FIFO size	[19:16]	RW	This field is used to set the buffer size for channel 4. All the allowed combinations of bit setting are listed here. b0000 = Ch4 buffer size is 4 Bytes	0x0

Name	Bit	Type	Description	Reset Value
			b0001 = Ch4 buffer size is 8 Bytes b0010 = Ch4 buffer size is 16 Bytes b0011 = Ch4 buffer size is 32 Bytes b0100 = Ch4 buffer size is 64 Bytes b0101 = Ch4 buffer size is 128 Bytes b0110 = Ch4 buffer size is 256 Bytes b0111 = Ch4 buffer size is 512 Bytes b1000 = Ch4 buffer size is 1024 Bytes b1001 = Ch4 buffer size is 2048 Bytes b1010 = Ch4 buffer size is 4096 Bytes b1111 – b1011 = Reserved	
Tx Channel3 FIFO size	[15:12]	RW	This field is used to set the buffer size for channel 3. All the allowed combinations of bit setting are listed here. b0000 = Ch3 buffer size is 4 Bytes b0001 = Ch3 buffer size is 8 Bytes b0010 = Ch3 buffer size is 16 Bytes b0011 = Ch3 buffer size is 32 Bytes b0100 = Ch3 buffer size is 64 Bytes b0101 = Ch3 buffer size is 128 Bytes b0110 = Ch3 buffer size is 256 Bytes b0111 = Ch3 buffer size is 512 Bytes b1000 = Ch3 buffer size is 1024 Bytes b1001 = Ch3 buffer size is 2048 Bytes b1010 = Ch3 buffer size is 4096 Bytes b1111 – b1011 = Reserved	0x0
Tx Channel2 FIFO size	[11:8]	RW	This field is used to set the buffer size for channel 2. All the allowed combinations of bit setting are listed here. b0000 = Ch2 buffer size is 4 Bytes b0001 = Ch2 buffer size is 8 Bytes b0010 = Ch2 buffer size is 16 Bytes b0011 = Ch2 buffer size is 32 Bytes b0100 = Ch2 buffer size is 64 Bytes b0101 = Ch2 buffer size is 128 Bytes b0110 = Ch2 buffer size is 256 Bytes b0111 = Ch2 buffer size is 512 Bytes b1000 = Ch2 buffer size is 1024 Bytes b1001 = Ch2 buffer size is 2048 Bytes b1010 = Ch2 buffer size is 4096 Bytes b1111 – b1011 = Reserved	0x0
Tx Channel1 FIFO size	[7:4]	RW	This field is used to set the buffer size for channel 1. All the allowed combinations of bit setting are listed here. b0000 = Ch1 buffer size is 4 Bytes b0001 = Ch1 buffer size is 8 Bytes b0010 = Ch1 buffer size is 16 Bytes b0011 = Ch1 buffer size is 32 Bytes b0100 = Ch1 buffer size is 64 Bytes b0101 = Ch1 buffer size is 128 Bytes b0110 = Ch1 buffer size is 256 Bytes b0111 = Ch1 buffer size is 512 Bytes b1000 = Ch1 buffer size is 1024 Bytes	0x0

Name	Bit	Type	Description	Reset Value
			b1001 = Ch1 buffer size is 2048 Bytes b1010 = Ch1 buffer size is 4096 Bytes b1111 – b1011 = Reserved	
Tx Channel0 FIFO size	[3:0]	RW	This field is used to set the buffer size for channel 0. All the allowed combinations of bit setting are listed here. b0000 = Ch0 buffer size is 4 Bytes b0001 = Ch0 buffer size is 8 Bytes b0010 = Ch0 buffer size is 16 Bytes b0011 = Ch0 buffer size is 32 Bytes b0100 = Ch0 buffer size is 64 Bytes b0101 = Ch0 buffer size is 128 Bytes b0110 = Ch0 buffer size is 256 Bytes b0111 = Ch0 buffer size is 512 Bytes b1000 = Ch0 buffer size is 1024 Bytes b1001 = Ch0 buffer size is 2048 Bytes b1010 = Ch0 buffer size is 4096 Bytes b1111 – b1011 = Reserved	0x0

40.4.1.10 TxFIFO_CTL_2

- Base Address: 0x1256_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	R	Reserved for future use.	0x000_0000
Tx Channel7 FIFO Threshold	[7]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
Tx Channel6 FIFO Threshold	[6]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
Tx Channel5 FIFO Threshold	[5]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
Tx Channel4 FIFO Threshold	[4]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
Tx Channel3 FIFO Threshold	[3]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
Tx Channel2 FIFO Threshold	[2]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
Tx Channel1 FIFO Threshold	[1]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
Tx Channel0 FIFO Threshold	[0]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0

40.4.1.11 RXFIFO_CTL_1

- Base Address: 0x1256_0000
- Address = Base Address + 0x0048, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RX Channel7 FIFO size	[31:28]	RW	This field is used to set the buffer size for channel 7. All the allowed combinations of bit setting are listed here. b0000 = Ch7 buffer size is 4 Bytes b0001 = Ch7 buffer size is 8 Bytes b0010 = Ch7 buffer size is 16 Bytes b0011 = Ch7 buffer size is 32 Bytes b0100 = Ch7 buffer size is 64 Bytes b0101 = Ch7 buffer size is 128 Bytes b0110 = Ch7 buffer size is 256 Bytes b0111 = Ch7 buffer size is 512 Bytes b1000 = Ch7 buffer size is 1024 Bytes b1001 = Ch7 buffer size is 2048 Bytes b1010 = Ch7 buffer size is 4096 Bytes b1111 – b1011 = Reserved	0x0
RX Channel6 FIFO size	[27:24]	RW	This field is used to set the buffer size for channel 6. All the allowed combinations of bit setting are listed here. b0000 = Ch6 buffer size is 4 Bytes b0001 = Ch6 buffer size is 8 Bytes b0010 = Ch6 buffer size is 16 Bytes b0011 = Ch6 buffer size is 32 Bytes b0100 = Ch6 buffer size is 64 Bytes b0101 = Ch6 buffer size is 128 Bytes b0110 = Ch6 buffer size is 256 Bytes b0111 = Ch6 buffer size is 512 Bytes b1000 = Ch6 buffer size is 1024 Bytes b1001 = Ch6 buffer size is 2048 Bytes b1010 = Ch6 buffer size is 4096 Bytes b1111 – b1011 = Reserved	0x0
RX Channel5 FIFO size	[23:20]	RW	This field is used to set the buffer size for channel 5. All the allowed combinations of bit setting are listed here. b0000 = Ch5 buffer size is 4 Bytes b0001 = Ch5 buffer size is 8 Bytes b0010 = Ch5 buffer size is 16 Bytes b0011 = Ch5 buffer size is 32 Bytes b0100 = Ch5 buffer size is 64 Bytes b0101 = Ch5 buffer size is 128 Bytes b0110 = Ch5 buffer size is 256 Bytes b0111 = Ch5 buffer size is 512 Bytes b1000 = Ch5 buffer size is 1024 Bytes b1001 = Ch5 buffer size is 2048 Bytes b1010 = Ch5 buffer size is 4096 Bytes b1111 – b1011 = Reserved	0x0
RX Channel4 FIFO size	[19:16]	RW	This field is used to set the buffer size for channel 4. All the allowed combinations of bit setting are listed here. b0000 = Ch4 buffer size is 4 Bytes	0x0

Name	Bit	Type	Description	Reset Value
			b0001 = Ch4 buffer size is 8 Bytes b0010 = Ch4 buffer size is 16 Bytes b0011 = Ch4 buffer size is 32 Bytes b0100 = Ch4 buffer size is 64 Bytes b0101 = Ch4 buffer size is 128 Bytes b0110 = Ch4 buffer size is 256 Bytes b0111 = Ch4 buffer size is 512 Bytes b1000 = Ch4 buffer size is 1024 Bytes b1001 = Ch4 buffer size is 2048 Bytes b1010 = Ch4 buffer size is 4096 Bytes b1111 – b1011 = Reserved	
RX Channel3 FIFO size	[15:12]	RW	This field is used to set the buffer size for channel 3. All the allowed combinations of bit setting are listed here. b0000 = Ch3 buffer size is 4 Bytes b0001 = Ch3 buffer size is 8 Bytes b0010 = Ch3 buffer size is 16 Bytes b0011 = Ch3 buffer size is 32 Bytes b0100 = Ch3 buffer size is 64 Bytes b0101 = Ch3 buffer size is 128 Bytes b0110 = Ch3 buffer size is 256 Bytes b0111 = Ch3 buffer size is 512 Bytes b1000 = Ch3 buffer size is 1024 Bytes b1001 = Ch3 buffer size is 2048 Bytes b1010 = Ch3 buffer size is 4096 Bytes b1111 – b1011 = Reserved	0x0
RX Channel2 FIFO size	[11:8]	RW	This field is used to set the buffer size for channel 2. All the allowed combinations of bit setting are listed here. b0000 = Ch2 buffer size is 4 Bytes b0001 = Ch2 buffer size is 8 Bytes b0010 = Ch2 buffer size is 16 Bytes b0011 = Ch2 buffer size is 32 Bytes b0100 = Ch2 buffer size is 64 Bytes b0101 = Ch2 buffer size is 128 Bytes b0110 = Ch2 buffer size is 256 Bytes b0111 = Ch2 buffer size is 512 Bytes b1000 = Ch2 buffer size is 1024 Bytes b1001 = Ch2 buffer size is 2048 Bytes b1010 = Ch2 buffer size is 4096 Bytes b1111 – b1011 = Reserved	0x0
RX Channel1 FIFO size	[7:4]	RW	This field is used to set the buffer size for channel 1. All the allowed combinations of bit setting are listed here. b0000 = Ch1 buffer size is 4 Bytes b0001 = Ch1 buffer size is 8 Bytes b0010 = Ch1 buffer size is 16 Bytes b0011 = Ch1 buffer size is 32 Bytes b0100 = Ch1 buffer size is 64 Bytes b0101 = Ch1 buffer size is 128 Bytes b0110 = Ch1 buffer size is 256 Bytes b0111 = Ch1 buffer size is 512 Bytes b1000 = Ch1 buffer size is 1024 Bytes	0x0

Name	Bit	Type	Description	Reset Value
			b1001 = Ch1 buffer size is 2048 Bytes b1010 = Ch1 buffer size is 4096 Bytes b1111 – b1011 = Reserved	
RX Channel0 FIFO size	[3:0]	RW	This field is used to set the buffer size for channel 0. All the allowed combinations of bit setting are listed here. b0000 = Ch0 buffer size is 4 Bytes b0001 = Ch0 buffer size is 8 Bytes b0010 = Ch0 buffer size is 16 Bytes b0011 = Ch0 buffer size is 32 Bytes b0100 = Ch0 buffer size is 64 Bytes b0101 = Ch0 buffer size is 128 Bytes b0110 = Ch0 buffer size is 256 Bytes b0111 = Ch0 buffer size is 512 Bytes b1000 = Ch0 buffer size is 1024 Bytes b1001 = Ch0 buffer size is 2048 Bytes b1010 = Ch0 buffer size is 4096 Bytes b1111 – b1011 = Reserved	0x0

40.4.1.12 RXFIFO_CTL_2

- Base Address: 0x1256_0000
- Address = Base Address + 0x004C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	R	Reserved for future use.	0x000_0000
RX Channel7 FIFO Threshold	[7]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
RX Channel6 FIFO Threshold	[6]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
RX Channel5 FIFO Threshold	[5]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
RX Channel4 FIFO Threshold	[4]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
RX Channel3 FIFO Threshold	[3]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
RX Channel2 FIFO Threshold	[2]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
RX Channel1 FIFO Threshold	[1]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0
RX Channel0 FIFO Threshold	[0]	RW	0 = Half Empty (FIFO size/2) 1 = Almost Empty (FIFO size/4)	0

40.4.1.13 CLK_CTRL

- Base Address: 0x1256_0000
- Address = Base Address + 0x0050, Reset Value = 0x1800_0000
- Tap delay value for RX_CLK should be changed to 0x2 from 0x7.(Must)

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	R	Reserved for future use.	00
Rx clk buf select	[29:27]	RW	d0 - 0.0ps d1 - 2.2ns d2 - 4.4ns d3 - 12.2ns d4 - 20.0ns d5 - 27.8ns d6 - 35.6ns d7 - 43.4ns These values denote the tap delay values for reception of data & flag	0x3
Rx Tailing Bit counter	[26:24]	RW	d0 = 400 d1 = 200 d2 = 100 d4 = 50 The value determines the length of the Tailing bit counter. The receiver shall start Receiver Tailing- bit counter after the nth frame programmed in Rx Frame Burst counter is received. The receiver shall then drive ready to logic one if the receiver Tailing-bit counter has completed with no errors detected, and the receiver has enough room for at least one new frame	0x0
Rx Frame Burst count	[23:16]	RW	d0 = Frame transmission count is set to 256 frames. d1 = Frame transmission count is set to 1 frame. d2 = Frame transmission count is set to 2 frames. ..... d255 = Frame transmission count is set to 255 frames. This value is to limit the continous Frame transmission count in Pipelined Data flow. The Receiver Frame Burst counter shall be able to support upto 256 frames of continous transfer.	0x00
Tx break	[15]	RW	Seeting this bit to one trigger a transmission break at HSI Tx. Once this bit is set to one, the HSI controller will send zeros on "tx_data" port. It will stop sending zeros once this bit gets to set to zero. 0 = HSI Tx not in transmission break mode.	0

Name	Bit	Type	Description	Reset Value
			1 = HSI Tx is in break mode.	
Data timeout counter value	[14:11]	RW	b0000 = HSI Tx Clock x 2 pow 13 b0001 = HSI Tx Clock x 2 pow 14 … b1110 = HSI Tx Clock x 2 pow 27 This value determines the interval by which DATA timeouts are detected. This data timeout counter logic is used only for Receive operations. The counter should start counting when data in any of the Rx channel FIFO is less than the threshold value and resets to zero when there is a threshold reached interrupt from any of the Rx buffers. The counter value should be zero, when Rx FIFO is empty. An interrupt will be asserted to the host driver, when the counter value reaches the data timeout counter value.	0000
Clock_divisor	[10:3]	RW	This register holds the divisor of the base clock (clk_tx_base) frequency for HSI Tx clock (internal clock used to drive Transmitter interface). h00 = Base clock divided by 1 h01 = Base clock divided by 2 h02 = Base clock divided by 4 h04 = Base clock divided by 8 h08 = Base clock divided by 16 h10 = Base clock divided by 32 h20 = Base clock divided by 64 h40 = Base clock divided by 128 h80 = Base clock divided by 256	0x00
Internal_clk_ enable	[2]	RW	Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, the HSI core shall set clk_ stable in this register to 1. 0 = Stop 1 = Oscillate	0
Clk_stable	[1]	R	This bit gets set to 1 when HSI Tx clock is stable for a programmed clock_divisor.	0
HSI_CLK_tx_ enable	[0]	RW	This bit is used to enable the tx clock. Firmware should set this bit to one only when clk_stable is one.	0

40.4.1.14 STATUS

- Base Address: 0x1256_0000
- Address = Base Address + 0x0054, Reset Value = 0xFF00_0000

Name	Bit	Type	Description	Reset Value
Tx Channel7 FIFO Empty	[31]	R	1 = Tx channel 7 FIFO is Empty. 0 = Tx channel 7 FIFO is not Empty.	1
Tx Channel6 FIFO Empty	[30]	R	1 = Tx channel 6 FIFO is Empty. 0 = Tx channel 6 FIFO is not Empty.	1
Tx Channel5 FIFO Empty	[29]	R	1 = Tx channel 5 FIFO is Empty. 0 = Tx channel 5 FIFO is not Empty.	1
Tx Channel4 FIFO Empty	[28]	R	1 = Tx channel 4 FIFO is Empty. 0 = Tx channel 4 FIFO is not Empty.	1
Tx Channel3 FIFO Empty	[27]	R	1 = Tx channel 3 FIFO is Empty. 0 = Tx channel 3 FIFO is not Empty.	1
Tx Channel2 FIFO Empty	[26]	R	1 = Tx channel 2 FIFO is Empty. 0 = Tx channel 2 FIFO is not Empty.	1
Tx Channel1 FIFO Empty	[25]	R	1 = Tx channel 1 FIFO is Empty. 0 = Tx channel 1 FIFO is not Empty.	1
Tx Channel0 FIFO Empty	[24]	R	1 = Tx channel 0 FIFO is Empty. 0 = Tx channel 0 FIFO is not Empty.	1
Tx Channel7 FIFO Full	[23]	R	1 = Tx channel 7 FIFO is Full. 0 = Tx channel 7 FIFO is not Full.	0
Tx Channel6 FIFO Full	[22]	R	1 = Tx channel 6 FIFO is Full. 0 = Tx channel 6 FIFO is not Full.	0
Tx Channel5 FIFO Full	[21]	R	1 = Tx channel 5 FIFO is Full. 0 = Tx channel 5 FIFO is not Full.	0
Tx Channel4 FIFO Full	[20]	R	1 = Tx channel 4 FIFO is Full. 0 = Tx channel 4 FIFO is not Full.	0
Tx Channel3 FIFO Full	[19]	R	1 = Tx channel 3 FIFO is Full. 0 = Tx channel 3 FIFO is not Full.	0
Tx Channel2 FIFO Full	[18]	R	1 = Tx channel 2 FIFO is Full. 0 = Tx channel 2 FIFO is not Full.	0
Tx Channel1 FIFO Full	[17]	R	1 = Tx channel 1 FIFO is Full. 0 = Tx channel 1 FIFO is not Full.	0
Tx Channel0 FIFO Full	[16]	R	1 = Tx channel 0 FIFO is Full. 0 = Tx channel 0 FIFO is not Full.	0
Rx Channel7 FIFO Not empty	[15]	R	1 = Rx channel 7 FIFO is not empty. 0 = Rx channel 7 FIFO is empty.	0
Rx Channel6 FIFO Not empty	[14]	R	1 = Rx channel 6 FIFO is not empty. 0 = Rx channel 6 FIFO is empty.	0

Name	Bit	Type	Description	Reset Value
Rx Channel5 FIFO Not empty	[13]	R	1 = Rx channel 5 FIFO is not empty. 0 = Rx channel 5 FIFO is empty.	0
Rx Channel4 FIFO Not empty	[12]	R	1 = Rx channel 4 FIFO is not empty. 0 = Rx channel 4 FIFO is empty.	0
Rx Channel3 FIFO Not empty	[11]	R	1 = Rx channel 3 FIFO is not empty. 0 = Rx channel 3 FIFO is empty.	0
Rx Channel2 FIFO Not empty	[10]	R	1 = Rx channel 2 FIFO is not empty. 0 = Rx channel 2 FIFO is empty.	0
Rx Channel1 FIFO Not empty	[9]	R	1 = Rx channel 1 FIFO is not empty. 0 = Rx channel 1 FIFO is empty.	0
Rx Channel0 FIFO Not empty	[8]	R	1 = Rx channel 0 FIFO is not empty. 0 = Rx channel 0 FIFO is empty.	0
Rx_rdy	[7]	R	This field reflects the rx_rdy pin (for debug purpose).	0
Rx_flag	[6]	R	This field reflects the rx_flag pin (for debug purpose).	0
Rx_data	[5]	R	This field reflects the rx_data pin (for debug purpose).	0
Rx_wake	[4]	R	This field reflects the rx_wake pin (for debug purpose).	0
Tx_rdy	[3]	R	This field reflects the tx_rdy pin (for debug purpose).	0
Tx_flag	[2]	R	This field reflects the tx_flag pin (for debug purpose).	0
Tx_data	[1]	R	This field reflects the tx_data pin (for debug purpose).	0
Tx_wake	[0]	R	This field reflects the tx_wake pin (for debug purpose).	0

40.4.1.15 INT_STAT

- Base Address: 0x1256_0000
- Address = Base Address + 0x0058, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Error Interrupt	[31]	RW1C	If any bit in the Error Interrupt Status Register is set, then this bit is set. On seeing this bit set, the ahb driver will read the Error Interrupt Staus Register. 0 = No Error. 1 = Error	0
RSVD	[30:25]	R	Reserved for future use.	0
Transfer Complete Interrupt sdma7	[24]	RW1C	This bit is set when a Transmit or Receive Operation is completed for sdma7. This bit is used only for Slave DMA.	0
Transfer Complete Interrupt sdma6	[23]	RW1C	This bit is set when a Transmit or Receive Operation is completed for sdma6. This bit is used only for Slave DMA.	0
Transfer Complete Interrupt sdma5	[22]	RW1C	This bit is set when a Transmit or Receive Operation is completed for sdma5. This bit is used only for Slave DMA.	0
Transfer Complete Interrupt sdma4	[21]	RW1C	This bit is set when a Transmit or Receive Operation is completed for sdma4. This bit is used only for Slave DMA.	0
Transfer Complete Interrupt sdma3	[20]	RW1C	This bit is set when a Transmit or Receive Operation is completed for sdma3. This bit is used only for Slave DMA.	0
Transfer Complete Interrupt sdma2	[19]	RW1C	This bit is set when a Transmit or Receive Operation is completed for sdma2. This bit is used only for Slave DMA.	0
Transfer Complete Interrupt sdma1	[18]	RW1C	This bit is set when a Transmit or Receive Operation is completed for sdma1. This bit is used only for Slave DMA.	0
Transfer Complete Interrupt sdma0	[17]	RW1C	This bit is set when a Transmit or Receive Operation is completed for sdma0. This bit is used only for Slave DMA.	0
Rx WKUP Interrupt	[16]	RW1C	0 = Receiver Wakeup event is not occurred 1 = Receiver Wakeup event is occurred	0
Rx Channel7 Threshold Reached	[15]	RW1C	0 = Threshold amount of data not reached in Rx Ch7 FIFO 1 = Threshold amount of data reached in Rx Ch7 FIFO	0
Rx Channel6 Threshold Reached	[14]	RW1C	0 = Threshold amount of data not reached in Rx Ch6 FIFO 1 = Threshold amount of data reached in Rx Ch6 FIFO	0

Name	Bit	Type	Description	Reset Value
Rx Channel5 Threshold Reached	[13]	RW1C	0 = Threshold amount of data not reached in Rx Ch5 FIFO 1 = Threshold amount of data reached in Rx Ch5 FIFO	0
Rx Channel4 Threshold Reached	[12]	RW1C	0 = Threshold amount of data not reached in Rx Ch4 FIFO 1 = Threshold amount of data reached in Rx Ch4 FIFO	0
Rx Channel3 Threshold Reached	[11]	RW1C	0 = Threshold amount of data not reached in Rx Ch3 FIFO 1 = Threshold amount of data reached in Rx Ch3 FIFO	0
Rx Channel2 Threshold Reached	[10]	RW1C	0 = Threshold amount of data not reached in Rx Ch2 FIFO 1 = Threshold amount of data reached in Rx Ch2 FIFO	0
Rx Channel1 Threshold Reached	[9]	RW1C	0 = Threshold amount of data not reached in Rx Ch1 FIFO 1 = Threshold amount of data reached in Rx Ch1 FIFO	0
Rx Channel0 Threshold Reached	[8]	RW1C	0 = Threshold amount of data not reached in Rx Ch0 FIFO 1 = Threshold amount of data reached in Rx Ch0 FIFO	0
Tx Channel7 Threshold Reached	[7]	RW1C	0 = Indicates that Tx channel 7 FIFO does not have threshold amount of space to accept the data. 1 = Indicates that Tx channel 7 FIFO has threshold amount of space to accept the data	0
Tx Channel6 Threshold Reached	[6]	RW1C	0 = Indicates that Tx channel 6 FIFO does not have threshold amount of space to accept the data 1 = Indicates that Tx channel 6 FIFO has threshold amount of space to accept the data	0
Tx Channel5 Threshold Reached	[5]	RW1C	0 = Indicates that Tx channel 5 FIFO does not have threshold amount of space to accept the data 1 = Indicates that Tx channel 5 FIFO has threshold amount of space to accept the data	0
Tx Channel4 Threshold Reached	[4]	RW1C	0 = Indicates that Tx channel 4 FIFO does not have threshold amount of space to accept the data 1 = Indicates that Tx channel 4 FIFO has threshold amount of space to accept the data	0
Tx Channel3 Threshold Reached	[3]	RW1C	0 = Indicates that Tx channel 3 FIFO does not have threshold amount of space to accept the data 1 = Indicates that Tx channel 3 FIFO has threshold amount of space to accept the data	0
Tx Channel2	[2]	RW1C	0 = Indicates that Tx channel 2 FIFO does not have	0

Name	Bit	Type	Description	Reset Value
Threshold Reached			threshold amount of space to accept the data 1 = Indicates that Tx channel 2 FIFO has threshold amount of space to accept the data	
Tx Channel1 Threshold Reached	[1]	RW1C	0 = Indicates that Tx channel 1 FIFO does not have threshold amount of space to accept the data 1 = Indicates that Tx channel 1 FIFO has threshold amount of space to accept the data	0
Tx Channel0 Threshold Reached	[0]	RW1C	0 = Indicates that Tx channel 0 FIFO does not have threshold amount of space to accept the data 1 = Indicates that Tx channel 0 FIFO has threshold amount of space to accept the data	0

40.4.1.16 INT_STAT_EN

- Base Address: 0x1256_0000
- Address = Base Address + 0x005C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Fixed to 0	[31]	R	The HSI controller shall control error Interrupts using the Error Interrupt Status Enable register	0
RSVD	[30:25]	R	Reserved for future use.	0
Transfer Complete Interrupt sdma7	[24]	RW	0 = Interrupt status masked for "Transfer complete sdma7" interrupt 1 = Interrupt status enabled for "Transfer complete sdma7" interrupt.	0
Transfer Complete Interrupt sdma6	[23]	RW	0 = Interrupt status masked for "Transfer complete sdma6" interrupt 1 = Interrupt status enabled for "Transfer complete sdma6" interrupt.	0
Transfer Complete Interrupt sdma5	[22]	RW	0 = Interrupt status masked for "Transfer complete sdma5" interrupt 1 = Interrupt status enabled for "Transfer complete sdma5" interrupt.	0
Transfer Complete Interrupt sdma4	[21]	RW	0 = Interrupt status masked for "Transfer complete sdma4" interrupt 1 = Interrupt status enabled for "Transfer complete sdma4" interrupt.	0
Transfer Com-plete Interrupt sdma3	[20]	RW	0 = Interrupt status masked for "Transfer complete sdma3" interrupt 1 = Interrupt status enabled for "Transfer complete sdma3" interrupt.	0
Transfer Complete Interrupt sdma2	[19]	RW	0 = Interrupt status masked for "Transfer complete sdma2" interrupt 1 = Interrupt status enabled for "Transfer complete sdma2" interrupt.	0
Transfer Complete Interrupt sdma1	[18]	RW	0 = Interrupt status masked for "Transfer complete sdma1" interrupt 1 = Interrupt status enabled for "Transfer complete sdma1" interrupt.	0
Transfer Complete Interrupt sdma0	[17]	RW	0 = Interrupt status masked for "Transfer complete sdma0" interrupt 1 = Interrupt status enabled for "Transfer complete sdma0" interrupt.	0
Rx WKUP Interrupt	[16]	RW	0 = Interrupt status masked for "Rx Wakeup" interrupt. 1 = Interrupt status enabled for "Rx Wakeup" interrupt.	0
Rx Channel7	[15]	RW	0 = Interrupt status masked for "Rx channel 7	0

Name	Bit	Type	Description	Reset Value
Threshold Reached			threshold Reached" interrupt. 1 = Interrupt status enabled for "Rx channel 7 threshold Reached" interrupt.	
Rx Channel6 Threshold Reached	[14]	RW	0 = Interrupt status masked for "Rx channel 6 threshold Reached" interrupt 1 = Interrupt status enabled for "Rx channel 6 threshold Reached" interrupt.	0
Rx Channel5 Threshold Reached	[13]	RW	0 = Interrupt status masked for "Rx channel 5 threshold Reached" interrupt 1 = Interrupt status enabled for "Rx channel 5 threshold Reached" interrupt.	0
Rx Channel4 Threshold Reached	[12]	RW	0 = Interrupt status masked for "Rx channel 4 threshold Reached" interrupt 1 = Interrupt status enabled for "Rx channel 4 threshold Reached" interrupt.	0
Rx Channel3 Threshold Reached	[11]	RW	0 = Interrupt status masked for "Rx channel 3 threshold Reached" interrupt 1 = Interrupt status enabled for "Rx channel 3 threshold Reached" interrupt.	0
Rx Channel2 Threshold Reached	[10]	RW	1 = Interrupt status enabled for "Rx channel 2 threshold Reached" interrupt. 0 = Interrupt status masked for "Rx channel 2 threshold Reached" interrupt	0
Rx Channel1 Threshold Reached	[9]	RW	0 = Interrupt status masked for "Rx channel 1 threshold Reached" interrupt 1 = Interrupt status enabled for "Rx channel 1 threshold Reached" interrupt.	0
Rx Channel0 Threshold Reached	[8]	RW	0 = Interrupt status masked for "Rx channel 0 threshold Reached" interrupt 1 = Interrupt status enabled for "Rx channel 0 threshold Reached" interrupt.	0
Tx Channel7 Threshold Reached	[7]	RW	0 = Interrupt status masked for "Tx channel 7 threshold Reached" interrupt. 1 = Interrupt status enabled for "Tx channel 7 threshold Reached" interrupt.	0
Tx Channel6 Threshold Reached	[6]	RW	0 = Interrupt status masked for "Tx channel 6 threshold Reached" interrupt 1 = Interrupt status enabled for "Tx channel 6 threshold Reached" interrupt.	0
Tx Channel5 Threshold Reached	[5]	RW	0 = Interrupt status masked for "Tx channel 5 threshold Reached" interrupt 1 = Interrupt status enabled for "Tx channel 5 threshold Reached" interrupt.	0
Tx Channel4 Threshold	[4]	RW	0 = Interrupt status masked for "Tx channel 4 threshold Reached" interrupt	0

Name	Bit	Type	Description	Reset Value
Reached			1 = Interrupt status enabled for "Tx channel 4 threshold Reached" interrupt.	
Tx Channel3 Threshold Reached	[3]	RW	0 = Interrupt status masked for "Tx channel 3 threshold Reached" interrupt 1 = Interrupt status enabled for "Tx channel 3 threshold Reached" interrupt.	0
Tx Channel2 Threshold Reached	[2]	RW	0 = Interrupt status masked for "Tx channel 2 threshold Reached" interrupt 1 = Interrupt status enabled for "Tx channel 2 threshold Reached" interrupt.	0
Tx Channel1 Threshold Reached	[1]	RW	0 = Interrupt status masked for "Tx channel 1 threshold Reached" interrupt 1 = Interrupt status enabled for "Tx channel 1 threshold Reached" interrupt.	0
Tx Channel0 Threshold Reached	[0]	RW	0 = Interrupt status masked for "Tx channel 0 threshold Reached" interrupt 1 = Interrupt status enabled for "Tx channel 0 threshold Reached" interrupt.	0

40.4.1.17 INT_SIG_EN

- Base Address: 0x1256_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Fixed to 0	[31]	R	The HSI controller shall control error Interrupts using the Error Interrupt Signal Enable register.	0
RSVD	[30:25]	R	Reserved for future use.	0
Transfer Complete Interrupt sdma7	[24]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Transfer complete sdma7" interrupt. 1 = Interrupt signal enabled for "Transfer complete sdma7" interrupt.	0
Transfer Complete Interrupt sdma6	[23]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Transfer complete sdma6" interrupt. 1 = Interrupt signal enabled for "Transfer complete sdma6" interrupt.	0
Transfer Complete Interrupt sdma5	[22]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Transfer complete sdma5" interrupt. 1 = Interrupt signal enabled for "Transfer complete sdma5" interrupt.	0
Transfer Complete Interrupt sdma4	[21]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Transfer complete sdma4" interrupt. 1 = Interrupt signal enabled for "Transfer complete sdma4" interrupt.	0
Transfer Complete Interrupt sdma3	[20]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Transfer complete sdma3" interrupt. 1 = Interrupt signal enabled for "Transfer complete sdma3" interrupt.	0
Transfer Complete Interrupt sdma2	[19]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Transfer complete sdma2" interrupt. 1 = Interrupt signal enabled for "Transfer complete sdma2" interrupt.	0
Transfer Complete Interrupt	[18]	RW	Setting this bit will enable interrupt generation on interrupt line.	0

Name	Bit	Type	Description	Reset Value
sdma1			0 = Interrupt signal masked for "Transfer complete sdma1" interrupt. 1 = Interrupt signal enabled for "Transfer complete sdma1" interrupt.	
Transfer Complete Interrupt sdma0	[17]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Transfer complete sdma0" interrupt. 1 = Interrupt signal enabled for "Transfer complete sdma0" interrupt.	0
Rx WKUP Interrupt	[16]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Rx Wakeup" interrupt. 1 = Interrupt signal enabled for "Rx Wakeup" interrupt.	0
Rx Channel7 Threshold Reached	[15]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Rx channel 7 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Rx channel 7 threshold Reached" interrupt.	0
Rx Channel6 Threshold Reached	[14]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Rx channel 6 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Rx channel 6 threshold Reached" interrupt.	0
Rx Channel5 Threshold Reached	[13]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Rx channel 5 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Rx channel 5 threshold Reached" interrupt.	0
Rx Channel4 Threshold Reached	[12]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Rx channel 4 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Rx channel 4 threshold Reached" interrupt.	0
Rx Channel3 Threshold Reached	[11]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Rx channel 3 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Rx channel 3 threshold Reached" interrupt.	0
Rx Channel2 Threshold	[10]	RW	Setting this bit will enable interrupt generation on	0

Name	Bit	Type	Description	Reset Value
Reached			interrupt line. 0 = Interrupt signal masked for "Rx channel 2 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Rx channel 2 threshold Reached" interrupt.	
Rx Channel1 Threshold Reached	[9]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Rx channel 1 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Rx channel 1 threshold Reached" interrupt.	0
Rx Channel0 Threshold Reached	[8]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Rx channel 0 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Rx channel 0 threshold Reached" interrupt.	0
Tx Channel7 Threshold Reached	[7]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Tx channel 7 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Tx channel 7 threshold Reached" interrupt.	0
Tx Channel6 Threshold Reached	[6]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Tx channel 6 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Tx channel 6 threshold Reached" interrupt.	0
Tx Channel5 Threshold Reached	[5]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Tx channel 5 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Tx channel 5 threshold Reached" interrupt.	0
Tx Channel4 Threshold Reached	[4]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Tx channel 4 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Tx channel 4 threshold Reached" interrupt.	0
Tx Channel3 Threshold Reached	[3]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Tx channel 3 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Tx channel 3	0

Name	Bit	Type	Description	Reset Value
			threshold Reached" interrupt.	
Tx Channel2 Threshold Reached	[2]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Tx channel 2 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Tx channel 2 threshold Reached" interrupt.	0
Tx Channel1 Threshold Reached	[1]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Tx channel 1 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Tx channel 1 threshold Reached" interrupt.	0
Tx Channel0 Threshold Reached	[0]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Tx channel 0 threshold Reached" interrupt. 1 = Interrupt signal enabled for "Tx channel 0 threshold Reached" interrupt.	0

40.4.1.18 PROGRAM

- Base Address: 0x1256_0000
- Address = Base Address + 0x0064, Reset Value = 0x0FFF_0000

Name	Bit	Type	Description	Reset Value
Tx wakeup	[31]	RW	0 = Transmitter is in Sleep State 1 = Transmitter is in Wakeup State. When this bit gets set to one, HSI transmitter sends "tx_wake" signal to Rx of other device. For a transmitting operation this bit should be one.	0
Rx Receive- Mode	[30]	RW	0 = Stream Receive Mode 1 = Frame Receive Mode	0
RSVD	[29]	R	Reserved for future use	0
Cfg_disable_stream_ break	[28]	RW	0 = Mask the Rx Break interrupt in Stream mode 1 = Unmask the Rx Break interrupt in Stream mode	0
Rx Channel7 Enable	[27]	RW	0 = Rx Ch7 is Disabled. 1 = Rx Ch7 is Enabled.	1
Rx Channel6 Enable	[26]	RW	0 = Rx Ch6 is Disabled. 1 = Rx Ch6 is Enabled.	1
Rx Channel5 Enable	[25]	RW	0 = Rx Ch5 is Disabled. 1 = Rx Ch5 is Enabled.	1
Rx Channel4 Enable	[24]	RW	0 = Rx Ch4 is Disabled. 1 = Rx Ch4 is Enabled.	1
Rx Channel3 Enable	[23]	RW	0 = Rx Ch3 is Disabled. 1 = Rx Ch3 is Enabled.	1
Rx Channel2 Enable	[22]	RW	0 = Rx Ch2 is Disabled. 1 = Rx Ch2 is Enabled.	1
Rx Channel1 Enable	[21]	RW	0 = Rx Ch1 is Disabled. 1 = Rx Ch1 is Enabled.	1
Rx Channel0 Enable	[20]	RW	0 = Rx Ch0 is Disabled. 1 = Rx Ch0 is Enabled.	1
Tx Channel7 Enable	[19]	RW	0 = Tx Ch7 is Disabled. 1 = Tx Ch7 is Enabled.	1
Tx Channel6 Enable	[18]	RW1C	0 = Tx Ch6 is Disabled. 1 = Tx Ch6 is Enabled.	1
Tx Channel5 Enable	[17]	RW	0 = Tx Ch5 is Disabled. 1 = Tx Ch5 is Enabled.	1
Tx Channel4 Enable	[16]	RW	0 = Tx Ch4 is Disabled. 1 = Tx Ch4 is Enabled.	1
Tx Channel3 Enable	[15]	RW	0 = Tx Ch3 is Disabled. 1 = Tx Ch3 is Enabled.	1
Tx Channel2 Enable	[14]	RW	0 = Tx Ch2 is Disabled. 1 = Tx Ch2 is Enabled.	1

Name	Bit	Type	Description	Reset Value
Tx Channel1 Enable	[13]	RW	0 = Tx Ch1 is Disabled. 1 = Tx Ch1 is Enabled.	1
Tx Channel0 Enable	[12]	RW	0 = Tx Ch0 is Disabled. 1 = Tx Ch0 is Enabled.	1
Rx Wake	[11]	RW	0 = Receiver is in Sleep State 1 = Receiver is in Wakeup State	0
Rx Data Flow	[10:9]	RW	b00 = Synchronized Data Flow b01 = Pipelined Data Flow b10 = Receiver Real-time Data Flow b11 = Reserved	00
Tx Transmis- sion Mode	[8]	RW	0 = Stream Transmission Mode 1 = Frame Transmission Mode	0
Receive Time- out Counter	[7:1]	RW	Receive Frame Timeout Counter: 7'h00 – 14400 → 200 MHz AHB CLK 7'h01 – 7200 → 100 MHz AHB CLK 7'h02 – 3600 → 50 MHz AHB CLK 7'h04 – 1800 → 25 MHz AHB CLK 7'h08 – 900 → 12.5 MHz AHB CLK 7'h10 – 450 → 6.25 MHz AHB CLK 7'h20 – 225 → 3.12 MHz AHB CLK 7'h40 – 112 → 1.55 MHz AHB CLK The counter shall be started when the first bit of the Frame has been found. The counter shall be stopped once the receiver has received the correct number of bits for a Frame. If the counter expires before Frame reception is completed, the receiver shall signal to the protocol layer that it has found an incomplete Frame and asserts Rx Error Interrupt.	0x00
Software Reset	[0]	RWAC	Software Reset: This reset affects the entire HSI Controller. Register bits of type ROC, RW, RW1C, RWAC get reset to reset value. The Host Driver shall set this bit to 1 to reset the HSI Controller. This bit is auto clear. The HSI controller should clear this bit, once reset operation is completed.	0

40.4.1.19 ARBITER_PROG

- Base Address: 0x1256_0000
- Address = Base Address + 0x0068, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	RW	Reserved for Future Use	0
Tx Channel7 Priority	[24:22]	RW	This value denotes the priority of Tx channel 7. b000 = 1st priority b001 = 2nd priority b010 = 3rd priority b011 = 4th priority b100 = 5th priority b101 = 6th priority b110 = 7th priority b111 = 8th priority	000
Tx Channel6 Priority	[21:19]	RW	This value denotes the priority of Tx channel 6. b000 = 1st priority b001 = 2nd priority b010 = 3rd priority b011 = 4th priority b100 = 5th priority b101 = 6th priority b110 = 7th priority b111 = 8th priority	000
Tx Channel5 Priority	[18:16]	RW	This value denotes the priority of Tx channel 5. b000 = 1st priority b001 = 2nd priority b010 = 3rd priority b011 = 4th priority b100 = 5th priority b101 = 6th priority b110 = 7th priority b111 = 8th priority	000
Tx Channel4 Priority	[15:13]	RW	This value denotes the priority of Tx channel 4. b000 = 1st priority b001 = 2nd priority b010 = 3rd priority b011 = 4th priority b100 = 5th priority b101 = 6th priority b110 = 7th priority b111 = 8th priority	000
Tx Channel3 Priority	[12:10]	RW	This value denotes the priority of Tx channel 3. b000 = 1st priority b001 = 2nd priority b010 = 3rd priority b011 = 4th priority b100 = 5th priority b101 = 6th priority	000

Name	Bit	Type	Description	Reset Value
			b110 = 7th priority b111 = 8th priority	
Tx Channel2 Priority	[9:7]	RW	This value denotes the priority of Tx channel 2. b000 = 1st priority b001 = 2nd priority b010 = 3rd priority b011 = 4th priority b100 = 5th priority b101 = 6th priority b110 = 7th priority b111 = 8th priority	000
Tx Channel1 Priority	[6:4]	RW	This value denotes the priority of Tx channel 1. b000 = 1st priority b001 = 2nd priority b010 = 3rd priority b011 = 4th priority b100 = 5th priority b101 = 6th priority b110 = 7th priority b111 = 8th priority	000
Tx Channel0 Priority	[3:1]	RW	This value denotes the priority of Tx channel 0. b000 = 1st priority b001 = 2nd priority b010 = 3rd priority b011 = 4th priority b100 = 5th priority b101 = 6th priority b110 = 7th priority b111 = 8th priority	000
Arbitration Priority	[0]	RW	0 = Round Robin Priority 1 = Fixed Priority	0

40.4.1.20 ARBITER_BW1

- Base Address: 0x1256_0000
- Address = Base Address + 0x006C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Tx Channel3 bandwidth	[31:24]	RW	Bandwidth for channel 3. This value denotes the maximum number of 4 Bytes that can be transmitted per transfer. If the Bandwidth count is more than the data availability then the HSI Transmitter should perform only data transfer for the available data. 8'h00 = 1024 Bytes 8'h01 = 4 Bytes 8'h02 = 8 Bytes ... 8'hFF = 1020 Bytes	0x0
Tx Channel2 bandwidth	[23:16]	RW	Bandwidth for channel 2. This value denotes the maximum number of 4 Bytes that can be transmitted per transfer. If the Bandwidth count is more than the data availability then the HSI Transmitter should perform only data transfer for the available data. 8'h00 = 1024 Bytes 8'h01 = 4 Bytes 8'h02 = 8 Bytes ... 8'hFF = 1020 Bytes	0x0
Tx Channel1 bandwidth	[15:8]	RW	Bandwidth for channel 1. This value denotes the maximum number of 4 Bytes that can be transmitted per transfer. If the Bandwidth count is more than the data availability then the HSI Transmitter should perform only data transfer for the available data. 8'h00 = 1024 Bytes 8'h01 = 4 Bytes 8'h02 = 8 Bytes ... 8'hFF = 1020 Bytes	0x0
Tx Channel0 bandwidth	[7:0]	RW	Bandwidth for channel 0. This value denotes the maximum number of 4 Bytes that can be transmitted per transfer. If the Bandwidth count is more than the data availability then the HSI Transmitter should perform only data transfer for the available data. 8'h00 = 1024 Bytes 8'h01 = 4 Bytes 8'h02 = 8 Bytes ... 8'hFF = 1020 Bytes	0x0

40.4.1.21 ARBITER_BW2

- Base Address: 0x1256_0000
- Address = Base Address + 0x0070, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Tx Channel7 bandwidth	[31:24]	RW	Bandwidth for channel 7. This value denotes the maximum number of 4 Bytes that can be trans- mitted per transfer. If the Bandwidth count is more than the data availability then the HSI Transmitter should perform only data transfer for the available data. 8'h00 = 1024 Bytes 8'h01 = 4 Bytes 8'h02 = 8 Bytes ... 8'hFF = 1020 Bytes	0x0
Tx Channel6 bandwidth	[23:16]	RW	Bandwidth for channel 6. This value denotes the maximum number of 4 Bytes that can be trans- mitted per transfer. If the Bandwidth count is more than the data availability then the HSI Transmitter should perform only data transfer for the available data. 8'h00 = 1024 Bytes 8'h01 = 4 Bytes 8'h02 = 8 Bytes ... 8'hFF = 1020 Bytes	0x0
Tx Channel5 bandwidth	[15:8]	RW	Bandwidth for channel 5. This value denotes the maximum number of 4 Bytes that can be trans- mitted per transfer. If the Bandwidth count is more than the data availability then the HSI Transmitter should perform only data transfer for the available data. 8'h00 = 1024 Bytes 8'h01 = 4 Bytes 8'h02 = 8 Bytes ... 8'hFF = 1020 Bytes	0x0
Tx Channel4 bandwidth	[7:0]	RW	Bandwidth for channel 4. This value denotes the maximum number of 4 Bytes that can be trans- mitted per transfer. If the Bandwidth count is more than the data availability then the HSI Transmitter should perform only data transfer for the available data. 8'h00 = 1024 Bytes 8'h01 = 4 Bytes 8'h02 = 8 Bytes ... 8'hFF = 1020 Bytes	0x0

40.4.1.22 CAPA

- Base Address: 0x1256_0000
- Address = Base Address + 0x0074, Reset Value = 0x0000_07FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:21]	R	Reserved for future use.	0x0
Wakeup Event	[20]	R	0 = Wakeup Event is supported 1 = Wakeup Event is not supported	0
Tx Base clock Frequency	[19:11]	R	d0 = Reserved d1 = 1 MHz d2 = 2 MHz .... d200 = 200 MHz	0
Number of Slave DMA's supported	[10:8]	R	b000 = 1 Slave DMA supported b001 = 2 Slave DMA supported b010 = 3 Slave DMA supported b011 = 4 Slave DMA supported b100 = 5 Slave DMA supported b101 = 6 Slave DMA supported b110 = 7 Slave DMA supported b111 = 8 Slave DMA supported	b111
Rx DMA	[7]	R	0 = Not supported 1 = DMA is supported.	1
Tx DMA	[6]	R	0 = Not supported 1 = DMA is supported.	1
Number of Rx Channels Sup-ported	[5:3]	R	b000 = Reserved b001 = 1 Rx channel supported b010 = 2 Rx channels supported ... b111 = 7 Rx channels supported	b111
Number of Tx Channels Sup-ported	[2:0]	R	b000 = Reserved b001 = 1 Tx channel supported b010 = 2 Tx channels supported ... b111 = 7 Tx channels supported	b111

40.4.1.23 Tx_CH0_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x0078, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Tx Ch0 Data Port register	[31:0]	RW	Write the data to the Tx channel 0 FIFO	0x0

40.4.1.24 Tx_CH1_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x007C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Tx Ch1 Data Port register	[31:0]	RW	Write the data to the Tx channel 1 FIFO	0x0

40.4.1.25 Tx_CH2_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Tx Ch2 Data Port register	[31:0]	RW	Write the data to the Tx channel 2 FIFO	0x0

40.4.1.26 Tx_CH3_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x0084, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Tx Ch3 Data Port register	[31:0]	RW	Write the data to the Tx channel 3 FIFO	0x0

40.4.1.27 Tx_CH4_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x0088, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Tx Ch4 Data Port register	[31:0]	RW	Write the data to the Tx channel 4 FIFO	0x0

40.4.1.28 Tx_CH5_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x008C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Tx Ch5 Data Port register	[31:0]	RW	Write the data to the Tx channel 5 FIFO	0x0

40.4.1.29 Tx_CH6_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Tx Ch6 Data Port register	[31:0]	RW	Write the data to the Tx channel 6 FIFO	0x0

40.4.1.30 Tx_CH7_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x0094, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Tx Ch7 Data Port register	[31:0]	RW	Write the data to the Tx channel 7 FIFO	0x0

40.4.1.31 Rx_CH0_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x0098, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Rx Ch0 Data Port register	[31:0]	R	Read the data from the Rx channel 0 FIFO	0x0

40.4.1.32 Rx_CH1_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x009C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Rx Ch1 Data Port register	[31:0]	R	Read the data from the Rx channel 1 FIFO	0x0

40.4.1.33 Rx_CH2_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Rx Ch2 Data Port register	[31:0]	R	Read the data from the Rx channel 2 FIFO	0x0

40.4.1.34 Rx_CH3_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x00A4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Rx Ch3 Data Port register	[31:0]	R	Read the data from the Rx channel 3 FIFO	0x0

40.4.1.35 Rx_CH4_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x00A8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Rx Ch4 Data Port register	[31:0]	R	Read the data from the Rx channel 4 FIFO	0x0

40.4.1.36 Rx_CH5_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x00AC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Rx Ch5 Data Port register	[31:0]	R	Read the data from the Rx channel 5 FIFO	0x0

40.4.1.37 Rx_CH6_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x00B0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Rx Ch6 Data Port register	[31:0]	R	Read the data from the Rx channel 6 FIFO	0x0

40.4.1.38 Rx_CH7_DATA

- Base Address: 0x1256_0000
- Address = Base Address + 0x00B4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Rx Ch7 Data Port register	[31:0]	R	Read the data from the Rx channel 7 FIFO	0x0

40.4.1.39 ERR_INT_STAT

- Base Address: 0x1256_0000
- Address = Base Address + 0x00B8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	R	Reserved for future use	0x0
Data timeout interrupt for channel 7	[9]	RW1C	This status bit is set when data timeout counter for channel 7 reaches the data timeout counter value. On receiving the interrupt the host driver should read 4 Bytes of data from Rx channel 7 buffer and then read HSI Status register to find the further status of the Rx channel 7 Buffer. The host driver has to read the Rx channel 7 FIFO on 4 Bytes basis, till the FIFO is completely empty.	0
Data timeout interrupt for channel 6	[8]	RW1C	This status bit is set when data timeout counter for channel 6 reaches the data timeout counter value. On receiving the interrupt the host driver should read 4 Bytes of data from Rx channel 6 buffer and then read HSI Status register to find the further status of the Rx channel 6 Buffer. The host driver has to read the Rx channel 6 FIFO on 4 Bytes basis, till the FIFO is completely empty.	0
Data timeout interrupt for channel 5	[7]	RW1C	This status bit is set when data timeout counter for channel 5 reaches the data timeout counter value. On receiving the interrupt the host driver should read 4 Bytes of data from Rx channel 5 buffer and then read HSI Status register to find the further status of the Rx channel 5 Buffer. The host driver has to read the Rx channel 5 FIFO on 4 Bytes basis, till the FIFO is completely empty.	0
Data timeout interrupt for channel 4	[6]	RW1C	This status bit is set when data timeout counter for channel 4 reaches the data timeout counter value. On receiving the interrupt the host driver should read 4 Bytes of data from Rx ch4 buffer and then read HSI Status register to find the further status of the Rx channel 4 Buffer. The host driver has to read the Rx channel 4 FIFO on 4 Bytes basis, till the FIFO is completely empty.	0
Data timeout interrupt for channel 3	[5]	RW1C	This status bit is set when data timeout counter for channel 3 reaches the data timeout counter value. On receiving the interrupt the host driver should read 4 Bytes of data from Rx channel 3 buffer and then read HSI Status register to find the further status of the Rx channel 3 Buffer. The host driver has to read the Rx channel 3 FIFO on 4 Bytes basis, till the FIFO is completely empty.	0
Data timeout interrupt for channel 2	[4]	RW1C	This status bit is set when data timeout counter for channel 2 reaches the data timeout counter value. On receiving the interrupt the host driver should read	0

Name	Bit	Type	Description	Reset Value
			4 Bytes of data from Rx ch2 buffer and then read HSI Status register to find the further status of the Rx channel 2 Buffer. The host driver has to read the Rx channel 2 FIFO on 4 Bytes basis, till the FIFO is completely empty.	
Data timeout interrupt for channel 1	[3]	RW1C	This status bit is set when data timeout counter for channel 1 reaches the data timeout counter value. On receiving the interrupt the host driver should read 4 Bytes of data from Rx ch1 buffer and then read HSI Status register to find the further status of the Rx channel 1 Buffer. The host driver has to read the Rx channel 1 FIFO on 4 Bytes basis, till the FIFO is completely empty.	0
Data timeout interrupt for channel 0	[2]	RW1C	This status bit is set when data timeout counter for channel 0 reaches the data timeout counter value. On receiving the interrupt the host driver should read 4 Bytes of data from Rx channel 0 buffer and then read HSI Status register to find the further status of the Rx channel 0 Buffer. The host driver has to read the Rx channel 0 FIFO on 4 Bytes basis, till the FIFO is completely empty.	0
Rx Error Interrupt	[1]	RW1C	0 = No Error 1 = Error Occurred during the Receive Operation, say for e.g. Break Trans- mission, bus contention etc.	0
RX Break interrupt	[0]	RW1C	This status bit is set "1" when HSI Rx senses break command (atleast 36 zeros) during bus struck.	0

40.4.1.40 ERR_INT_STAT_EN

- Base Address: 0x1256_0000
- Address = Base Address + 0x00BC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	R	Reserved for future use.	0x0
Data timeout interrupt for channel 7	[9]	RW	0 = Interrupt status masked for "data timeout for channel 7" interrupt. 1 = Interrupt status enabled for "data timeout for channel 7" interrupt.	0
Data timeout interrupt for channel 6	[8]	RW	0 = Interrupt status masked for "data timeout for channel 6" interrupt. 1 = Interrupt status enabled for "data timeout for channel 6" interrupt.	0
Data timeout interrupt for channel 5	[7]	RW	0 = Interrupt status masked for "data timeout for channel 5" interrupt. 1 = Interrupt status enabled for "data timeout for channel 5" interrupt.	0
Data timeout interrupt for channel 4	[6]	RW	0 = Interrupt status masked for "data timeout for channel 4" interrupt. 1 = Interrupt status enabled for "data timeout for channel 4" interrupt.	0
Data timeout interrupt for channel 3	[5]	RW	0 = Interrupt status masked for "data timeout for channel 3" interrupt. 1 = Interrupt status enabled for "data timeout for v3" interrupt.	0
Data timeout interrupt for channel 2	[4]	RW	0 = Interrupt status masked for "data timeout for v2" interrupt. 1 = Interrupt status enabled for "data timeout for channel 2" interrupt.	0
Data timeout interrupt for channel 1	[3]	RW	0 = Interrupt status masked for "data timeout for channel 1" interrupt. 1 = Interrupt status enabled for "data timeout for channel 1" interrupt.	0
Data timeout interrupt for channel 0	[2]	RW	0 = Interrupt status masked for "data timeout for channel 0" interrupt. 1 = Interrupt status enabled for "data timeout for channel 0" interrupt.	0
Rx Error Interrupt	[1]	RW	0 = Interrupt status masked for "Rx Error" interrupt. 1 = Interrupt status enabled for "Rx Error" interrupt.	0
Rx Break Interrupt	[0]	RW	0 = Interrupt status masked for "Rx Break" interrupt. 1 = Interrupt status enabled for "Rx Break" interrupt.	0

40.4.1.41 ERR_INT_SIG_EN

- Base Address: 0x1256_0000
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	R	Reserved for future use	0x0
Data timeout interrupt for channel 7	[9]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "data timeout for channel 7" interrupt. 1 = Interrupt signal enabled for "data timeout for channel 7" interrupt.	0
Data timeout interrupt for channel 6	[8]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "data timeout for channel 6" interrupt. 1 = Interrupt signal enabled for "data timeout for channel 6" interrupt.	0
Data timeout interrupt for channel 5	[7]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "data timeout for channel 5" interrupt. 1 = Interrupt signal enabled for "data timeout for channel 5" interrupt.	0
Data timeout interrupt for channel 4	[6]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "data timeout for channel 4" interrupt. 1 = Interrupt signal enabled for "data timeout for channel 4" interrupt.	0
Data timeout interrupt for channel 3	[5]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "data timeout for channel 3" interrupt. 1 = Interrupt signal enabled for "data timeout for channel 3" interrupt.	0
Data timeout interrupt for channel 2	[4]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "data timeout for channel 2" interrupt. 1 = Interrupt signal enabled for "data timeout for channel 2" interrupt.	0
Data timeout interrupt for channel 1	[3]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "data timeout for channel 1" interrupt.	0

Name	Bit	Type	Description	Reset Value
			1 = Interrupt signal enabled for "data timeout for channel 1" interrupt.	
Data timeout interrupt for channel 0	[2]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "data timeout for channel 0" interrupt. 1 = Interrupt signal enabled for "data timeout for channel 0" interrupt.	0
Rx Error Interrupt	[1]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Rx Error" interrupt. 1 = Interrupt signal enabled for "Rx Error" interrupt.	0
Rx Break Interrupt	[0]	RW	Setting this bit will enable interrupt generation on interrupt line. 0 = Interrupt signal masked for "Rx Break" interrupt. 1 = Interrupt signal enabled for "Rx Break" interrupt.	0

40.4.1.42 STATUS1

- Base Address: 0x1256_0000
- Address = Base Address + 0x00C4, Reset Value = 0x0000_01FF

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	R	Reserved for future use	0x0
Rx_idle	[8]	R	0 = Rx is not idle 1 = Rx is idle	1
Tx_ch7_idle	[7]	R	0 = Tx channel 7 is not idle 1 = Tx channel 7 is idle	1
Tx_ch6_idle	[6]	R	0 = Tx channel 6 is not idle 1 = Tx channel 6 is idle	1
Tx_ch5_idle	[5]	R	0 = Tx channel 5 is not idle 1 = Tx channel 5 is idle	1
Tx_ch4_idle	[4]	R	0 = Tx channel 4 is not idle 1 = Tx channel 4 is idle	1
Tx_ch3_idle	[3]	R	0 = Tx channel 3 is not idle 1 = Tx channel 3 is idle	1
Tx_ch2_idle	[2]	R	0 = Tx channel 2 is not idle 1 = Tx channel 2 is idle	1
Tx_ch1_idle	[1]	R	0 = Tx channel 1 is not idle 1 = Tx channel 1 is idle	1
Tx_ch0_idle	[0]	R	0 = Tx channel 0 is not idle 1 = Tx channel 0 is idle	1

40.4.1.43 PROGRAM1

- Base Address: 0x1256_0000
- Address = Base Address + 0x00C8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	R	Reserved for future use	0x0
Number of Rx channel ID bits	[3:2]	RW	This bit sets the number of channel ID bits per frame or stream for a receiving operation. 0 = 0 bit 1 = 1 bit 2 = 2 bit 3 = 3 bit	0
Number of Tx channel ID bits	[1:0]	RW	This bit sets the number of channel ID bits per frame or stream for a transmitting operation. 0 = 0 bit 1 = 1 bit 2 = 2 bit 3 = 3 bit	0

40.4.1.44 VER_INFO

- Base Address: 0x1256_0000
- Address = Base Address + 0x00FC, Reset Value = 0x0000_1014

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	R	Reserved for future use	0x0
HSI_spec_versi on	[15:8]	R	HSI Specification Version 00 = Version number is 0.0 10 = Version number is 1.0 …. 12 = Version number is 1.2	0x10
RTL_version	[7:0]	R	RTL Version e.g. 00 = Version number is 0.0 23 = Version number is 2.3	0x14

41 Display Controller

41.1 Overview

Display controller consists of logic for transferring image data from a local bus of the camera interface controller or a video buffer located in system memory to an external LCD driver interface. The LCD driver interface supports three kinds of interfaces. They are RGB-interface, indirect-i80 interface, and YUV interface for write-back. The display controller uses up to five overlay image windows that support various color formats, 256 level alpha blending, color key, x-y position control, soft scrolling, and variable window size, among others.

Display controller supports various color formats such as RGB (1 to 24 BPP) and YCbCr 4:4:4 (only local bus). You can program the display controller to support the different requirements on screen that associates with the number of horizontal and vertical pixels, data line width for the data interface, interface timing, and refresh rate.

Display controller transfers the video data and generates the necessary control signals, such as, RGB_VSYNC, RGB_HSYNC, RGB_VCLK, RGB_VDEN, SYS_CS0, SYS_CS1, and SYS_WE. Additionally generating control signals, display controller contains data ports for video data (RGB_VD[23:0], and SYS_VD)

Figure 41-1 illustrates the block diagram of display controller.

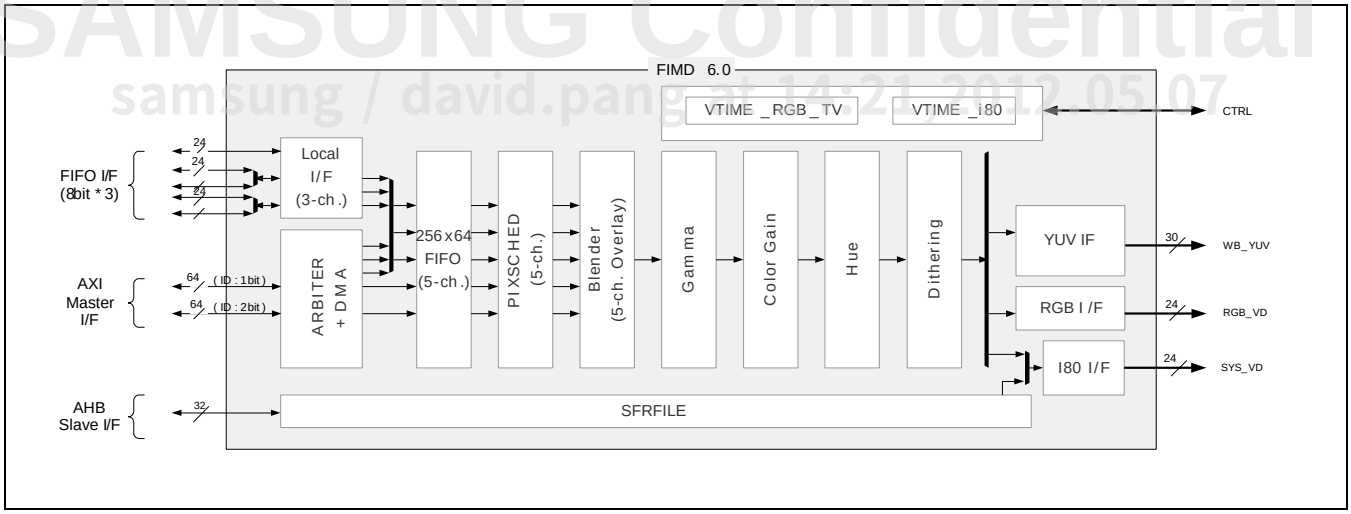


Figure 41-1 Block Diagram of Display Controller

41.2 Features of Display Controller

The features of the display controller include:

Bus Interface	AMBA AXI 64-bit Master/AHB 32-bit Slave Local Video Bus (YCbCr/RGB)
Video Output Interface	RGB Interface (24-bit Parallel/8-bit Serial) • Dummy insertion, color sub-sampling (RGB skip) mode • General, delta structure Indirect i80 interface Write-Back interface (YUV444 30-bit)
Dual Output Mode	Supports i80 and Write-Back Supports RGB and Write-Back
PIP (OSD) function	Supports 8-BPP (bit per pixel) palletized color Supports 16-BPP non-palletized color Supports unpacked 18 BPP non-palletized color Supports unpacked 24 BPP non-palletized color
	Supports X,Y indexed position
	Supports 8-bit Alpha blending (Plane/Pixel)
CSC (Internal)	RGB to YCbCr (4:2:2)
Source format	Window 0 Supports 1, 2, 4, or 8 BPP palletized color Supports 16, 18, or 24 BPP non-palletized color Supports RGB (8:8:8) local input from Local Bus (FIMC0) Window 1 Supports 1, 2, 4, or 8 BPP palletized color Supports 16, 18, or 24 BPP non-palletized color Supports RGB (8:8:8) local input from Local Bus (FIMC1) Window 2 Supports 1, 2, 4, or 8 BPP palletized color Supports 16, 18, or 24 BPP non-palletized color Supports RGB (8:8:8) local input from Local Bus (FIMC2) Window 3/4 Supports 1, 2, 4, or 8 BPP palletized color Supports 16, 18, or 24 BPP non-palletized color
Configurable Burst Length	Programmable 4/8/16 Burst DMA
Palette	Window 0/1/2/3/4 Supports 256 × 32 bits palette memory (5EA: One palette memory for each window)
Soft Scrolling	Horizontal = 1 Byte resolution Vertical = 1 pixel resolution
Virtual Screen	Virtual image can have up to 16 MB image size. Each window can have its own virtual area.

Transparent Overlay	Supports transparent overlay
Color Key (Chroma Key)	Supports color key function Supports simultaneously color key and blending function
Partial Display	Supports LCD partial display function through i80 interface
Image Enhancement	Supports gamma control
	Supports hue control
	Supports color gain control
	Supports pixel compensation (only for delta structure)
Video Clock Source	SCLK_FIMD0 for display controller (from CMU module)
Maximum VCLK in RGB Interface	Display Controller = 80 MHz

41.3 Functional Description

The functional description section describes the functionality of display controller.

41.3.1 Brief Description of the Sub-Block

The display controller consists of a VSFR, VDMA, VPRCS, VTIME, and video clock generator.

To configure the display controller, the VSFR has

- 121 programmable register sets
- one gamma LUT register set (64 registers)
- one i80 command register set (12 registers)
- five 256 × 32 palette memories

VDMA is a dedicated display DMA that transfers video data in frame memory to VPRCS. By using this special DMA, you can display video data on screen without CPU intervention.

VPRCS receives video data from VDMA and sends it to display device (LCD) through data ports (RGB_VD, or SYS_VD), after changing the video data into a suitable data format, for example, 8-bit per pixel mode (8 BPP mode) or 16-bit per pixel mode (16 BPP mode).

VTIME consists of programmable logic to support the variable requirement of interface timing and rates commonly found in different LCD drivers. The VTIME block generates RGB_VSYNC, RGB_HSYNC, RGB_VCLK, RGB_VDEN, VEN_VSYNC, VEN_HSYNC, VEN_FIELD, VEN_HREF, SYS_CS0, SYS_CS1, SYS_WE, and so on.

Using the display controller data, you can select one of the above data paths by setting LCDBLK_CFG Register (0x1001_0210). For more information, refer to the "System Others" manual.

41.3.2 Data Flow

FIFO is in the VDMA. If FIFO is empty or partially empty, the VDMA requests data fetching from frame memory based on burst memory transfer mode. The data transfer rate determines the size of FIFO.

The display controller contains five FIFOs (Three local FIFOs and two DMA FIFOs), since it needs to support the overlay window display mode. Use one FIFO for one screen display mode.

VPRCS fetches data from FIFO. It contains the following functions for final image data: blending, image enhancing, and scheduling. It also supports the overlay function. This can overlay any image up to five window images, whose smaller or same size can be blended with the main window image having programmable alpha blending or color (chroma) key function.

Figure 41-2 shows the data flow from system bus to output buffer.

VDMA has five DMA channels (Ch0-Ch4) and three local input interfaces (CAMIF0, CAMIF1, and (CAMIF2 or CAMIF3)). The Color Space Conversion (CSC) block changes Hue (YCbCr, local input only) data to RGB data for blending operation. Also, the alpha values written in SFR determine the level of blending. Data from output buffer appears in the Video Data Port.

NOTE: The performance of the all these local input interfaces is limited by the scale ratio of the input and output image resolution (TBD).

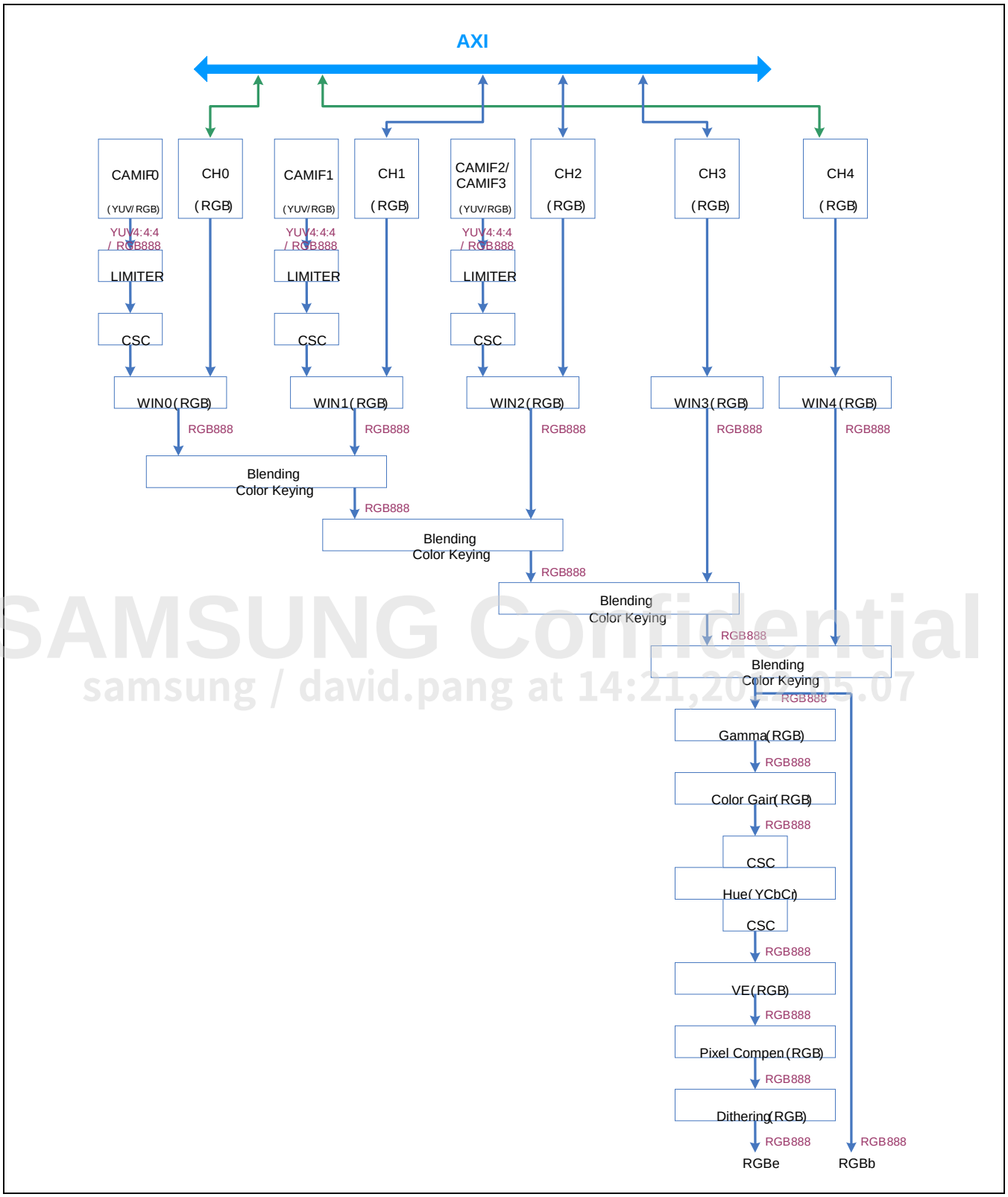


Figure 41-2 Block Diagram of the Data Flow

41.3.2.1 Interface

The display controller supports three types of interfaces:

- The first type is the conventional RGB interface, which uses RGB data, vertical/ horizontal sync, data valid signal, and data sync clock.
- The second type is the indirect i80 Interface, which uses address, data, chip select, read/ write control, and register/ status indicating signal. The LCD driver using i80 Interface contains a frame buffer and can self-refresh, so the display controller updates one still image by writing only one time to the LCD.
- The third type is FIFO interface with CAMIFx selected FIMDxWB_DEST Bit Field on CAMERA_CONTROL Register in System Register for writeback.

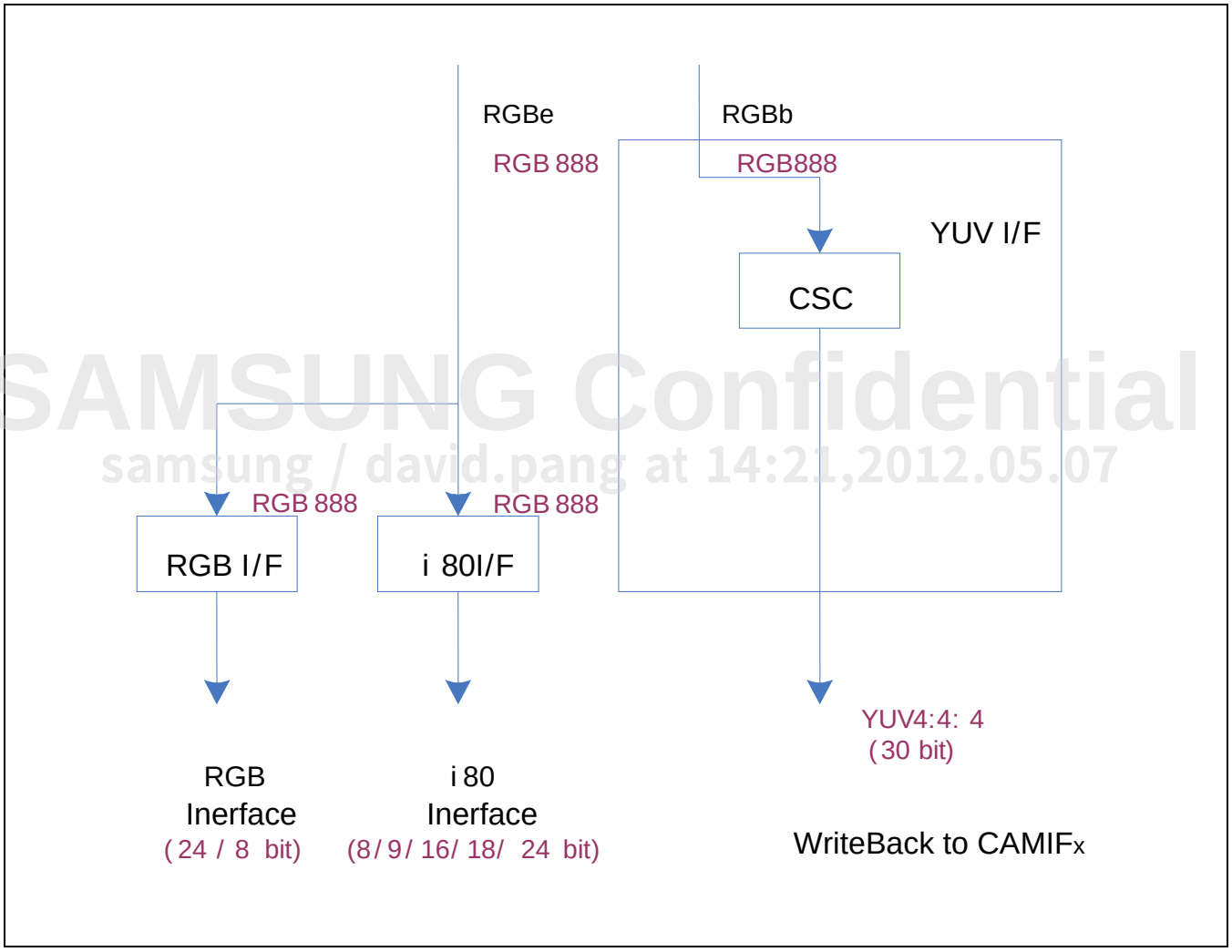


Figure 41-3 Block Diagram of the Interface

41.3.3 Overview of the Color Data

The overview of the color data section describes the RGB data format and 25 BPP display of display controller.

41.3.3.1 RGB Data Format

The display controller requests the specified memory format of frame buffer. *Figure 41-4* illustrates some examples of each display mode.

41.3.3.2 25 BPP Display (A888)

Figure 41-4 illustrates the examples of each display mode.

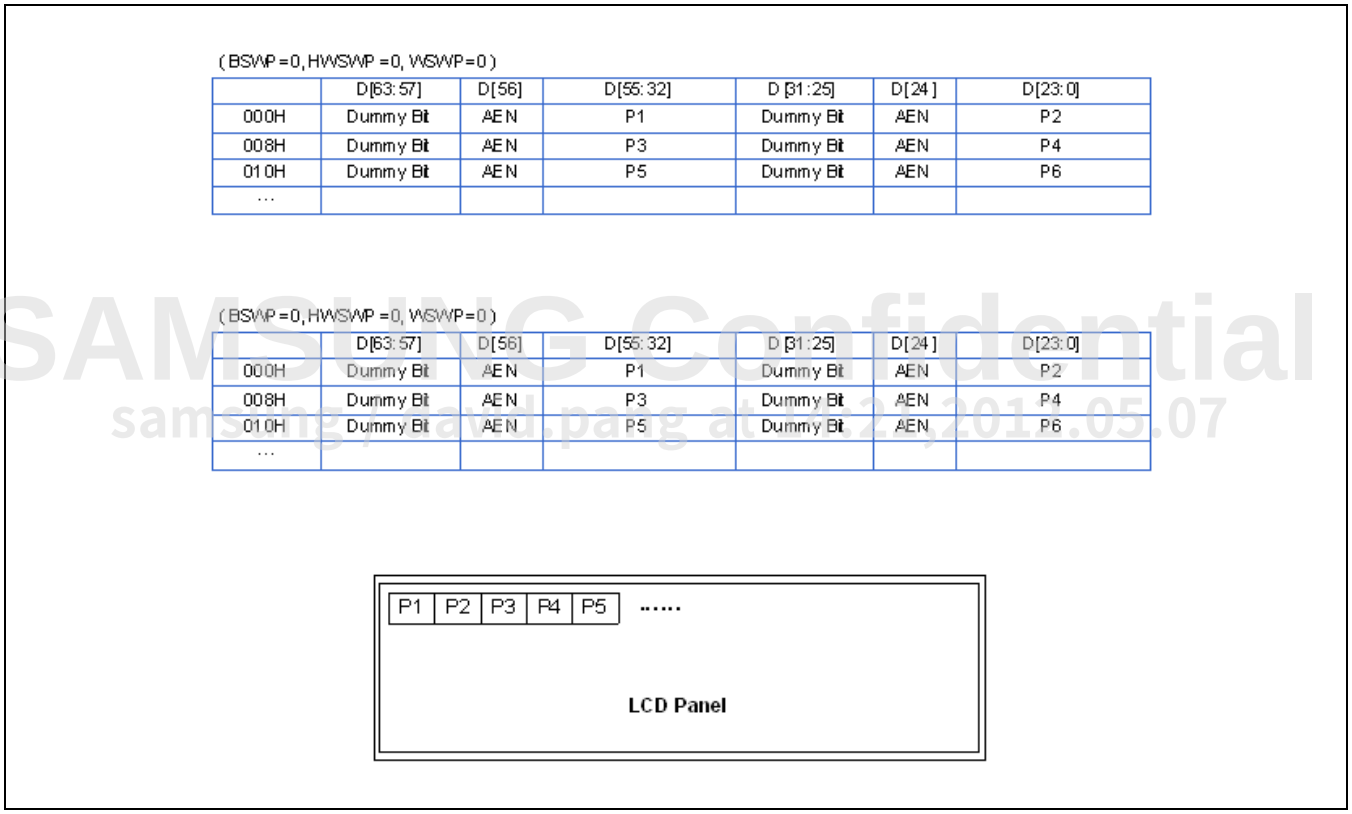


Figure 41-4 Memory Format of 25 BPP(A888) Display

NOTE:

1. AEN: Specifies the transparency value selection bit.
   AEN = 0: Selects ALPHA0.
   AEN = 1: Selects ALPHA1.
   When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects..
   SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
   For more information, refer to the section on "SFR".
2. D[23:16] = Red data, D[15:8] = Green data, and D[7:0] = Blue data.

41.3.3.2.1 32BPP (8888) Mode

Figure 41-5 illustrates the pixel data that contains alpha value.

(BYSWP=0 , HWSWP=0, WSWP= 0)

	D[63:56]	D[55:32]	D[31:24]	D[23:0]
000H	ALPHA value	P1	ALPHA value	P2
008H	ALPHA value	P3	ALPHA value	P4
010H	ALPHA value	P5	ALPHA value	P6
…				

(BYSWP=0 , HWSWP=0, WSWP= 1)

	D[63:56]	D[55:32]	D[31:24]	D[23:0]
000H	ALPHA value	P2	ALPHA value	P1
008H	ALPHA value	P4	ALPHA value	P3
010H	ALPHA value	P6	ALPHA value	P5
…				

Figure 41-5 Memory Format of 32 BPP (8888) Display

41.3.3.2.2 24 BPP Display (A887)

Figure 41-6 illustrates the 24 BPP display.

(BSWP=0, HWSWP=0, WSWP=0)

	D[63:56]	D[55]	D[54:32]	D[31:24]	D[23]	D[22:0]
000H	Dummy Bit	AEN	P1	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3	Dummy Bit	AEN	P4
010H	Dummy Bit	AEN	P5	Dummy Bit	AEN	P6
…						

(BSWP=0, HWSWP=0, WSWP=1)

	D[63:56]	D[55]	D[54:32]	D[31:24]	D[23]	D[22:0]
000H	Dummy Bit	AEN	P2	Dummy Bit	AEN	P1
008H	Dummy Bit	AEN	P4	Dummy Bit	AEN	P3
010H	Dummy Bit	AEN	P6	Dummy Bit	AEN	P5
…						

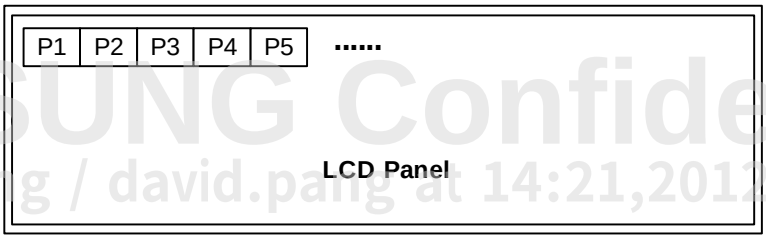


Figure 41-6 Memory Format of 24 BPP(A887) Display

NOTE:

1. AEN: Specifies the transparency value selection bit.
   AEN = 0: Selects ALPHA0.
   AEN = 1: Selects ALPHA1.
   When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects.
   SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
   For more information, refer to the section on "SFR".
2. D[22:15] = Red data, D[14:7] = Green data, and D[6:0] = Blue data.

41.3.3.2.3 24 BPP Display (888)

Figure 41-7 illustrates the 24 BPP display.

(BSWP=0, HWSWP=0, WSWP=0)

	D[63:56]	D[55:32]	D[31:24]	D[23:0]
000H	Dummy Bit	P1	Dummy Bit	P2
008H	Dummy Bit	P3	Dummy Bit	P4
010H	Dummy Bit	P5	Dummy Bit	P6
…				

(BSWP=0, HWSWP=0, WSWP=1)

	D[63:56]	D[55:32]	D[31:24]	D[23:0]
000H	Dummy Bit	P2	Dummy Bit	P1
008H	Dummy Bit	P4	Dummy Bit	P3
010H	Dummy Bit	P6	Dummy Bit	P5
…				

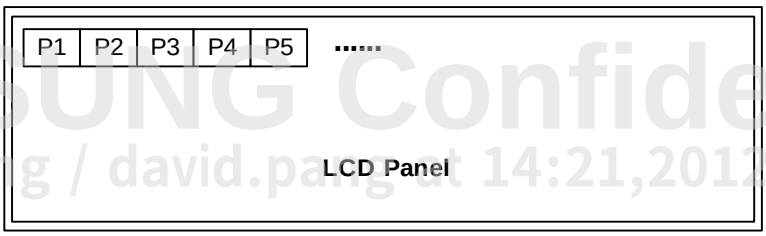


Figure 41-7 Memory Format of 24 BPP (888) Display

NOTE: D[23:16] = Red data, D[15:8] = Green data, and D[7:0] = Blue data.

41.3.3.2.4 19 BPP Display (A666)

Figure 41-8 illustrates the 19 BPP display.

(BSWP=0, HWSWP=0, WSWP=0)

	D[63:51]	D[50]	D[49:32]	D[31:19]	D[18]	D[17:0]
000H	Dummy Bit	AEN	P1	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3	Dummy Bit	AEN	P4
010H	Dummy Bit	AEN	P5	Dummy Bit	AEN	P6
…						

(BSWP=0, HWSWP=0, WSWP=1)

	D[63:51]	D[50]	D[49:32]	D[31:19]	D[18]	D[17:0]
000H	Dummy Bit	AEN	P2	Dummy Bit	AEN	P1
008H	Dummy Bit	AEN	P4	Dummy Bit	AEN	P3
010H	Dummy Bit	AEN	P6	Dummy Bit	AEN	P5
…						

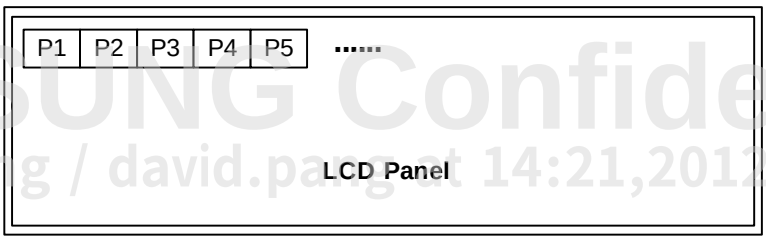


Figure 41-8 Memory Format of 19 BPP (A666) Display

NOTE:

1. AEN: Specifies the transparency value selection bit.
   AEN = 0: Selects ALPHA0.
   AEN = 1: Selects ALPHA1.
   When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects..
   SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
   For more information, refer to the section on "SFR".
2. D[17:12] = Red data, D[11:6] = Green data, and D[5:0] = Blue data.

41.3.3.2.5 18 BPP Display (666)

Figure 41-9 illustrates the 18 BPP display.

(BSWP=0, HWSWP=0, WSWP=0)

	D[63:50]	D[49:32]	D[31:18]	D[17:0]
000H	Dummy Bit	P1	Dummy Bit	P2
008H	Dummy Bit	P3	Dummy Bit	P4
010H	Dummy Bit	P5	Dummy Bit	P6
…				

(BSWP=0, HWSWP=0, WSWP=1)

	D[63:50]	D[49:32]	D[31:18]	D[17:0]
000H	Dummy Bit	P2	Dummy Bit	P1
008H	Dummy Bit	P4	Dummy Bit	P3
010H	Dummy Bit	P6	Dummy Bit	P5
…				

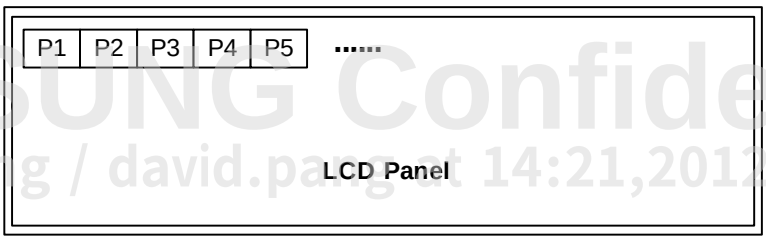


Figure 41-9 Memory Format of 18 BPP (666) Display

NOTE: D[17:12] = Red data, D[11:6] = Green data, and D[5:0] = Blue data.

41.3.3.2.6 16 BPP Display (A555)

Figure 41-10 illustrates the 16 BPP display.

(BSWP=0, HWSWP=0, WSWP=0)

	D[63]	D[62:48]	D[47]	D[46:32]	D[31]	D[30:16]	D[15]	D[14:0]
000H	AEN1	P1	AEN2	P2	AEN3	P3	AEN4	P4
004H	AEN5	P5	AEN6	P6	AEN7	P7	AEN8	P8
008H	AEN9	P9	AEN10	P10	AEN11	P11	AEN12	P12
…								

(BSWP=0, HWSWP=0, WSWP=1)

	D[63]	D[62:48]	D[47]	D[46:32]	D[31]	D[30:16]	D[15]	D[14:0]
000H	AEN3	P3	AEN4	P4	AEN1	P1	AEN2	P2
004H	AEN7	P7	AEN8	P8	AEN5	P5	AEN6	P6
008H	AEN11	P11	AEN12	P12	AEN9	P9	AEN10	P10
…								

(BSWP=0, HWSWP=1, WSWP=0)

	D[63]	D[62:48]	D[47]	D[46:32]	D[31]	D[30:16]	D[15]	D[14:0]
000H	AEN4	P4	AEN3	P3	AEN2	P2	AEN1	P1
004H	AEN8	P8	AEN7	P7	AEN6	P6	AEN5	P5
008H	AEN12	P12	AEN11	P11	AEN10	P10	AEN9	P9
…								

Figure 41-10 Memory Format of 16 BPP (A555) Display

NOTE:

1. AEN: Specifies the transparency value selection bit.
   AEN = 0: Selects ALPHA0.
   AEN = 1: Selects ALPHA1.
   When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects.
   SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
   For more information, refer to the section on "SFR".
2. D[14:10] = Red data, D[9:5] = Green data, and D[4:0] = Blue data.

41.3.3.2.7 16 BPP Display (1555)

Figure 41-11 illustrates the 16 BPP display.

(BSWP=0 , HWSWP=0 , WSWP=0)

	D[63:48]	D[47:32]	D[31:16]	D[15:0]
000H	P1	P2	P3	P4
008H	P5	P6	P7	P8
010H	P9	P10	P11	P12
…				

(BSWP=0 , HWSWP=0 , WSWP=1)

	D[63:48]	D[47:32]	D[31:16]	D[15:0]
000H	P3	P4	P1	P2
008H	P7	P8	P5	P6
010H	P11	P12	P9	P10
…				

(BSWP=0 , HWSWP=1 , WSWP=0)

	D[63:48]	D[47:32]	D[31:16]	D[15:0]
000H	P4	P3	P2	P1
008H	P8	P7	P6	P5
010H	P12	P11	P10	P9
…				

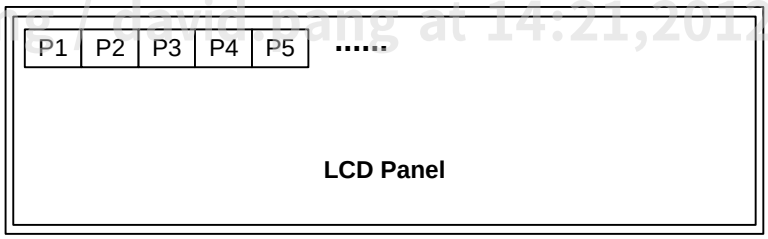


Figure 41-11 Memory Format of 16 BPP (1555) Display

NOTE: {D[14:10], D[15]} = Red data, {D[9:5], D15]} = Green data, and {D[4:0], D[15]} = Blue data.

41.3.3.2.8 16 BPP Display (565)

Figure 41-12 illustrates the 16 BPP display.

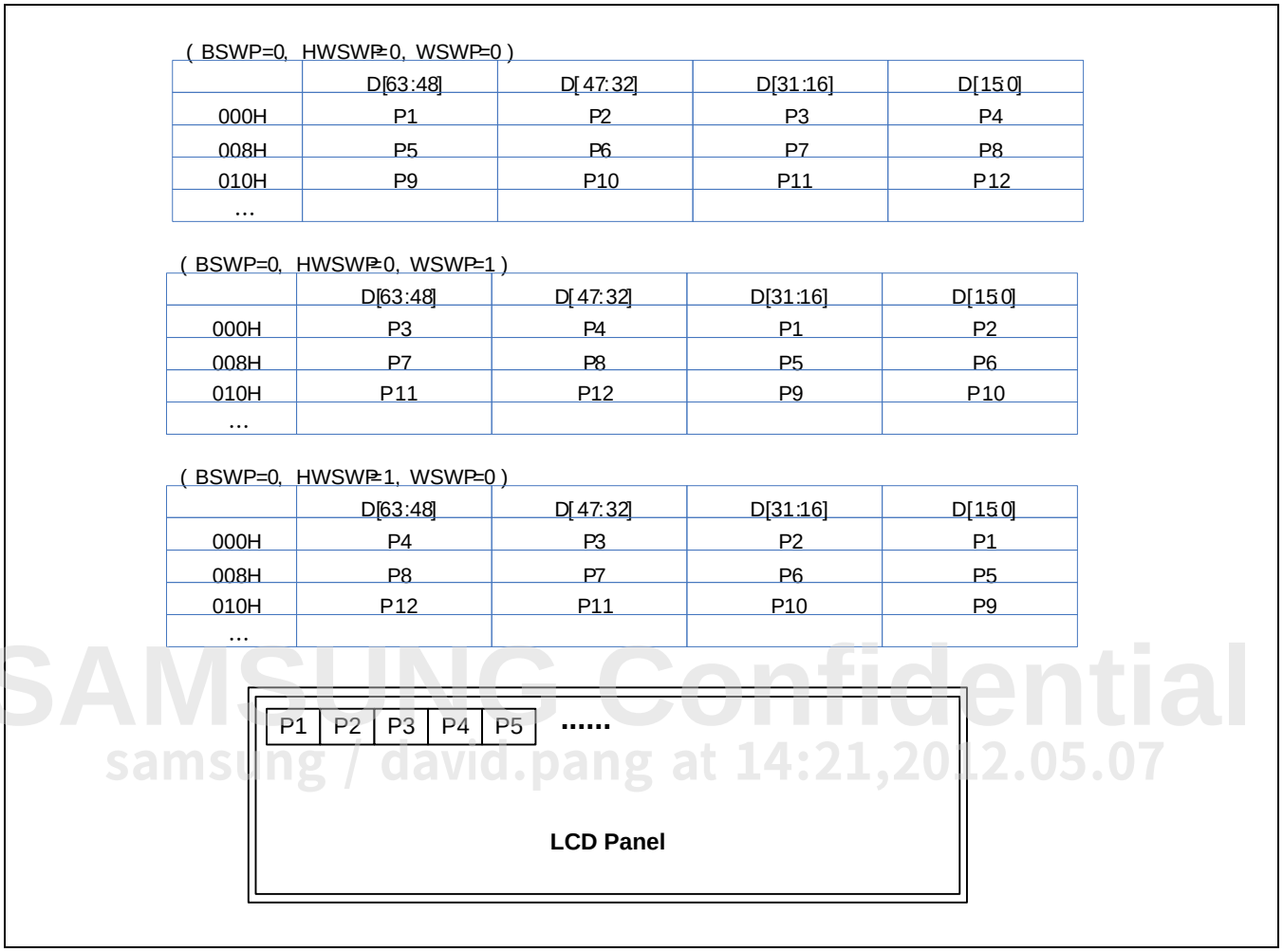


Figure 41-12 Memory Format of 16 BPP (565) Display

NOTE: D[15:10] = Red data, D[10:5] = Green data, and D[4:0] = Blue data.

Figure 41-13 illustrates the 16 BPP (5:6:5) display types.

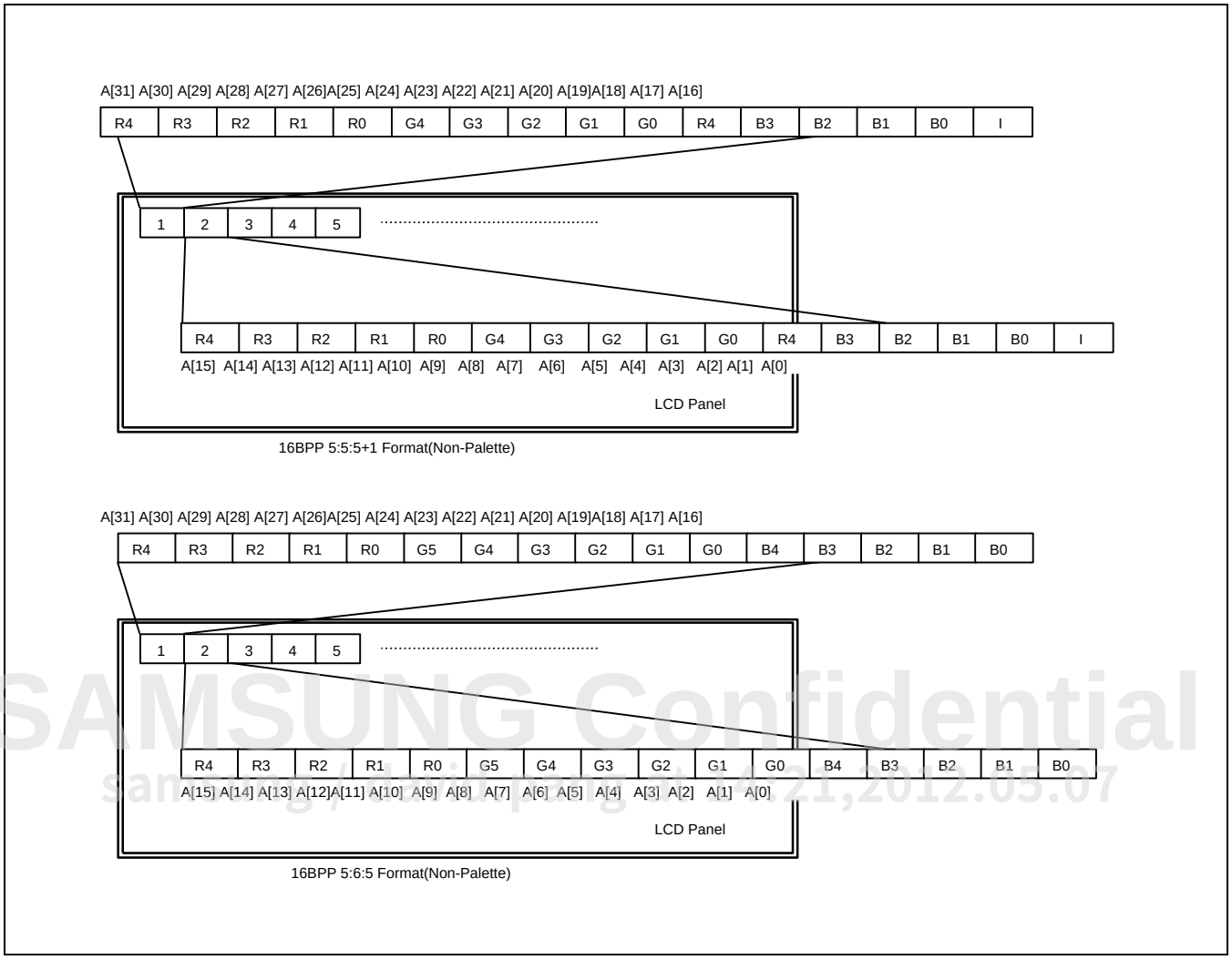
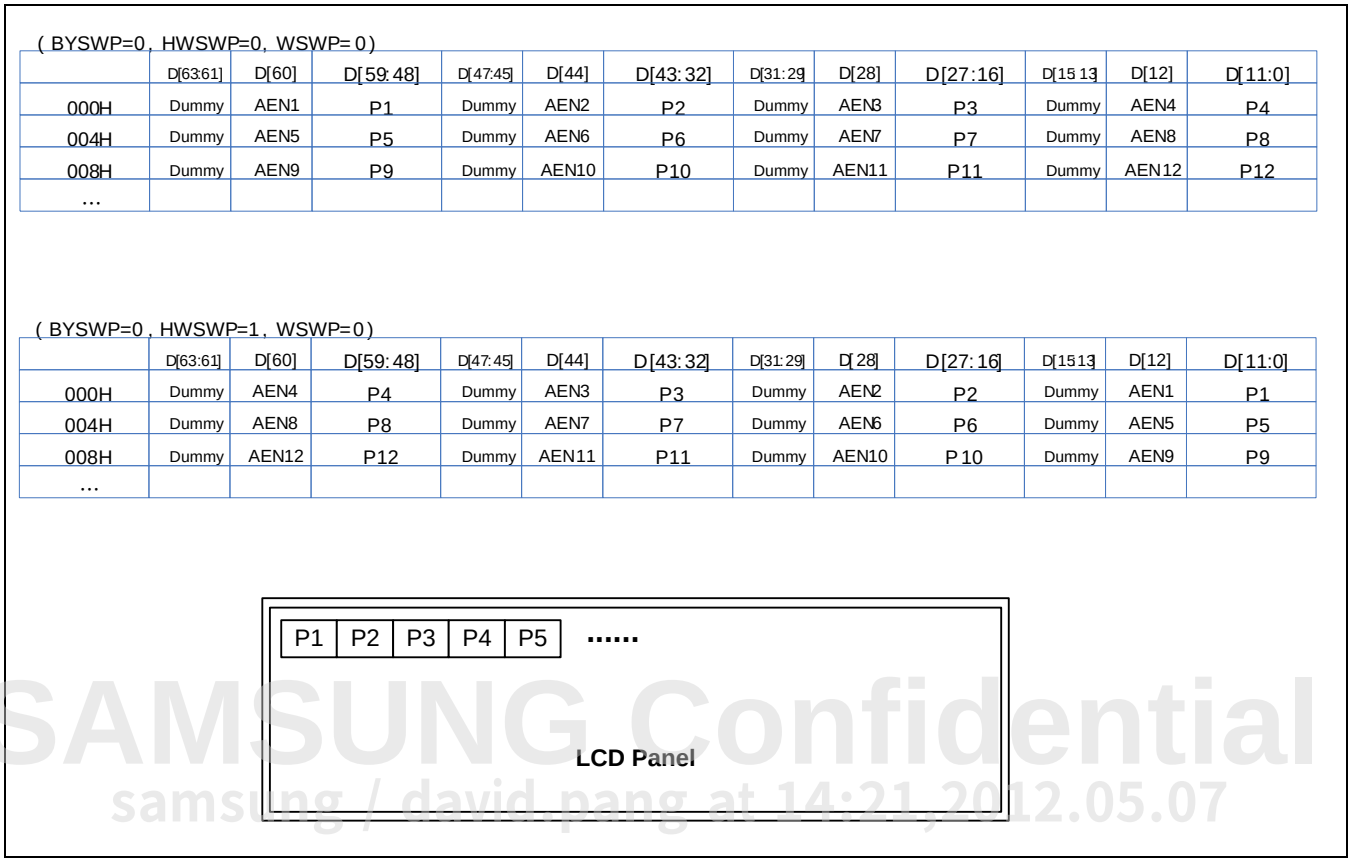


Figure 41-13 16 BPP (5:6:5) Display Types

41.3.3.2.9 13 BPP Display (A444)

Figure 41-14 illustrates the13 BPP display.



(BYSWP=0, HWSWP=0, WSWP=0)

	D[63:61]	D[60]	D[59:48]	D[47:45]	D[44]	D[43:32]	D[31:29]	D[28]	D[27:16]	D[15:13]	D[12]	D[11:0]
000H	Dummy	AEN1	P1	Dummy	AEN2	P2	Dummy	AEN3	P3	Dummy	AEN4	P4
004H	Dummy	AEN5	P5	Dummy	AEN6	P6	Dummy	AEN7	P7	Dummy	AEN8	P8
008H	Dummy	AEN9	P9	Dummy	AEN10	P10	Dummy	AEN11	P11	Dummy	AEN12	P12
…												

(BYSWP=0 , HWSWP=1 , WSWP=0)

	D[63:61]	D[60]	D[59:48]	D[47:45]	D[44]	D[43:32]	D[31:29]	D[28]	D[27:16]	D[15:13]	D[12]	D[11:0]
000H	Dummy	AEN4	P4	Dummy	AEN3	P3	Dummy	AEN2	P2	Dummy	AEN1	P1
004H	Dummy	AEN8	P8	Dummy	AEN7	P7	Dummy	AEN6	P6	Dummy	AEN5	P5
008H	Dummy	AEN12	P12	Dummy	AEN11	P11	Dummy	AEN10	P10	Dummy	AEN9	P9
…												

Figure 41-14 Memory Format of 13 BPP (A444) Display

NOTE:

1. AEN: Specifies the transparency value selection bit.
   AEN = 0: Selects ALPHA0.
   AEN = 1: Selects ALPHA1.
   When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects.
2. D[11:8] = Red data, D[7:4] = Green data, and D[3:0] = Blue data.
3. 16 BPP (4444) mode. (For more information, refer to the section on "SFR") Data has Alpha value.

(BYSWP=0, HWSWP=0, WSWP=0)

	D[63:60]	D[59:48]	D[47:44]	D[43:32]	D[31:28]	D[27:16]	D[15:12]	D[11:0]
000H	ALPHA1	P1	ALPHA2	P2	ALPHA3	P3	ALPHA4	P4
004H	ALPHA5	P5	ALPHA6	P6	ALPHA7	P7	ALPHA8	P8
008H	ALPHA9	P9	ALPHA10	P10	ALPHA11	P11	ALPHA12	P12
…								

41.3.3.2.10 8 BPP Display (A232)

Figure 41-15 illustrates the 8 BPP display.

(BYSWP=0, HWSWP=0, WSWP=0)

	D[63]	D[62:56]	D[55]	D[54:48]	D[47]	D[46:40]	D[39]	D[38:32]	D[31]	D[30:24]	D[23]	D[22:16]	D[15]	D[14:8]	D[7]	D[6:0]
000H	AEN	P1	AEN	P2	AEN	P3	AEN	P4	AEN	P5	AEN	P6	AEN	P7	AEN	P8
008H	AEN	P9	AEN	P10	AEN	P11	AEN	P12	AEN	P13	AEN	P14	AEN	P15	AEN	P16
010H	AEN	P17	AEN	P18	AEN	P19	AEN	P20	AEN	P21	AEN	P22	AEN	P23	AEN	P24
...																

(BYSWP=1, HWSWP=0, WSWP=0)

	D[63]	D[62:56]	D[55]	D[54:48]	D[47]	D[46:40]	D[39]	D[38:32]	D[31]	D[30:24]	D[23]	D[22:16]	D[15]	D[14:8]	D[7]	D[6:0]
000H	AEN	P8	AEN	P7	AEN	P6	AEN	P5	AEN	P4	AEN	P3	AEN	P2	AEN	P1
008H	AEN	P16	AEN	P15	AEN	P14	AEN	P13	AEN	P12	AEN	P11	AEN	P10	AEN	P9
010H	AEN	P24	AEN	P23	AEN	P22	AEN	P21	AEN	P20	AEN	P19	AEN	P18	AEN	P17
...																

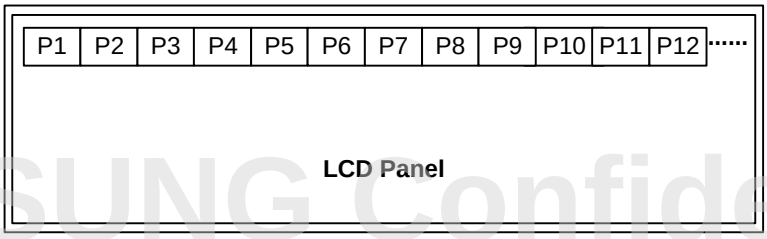


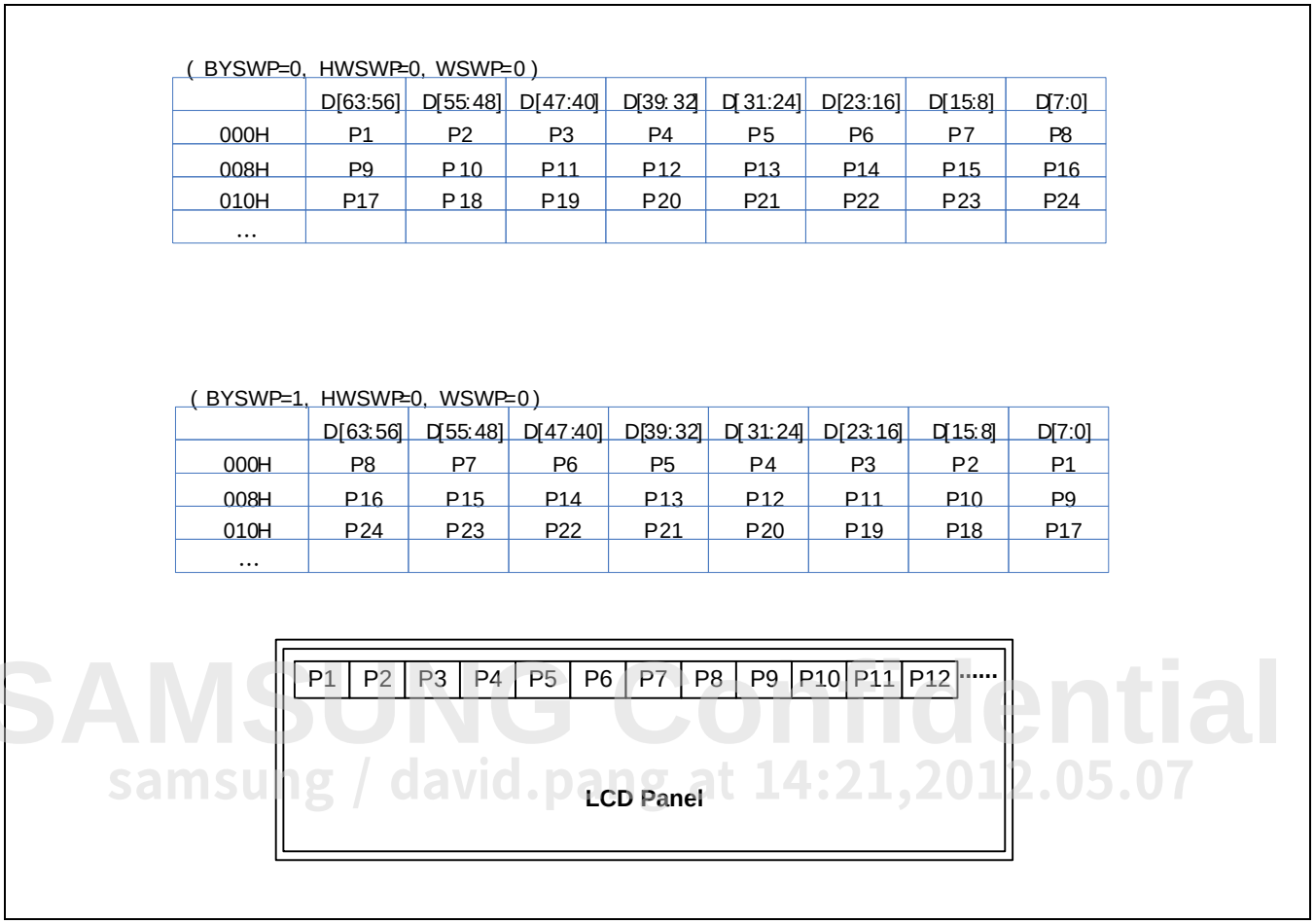
Figure 41-15 Memory Format of 8 BPP (A232) Display

NOTE:

1. AEN: Specifies the transparency value selection bit.
   AEN = 0: Selects ALPHA0.
   AEN = 1: Selects ALPHA1.
   When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects.
   SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
   For more information, refer to the section on "SFR".
2. D[6:5] = Red data, D[4:2] = Green data, and D[1:0] = Blue data.

41.3.3.2.11 8 BPP Display (Palette)

Figure 41-16 illustrates the 8 BPP display.



	D[63:56]	D[55:48]	D[47:40]	D[39:32]	D[31:24]	D[23:16]	D[15:8]	D[7:0]
000H	P1	P2	P3	P4	P5	P6	P7	P8
008H	P9	P10	P11	P12	P13	P14	P15	P16
010H	P17	P18	P19	P20	P21	P22	P23	P24
…								

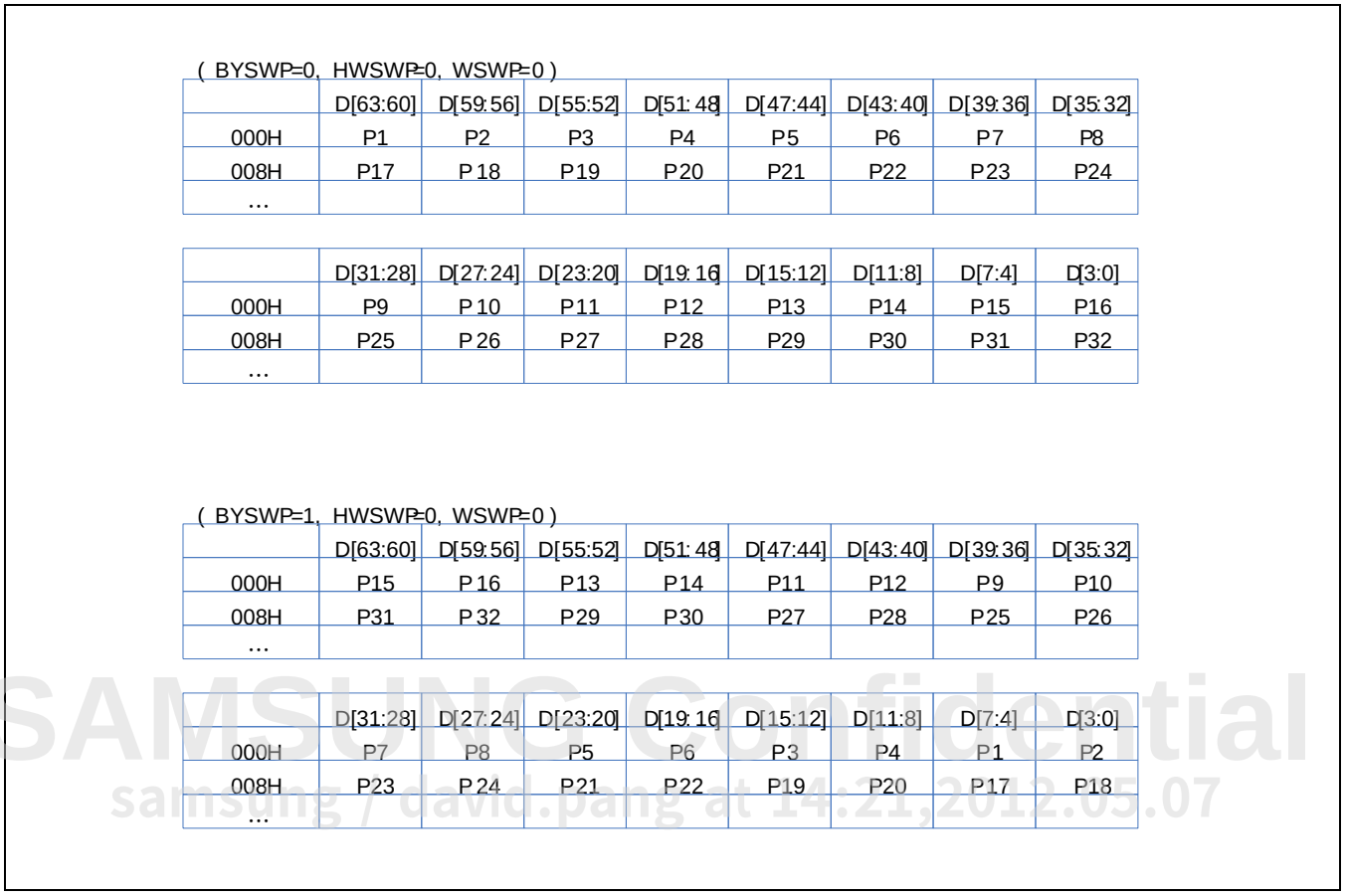
	D[63:56]	D[55:48]	D[47:40]	D[39:32]	D[31:24]	D[23:16]	D[15:8]	D[7:0]
000H	P8	P7	P6	P5	P4	P3	P2	P1
008H	P16	P15	P14	P13	P12	P11	P10	P9
010H	P24	P23	P22	P21	P20	P19	P18	P17
…								

Figure 41-16 Memory Format of 8 BPP Display

NOTE: AEN: Specifies the transparency value selection bit (with WPALCON: Palette output format).
AEN = 0: Selects ALPHA0.
AEN = 1: Selects ALPHA1.
When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects.
SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
For more information, refer to the section on "SFR".

41.3.3.2.12 4 BPP Display (Palette)

Figure 41-17 illustrates the 4 BPP display.



(BYSWP=0, HWSWP=0, WSWP=0)

	D[63:60]	D[59:56]	D[55:52]	D[51:48]	D[47:44]	D[43:40]	D[39:36]	D[35:32]
000H	P1	P2	P3	P4	P5	P6	P7	P8
008H	P17	P18	P19	P20	P21	P22	P23	P24
…								

	D[31:28]	D[27:24]	D[23:20]	D[19:16]	D[15:12]	D[11:8]	D[7:4]	D[3:0]
000H	P9	P10	P11	P12	P13	P14	P15	P16
008H	P25	P26	P27	P28	P29	P30	P31	P32
…								

(BYSWP=1, HWSWP=0, WSWP=0)

	D[63:60]	D[59:56]	D[55:52]	D[51:48]	D[47:44]	D[43:40]	D[39:36]	D[35:32]
000H	P15	P16	P13	P14	P11	P12	P9	P10
008H	P31	P32	P29	P30	P27	P28	P25	P26
…								

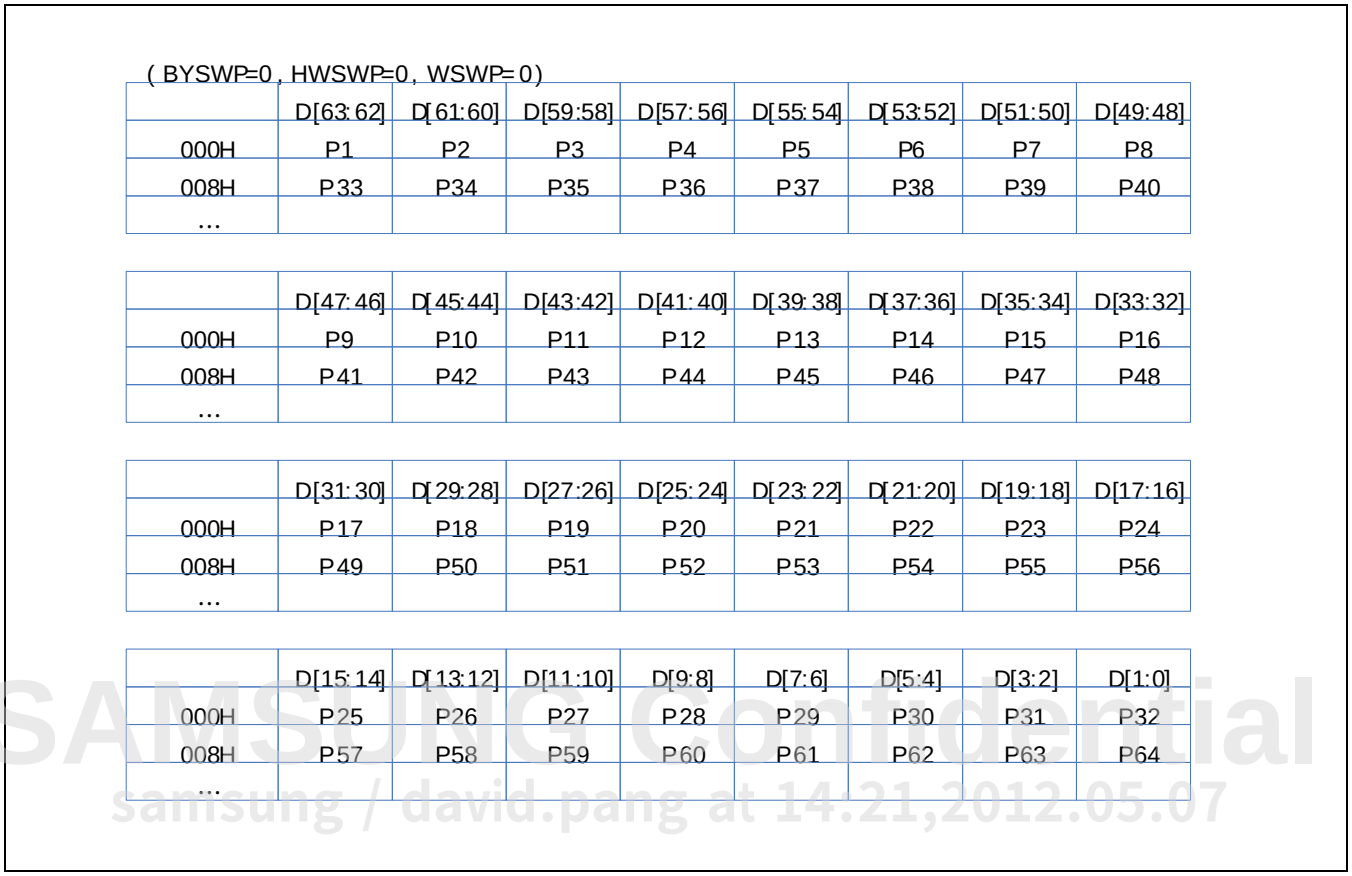
	D[31:28]	D[27:24]	D[23:20]	D[19:16]	D[15:12]	D[11:8]	D[7:4]	D[3:0]
000H	P7	P8	P5	P6	P3	P4	P1	P2
008H	P23	P24	P21	P22	P19	P20	P17	P18
…								

Figure 41-17 Memory Format of 4 BPP Display

NOTE: AEN: Specifies the transparency value selection bit (with WPALCON: Palette output format)
AEN = 0: Selects ALPHA0.
AEN = 1: Selects ALPHA1.
When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects..
SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
For more information, refer to the section on "SFR".

41.3.3.2.13 2 BPP Display (Palette)

Figure 41-18 illustrates the 2 BPP display.



(BYSWP=0 , HWSWP=0 , WSWP= 0)

	D[63:62]	D[61:60]	D[59:58]	D[57:56]	D[55:54]	D[53:52]	D[51:50]	D[49:48]
000H	P1	P2	P3	P4	P5	P6	P7	P8
008H	P33	P34	P35	P36	P37	P38	P39	P40
…								

	D[47:46]	D[45:44]	D[43:42]	D[41:40]	D[39:38]	D[37:36]	D[35:34]	D[33:32]
000H	P9	P10	P11	P12	P13	P14	P15	P16
008H	P41	P42	P43	P44	P45	P46	P47	P48
…								

	D[31:30]	D[29:28]	D[27:26]	D[25:24]	D[23:22]	D[21:20]	D[19:18]	D[17:16]
000H	P17	P18	P19	P20	P21	P22	P23	P24
008H	P49	P50	P51	P52	P53	P54	P55	P56
…								

	D[15:14]	D[13:12]	D[11:10]	D[9:8]	D[7:6]	D[5:4]	D[3:2]	D[1:0]
000H	P25	P26	P27	P28	P29	P30	P31	P32
008H	P57	P58	P59	P60	P61	P62	P63	P64
…								

Figure 41-18 Memory Format of 2 BPP Display

NOTE: AEN: Specifies the transparency value selection bit (with WPALCON: Palette output format).
AEN = 0: Selects ALPHA0.
AEN = 1: Selects ALPHA1.
When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects.
SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
For more information, refer to the section on "SFR".

41.3.4 Color Space Conversion

Color Space Conversion (CSC) section includes:

- Color Space Conversion YCbCr to RGB (CSCY2R)
- Color Space Conversion RGB to YCbCr (CSCR2Y)

41.3.4.1 Color Space Conversion YCbCr to RGB (CSCY2R)

This section describes the color space conversion YCbCr to RGB.

CSCY2 R (Color Space Conversion Y to R)

		601	709
Wide	R =	Y + 1. 371(Cr- 128)	Y + 1. 540(Cr- 128)
	G =	Y ? 0. 698(Cr ? 128) ? 0. 336(Cb ? 128)	Y ? 0. 459(Cr ? 128) ? 0. 183(Cb ? 128)
	B =	Y + 1. 732(Cb ? 128)	Y + 1. 816(Cb ? 128)
Narrow	R =	1. 164(Y ? 16) + 1. 596(Cr- 128)	1. 164(Y ? 16) + 1. 793(Cr- 128)
	G =	1. 164(Y ? 16) ? 0. 813(Cr ? 128) ? 0. 391(Cb ? 128)	1. 164(Y ? 16) ? 0. 534(Cr ? 128) ? 0 . 213(Cb ? 128)
	B =	1. 164(Y ? 16) + 2. 018(Cb ? 128)	1. 164(Y ? 16) + 2. 115(Cb ? 128)

NOTE: "Wide" means RGB data has a nominal range from 16 to 235. On the other hand, "Narrow" means RGB data has a nominal range from 0 to 255.

Coefficient approximation is:

$1.164 = (2^7 + 2^4 + 2^2 + 2^0) >> 7$
$1.596 = (2^7 + 2^6 + 2^3 + 2^2) >> 7$
$0.813 = (2^6 + 2^5 + 2^3) >> 7$
$0.391 = (2^5 + 2^4 + 2^1) >> 7$
$2.018 = (2^8 + 2^1) >> 7$

$1,793 = (2^7 + 2^6 + 2^5 + 2^2 + 2^1) >> 7$
$0.534 = (2^6 + 2^2) >> 7$
$0.213 = (2^4 + 2^3 + 2^1 + 2^0) >> 7$
$2.115 = (2^8 + 2^3 + 2^2 + 2^1 + 2^0) >> 7$

$1.371 = (2^8 + 2^6 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0) >> 8$
$0.698 = (2^7 + 2^5 + 2^4 + 2^1 + 2^0) >> 8$
$0.336 = (2^6 + 2^4 + 2^2 + 2^1) >> 8$
$1.732 = (2^8 + 2^7 + 2^5 + 2^4 + 2^3 + 2^1 + 2^0) >> 8$

$1.540 = (2^8 + 2^7 + 2^3 + 2^1) >> 8$
$0.459 = (2^6 + 2^5 + 2^4 + 2^2 + 2^1) >> 8$
$0.183 = (2^5 + 2^3 + 2^2 + 2^1 + 2^0) >> 8$
$1.816 = (2^8 + 2^7 + 2^6 + 2^4 + 2^0) >> 8$

41.3.4.2 Color Space Conversion RGB to YCbCr (CSCR2Y)

This section describes the Color Space Conversion RGB to YCbCr.

CSCR2Y (Color Space Conversion R to Y)

		601	709
Wide	Y=	0.299R + 0.587G + 0.114B	0.213R + 0.715G + 0.072B
	Cb=	-0.172R - 0.339G + 0.511B + 128	-0.117R - 0.394G + 0.511B + 128
	Cr=	0.511R - 0.428G - 0.083B + 128	0.511R - 0.464G - 0.047B + 128
Narrow	Y=	0.257R + 0.504G + 0.098B + 16	0.183R + 0.614G + 0.062B + 16
	Cb=	-0.148R - 0.291G + 0.439B + 128	-0.101R - 0.338G + 0.439B + 128
	Cr=	0.439R - 0.368G - 0.071B + 128	0.439R - 0.399G - 0.040B + 128

NOTE: "Wide" means RGB data has a nominal range from 16 to 235. On the other hand, "Narrow" means RGB data has a nominal range from 0 to 255.

Coefficient approximation is:

$0.257 = (2^6 + 2^1) >> 8$
$0.504 = (2^7 + 2^0) >> 8$
$0.098 = (2^4 + 2^3 + 2^0) >> 8$
$0.148 = (2^5 + 2^2 + 2^1) >> 8$
$0.291 = (2^6 + 2^3 + 2^1) >> 8$
$0.439 = (2^6 + 2^5 + 2^4) >> 8$
$0.368 = (2^7 - 2^5 - 2^1) >> 8$
$0.071 = (2^4 + 2^1) >> 8$

$0.299 = (2^6 + 2^3 + 2^2 + 2^0) >> 8$
$0.587 = (2^7 + 2^4 + 2^2 + 2^1) >> 8$
$0.114 = (2^4 + 2^3 + 2^2 + 2^0) >> 8$
$0.172 = (2^5 + 2^3 + 2^2) >> 8$
$0.339 = (2^6 + 2^4 + 2^3 - 2^0) >> 8$
$0.511 = (2^7 + 2^1 + 2^0) >> 8$
$0.428 = (2^7 - 2^4 - 2^1) >> 8$
$0.083 = (2^4 + 2^2 + 2^0) >> 8$

$0.183 = (2^5 + 2^3 + 2^2 + 2^1 + 2^0) >> 8$
$0.614 = (2^7 + 2^4 + 2^3 + 2^2 + 2^0) >> 8$
$0.062 = (2^4) >> 8$
$0.101 = (2^4 + 2^3 + 2^1) >> 8$
$0.338 = (2^6 + 2^4 + 2^2 + 2^1 + 2^0) >> 8$

$0.399 = (2^6 + 2^5 + 2^2 + 2^1) >> 8$
$0.040 = (2^3 + 2^1) >> 8$

$0.213 = (2^5 + 2^4 + 2^2 + 2^1 + 2^0) >> 8$
$0.715 = (2^7 + 2^5 + 2^4 + 2^2 + 2^1 + 2^0) >> 8$
$0.072 = (2^4 + 2^1) >> 8$
$0.117 = (2^4 + 2^3 + 2^2 + 2^1) >> 8$
$0.394 = (2^6 + 2^5 + 2^2 + 2^0) >> 8$

$0.464 = (2^6 + 2^5 + 2^4 + 2^2 + 2^1 + 2^0) >> 8$
$0.047 = (2^3 + 2^2) >> 8$

41.3.5 Palette Usage

The Palette Usage section includes:

- Palette Configuration and Format Control
- Palette Read/Write

41.3.5.1 Palette Configuration and Format Control

The display controller supports 256-color palette to select color mapping. You can select up to 256 colors from 32-bit colors using these formats.

256 color palette consists of 256 (depth) × 32-bit SPSRAM. Palette supports 8:8:8, 6:6:6, 5:6:5 (R: G: B), and other formats.

For Example:

See A:5:5:5 format, write palette, as illustrated in *Figure 41-20*.

1. Connect VD pin to TFT LCD panel (R (5) = VD[23:19], G (5) = VD[15:11], and B (5) = VD[7:3]).
2. AEN bit controls the blending function, enable or disable.
3. Set WPALCON (W1PAL, case window0) register to 0'b101. The 32-bit (8:8:8:8) format has an alpha value directly, without using alpha value register (ALPHA_0/1).

Figure 41-19 illustrates the 32 BPP (8:8:8:8) palette data format.

INDEX/ Bit Pos.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00 h	ALPHA								R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
01 h	ALPHA								R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
.....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
FFh	ALPHA								R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
Number of VD	-	-	-	-	-	-	-	-	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 41-19 32 BPP (8:8:8:8) Palette Data Format

Figure 41-20 illustrates the 25 BPP (A: 8:8:8) palette data format.

INDEX/ Bit Pos.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00 h	-	-	-	-	-	-	-	AEN	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
01 h	-	-	-	-	-	-	-	AEN	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
.....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
FFh	-	-	-	-	-	-	-	AEN	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
Number of VD	-	-	-	-	-	-	-	-	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 41-20 25 BPP (A: 8:8:8) Palette Data Format

Figure 41-21 illustrates the 19 BPP (A: 6:6:6) palette data format.

INDEX/ Bit Pos.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00 h	-	-	-	-	-	-	-	-	-	-	-	-	-	AEN	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
01 h	-	-	-	-	-	-	-	-	-	-	-	-	-	AEN	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
.....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
FFh	-	-	-	-	-	-	-	-	-	-	-	-	-	AEN	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
Number of VD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	23	22	21	20	19	18	15	14	13	12	11	10	7	6	5	4	3	2

Figure 41-21 19 BPP (A: 6:6:6) Palette Data Format

Figure 41-22 illustrates the 16 BPP (A: 5:5:5) palette data format

INDEX / Bit Pos.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00 h	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AEN	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
01 h	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AEN	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
……	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
FFh	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AEN	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
Number of VD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	23	22	21	20	19	15	14	13	12	11	7	6	5	4	3

Figure 41-22 16 BPP (A: 5:5:5) Palette Data Format

41.3.5.2 Palette Read/Write

You should not access palette memory when the Vertical Status (VSTATUS) register has an ACTIVE status. You should check the VSTATUS to do Read/Write operation on the palette.

41.3.6 Window Blending

This section includes:

- Overview of Window Blending
- Blending Diagram
- Color-Key Function
- Blending and Color-Key Function

41.3.6.1 Overview of Window Blending

Window blending is the main function of VPRCS module. Display controller comprises of five window layers (win 0 to win 4).

Example 41-1 Application

The system uses
win0 as OS window, full TV screen window, and so on.
win1 as small (next channel) TV screen with win2 as menu.
win3 as caption.
win4 as channel information.
win3 and win4 have color limitation while using color index with Color LUT. This feature enhances the system performance by reducing the data rate of total system.

Example 41-2 Total Five Windows

```
win0 (base): Local/YCbCr, RGB without palette)
win1 (Overlay1): RGB with palette
win2 (Overlay2): RGB with palette
win3 (Caption): RGB (1/2/4) with 16-level Color LUT
win4 (Cursor): RGB (1/2) with 4-level Color LUT
```

Overlay Priority

Win4 > Win3 > Win2 > Win1 > Win0

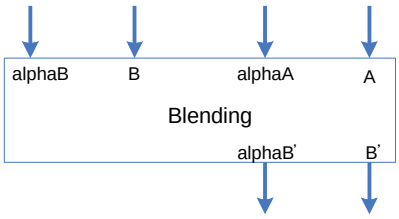
Color Key

24-bit RGB format should set the register value to color key register.

Example 41-3 Blending Equation

```
<Data blending>
Win01 (R,G,B) = Win0 (R,G,B) × b1 + Win1 (R,G,B) × a1
Win012 (R/G/B) = Win01 (R/G/B) × b2 + Win2 (R/G/B) × a2
Win0123 (R/G/B) = Win012 (R/G/B) × b3 + Win3 (R/G/B) × a3
WinOut (R/G/B) = Win0123 (R/G/B) × b4 + Win4 (R/G/B) × a4

, where,

Win0 (R) = Window 0's Red data,
Win0 (G) = Window 0's Green data,
Win0 (B) = Window 0's Blue data,
Win1 (R) = Window 1's Red data,
...
b1 = Background's Data blending equation1 factor,
a1 = Foreground's Data blending equation1 factor,
b2 = Background's Data blending equation2 factor,
a2 = Foreground's Data blending equation2 factor,

<Alpha value blending>
AR (G,B)01 = AR (G,B)0 × q1 + AR (G,B)1 × p1
AR (G,B)012 = AR (G,B)01 × q2 + AR (G,B)2 × p2
AR (G,B)0123 = AR (G,B)012 × q3 + AR (G,B)3 × p3

, where,

AR0 = Window 0's Red blending factor,
AG0 = Window 0's Green blending factor,
AB0 = Window 0's Blue blending factor,
AR1 = Window 1's Red blending factor,...
AR01 = Window01's Red blending factor (alpha value blending between AR0 and AR1),
AG01 = Window01's Green blending factor (alpha value blending between AG0 and AG1),
AB01 = Window01's Blue blending factor (alpha value blending between AB0 and AB1),
AR012 = Window012's Red blending factor (alpha value blending between AR01 and AR2),
...
q1 = Background's Alpha value blending equation1 factor,
p1 = Foreground's Alpha value blending equation1 factor,
q2 = Background's Alpha value blending equation2 factor,
p2 = Foreground's Alpha value blending equation2 factor,…
```

Figure 41-23 illustrates the blending equation.



B' = a x A + b x B

alphaB' = p x alphaA + q x alphaB

a/b/p/q = 0 or 1 or alphaA or 1- alphaA or alphaB or 1- alphaB or A or 1- A or B or 1- B or ALPHA0

'a' is controlled by 'A_FUNC' @ BLENDEQ register
'b' " by 'B_FUNC' @ BLENDEQ register
'p' " by 'P_FUNC' @ BLENDEQ register
'q' " by 'Q_FUNC' @ BLENDEQ register

Figure 41-23 Blending Equation

Example 41-4 Default Blending Equation

```
<Data blending>
B' = B × (1 - alphaA) + A × alphaA
Alpha value blending>
alphaB' = 0 (= alphaB × 0 + alphaA × 0)
```

41.3.6.2 Blending Diagram

The display controller can blend five layers for one pixel at the same time. ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B registers control the alpha value (blending factor), which you can implement for each window layer and color (R, G, B).

The example below shows the R (Red) output using ALPHA_R value of each window.

All windows have two kinds of alpha blending value:

Alpha value that enables transparency (AEN value == 1) Alpha value that disables transparency (AEN value == 0) If you enable WINEN_F and BLD_PIX and disable ALPHA_SEL, then it selects the AR. The equation to select the AR is:

- AR = (Pixel (R)'s AEN value == 1'b1) ? Reg (ALPHA1_R): Reg (ALPHA0_R);
- AG = (Pixel (G)'s AEN value == 1'b1) ? Reg (ALPHA1_G): Reg (ALPHA0_G);
- AB = (Pixel (B)'s AEN value == 1'b1) ? Reg (ALPHA1_B): Reg (ALPHA0_B);
  (where, BLD_ PIX == 1, ALPHA_SEL == 0)

If you enable WINEN_F and disable BLD_PIX, then the ALPHA_SEL ALPHA0 controls the AR. AEN bit information is not used anymore.

Figure 41-24 illustrates the blending diagram.

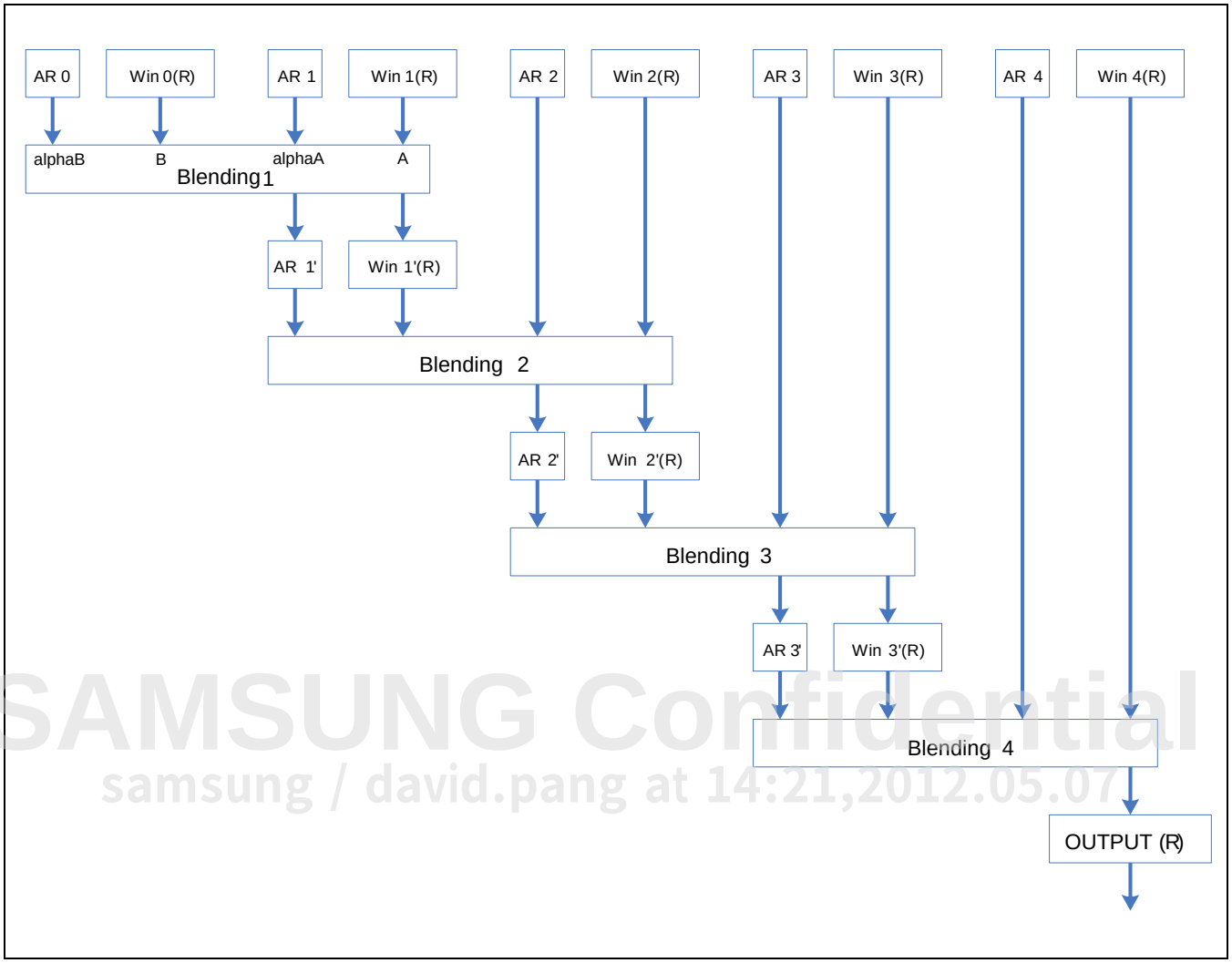


Figure 41-24 Blending Diagram

Example 41-5 Window Blending Factor Decision

Window n's blending factor decision (n = 0, 1, 2, 3, 4). For more information, refer to the section on "SFR".

Figure 41-25 illustrates the blending factor decision.

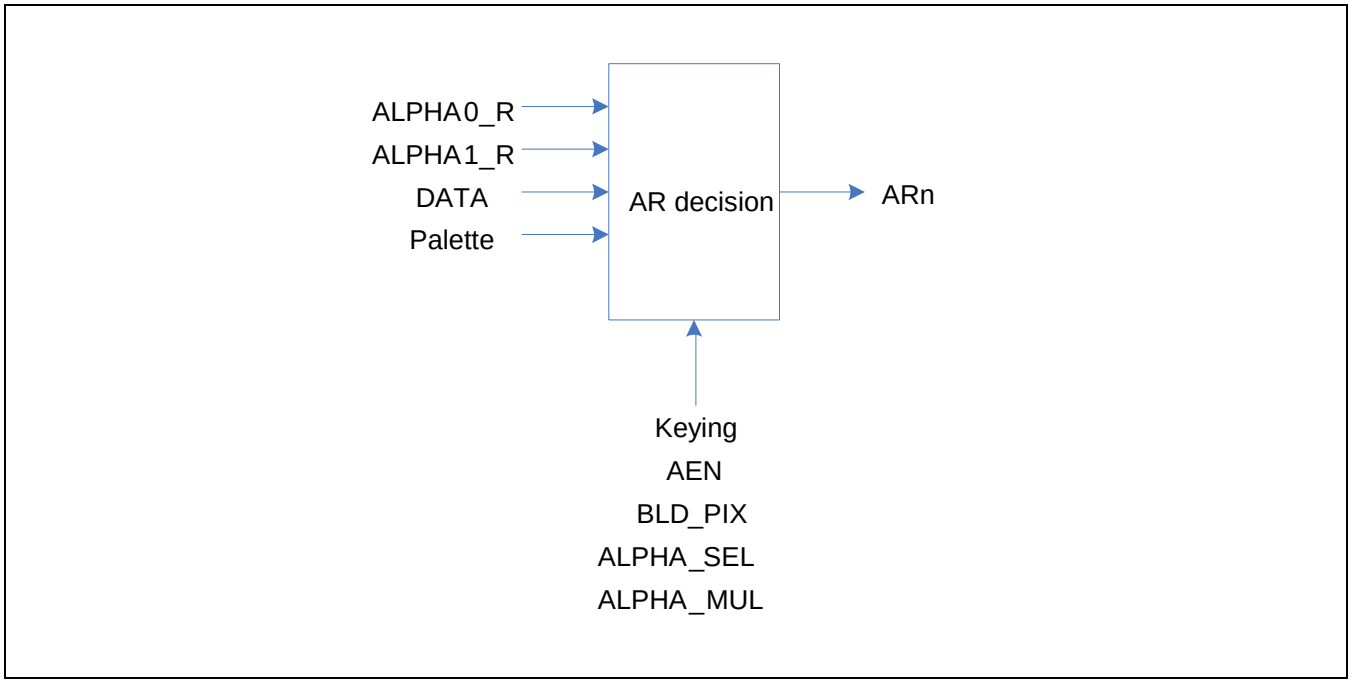


Figure 41-25 Blending Factor Decision

NOTE: If you use DATA[15:12] (BPPMODE_F = b'1110, ARGB4444 format) to blend, then the alpha value is {DATA[15:12], DATA [15:12]} (4-bit → 8-bit expanding).

41.3.6.3 Color-Key Function

The Color-Key function in display controller supports various effects for image mapping. For special functionality, the Color-Key register that specifies the color image of OSD layer is substituted by the background image-either as cursor image or preview image of the camera.

Figure 41-26 illustrates the Color-Key function configurations.

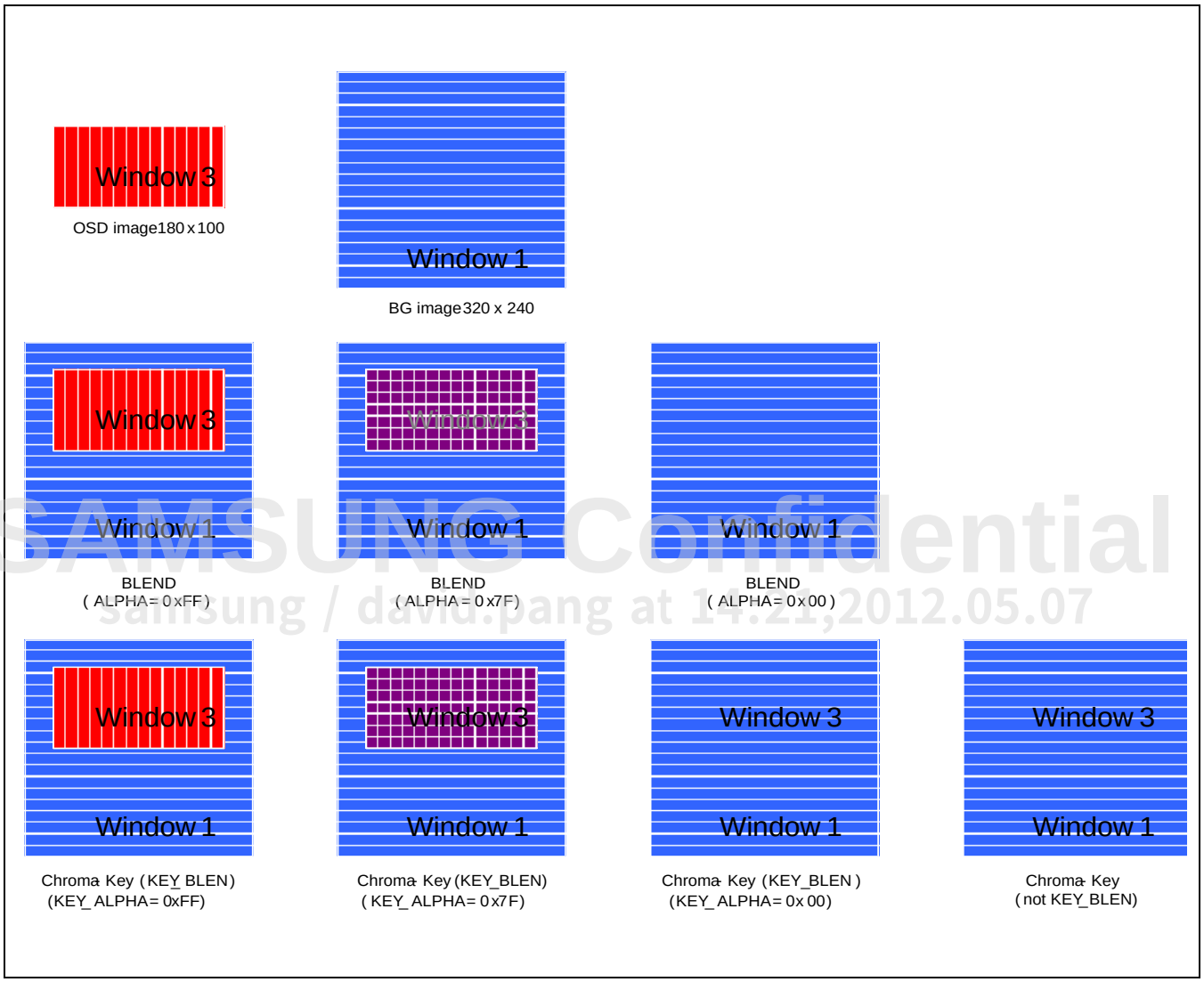


Figure 41-26 Color-Key Function Configurations

41.3.6.4 Blending and Color-Key Function

The display controller supports simultaneous blending function-with two transparency factors and Color-Key function in the same window.

Figure 41-27 illustrates the blending and Color-Key function.

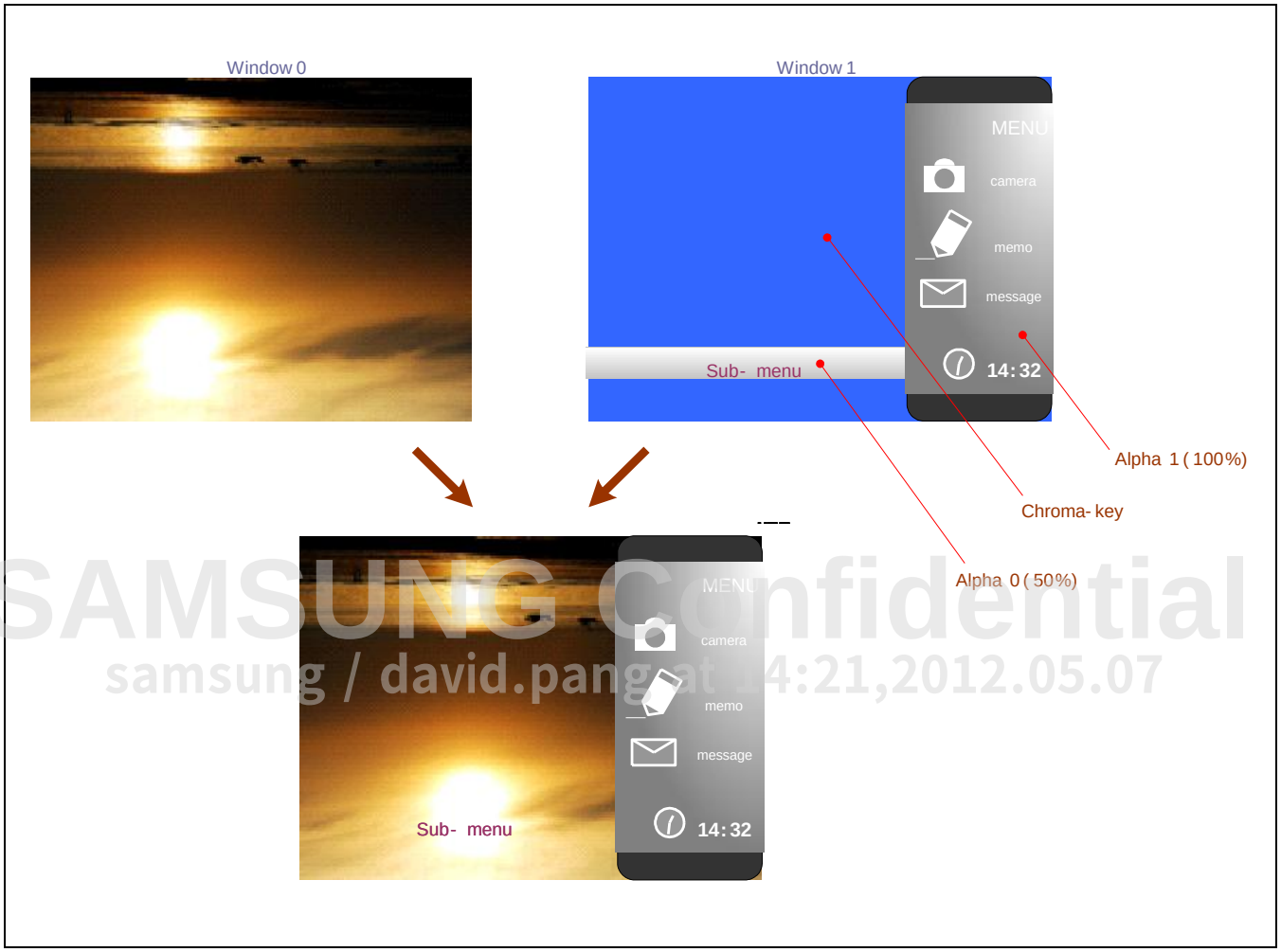


Figure 41-27 Blending and Color-Key Function

Figure 41-28 illustrates the blending decision diagram.

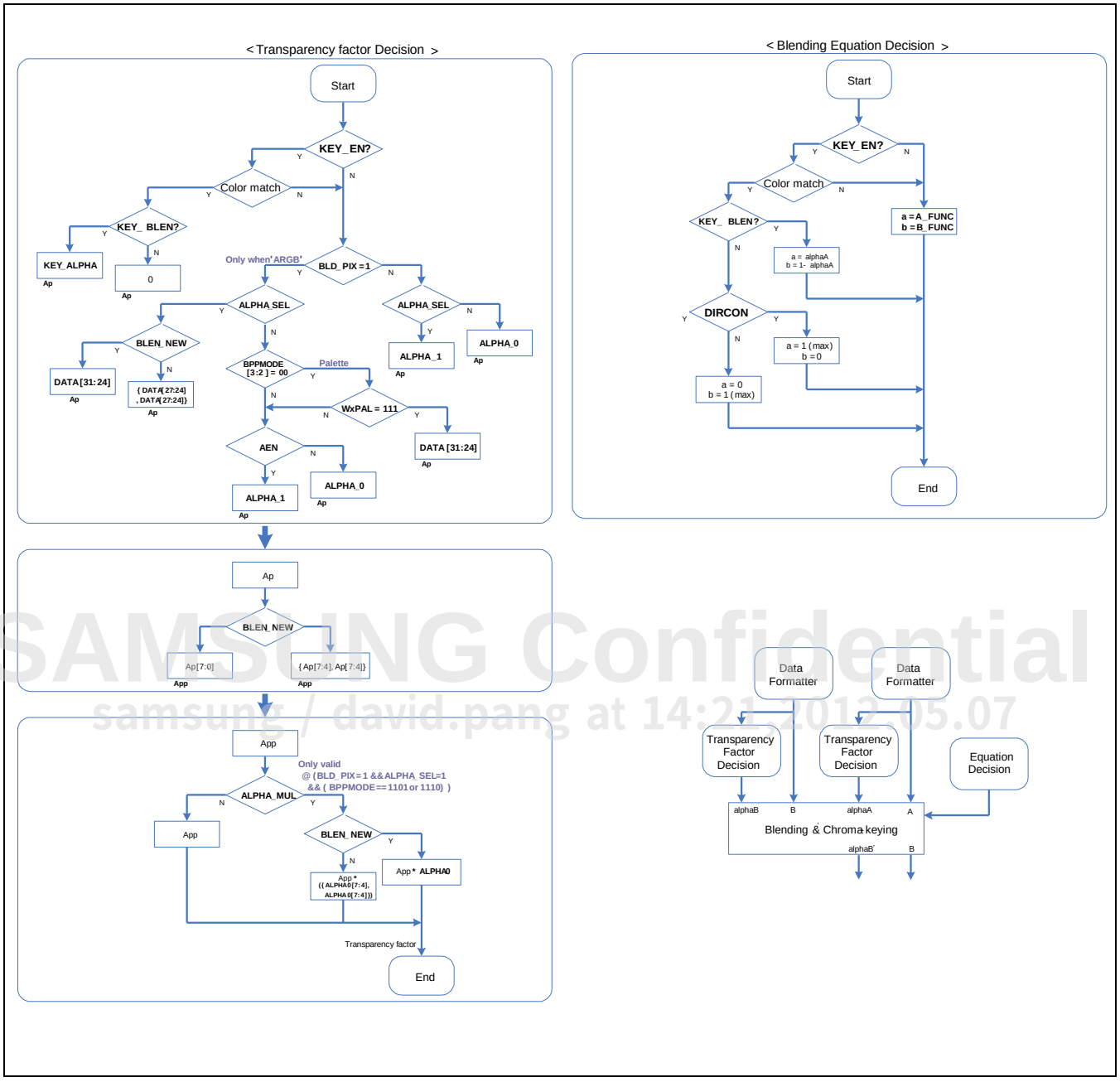


Figure 41-28 Blending Decision Diagram

41.3.7 Image Enhancement

Image Enhancement section includes:

- Overview of Image Enhancement
- Gamma Control (64 STEP)
- Gamma Control (16 STEP)
- Color Gain Control
- Hue Control (TBD)
- Pixel Compensation Control

41.3.7.1 Overview of Image Enhancement

Image enhancement is one of the primary functions of VPRCS module. The display controller supports Gamma, Hue, Color Gain, and Pixel Compensation Control functions.

Figure 41-29 illustrates the image enhancement flow.

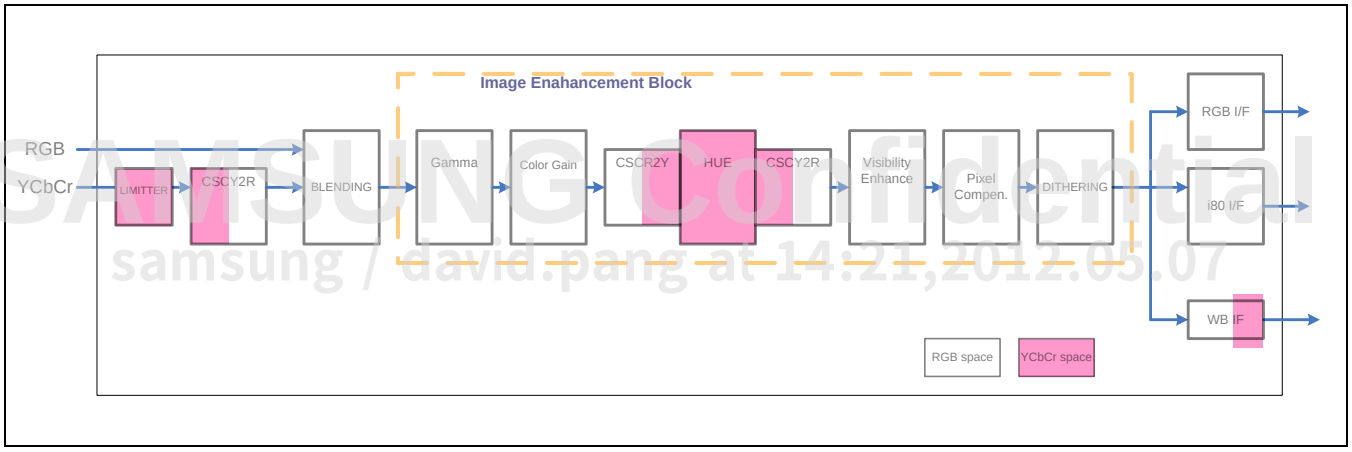


Figure 41-29 Image Enhancement Flow

41.3.7.2 Gamma Control (64 STEP)

Gamma control comprises of 65 LUT registers. Piecewise-linear operation determines the output value between two LUT registers. The output value saturates at 255.

Figure 41-30 illustrates the image enhancement flow.

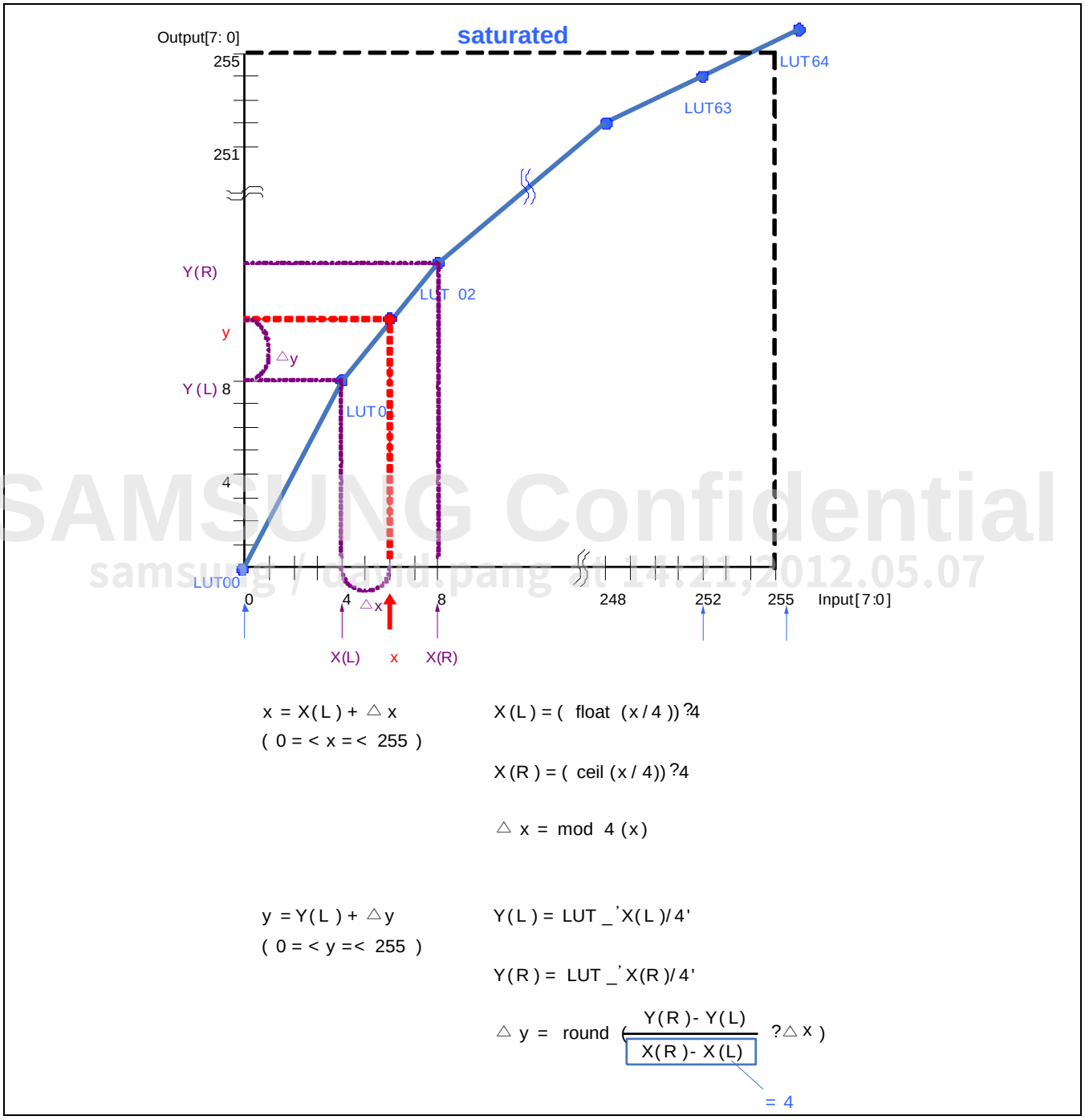


Figure 41-30 Image Enhancement Flow

41.3.7.3 Gamma Control (16 STEP)

Gamma control comprises of 17 × 3 (separate R, G, B) LUT registers. Piecewise-linear operation determines the output value between two LUT registers. The output value saturates at 255.

Figure 41-31 illustrates the image enhancement flow.

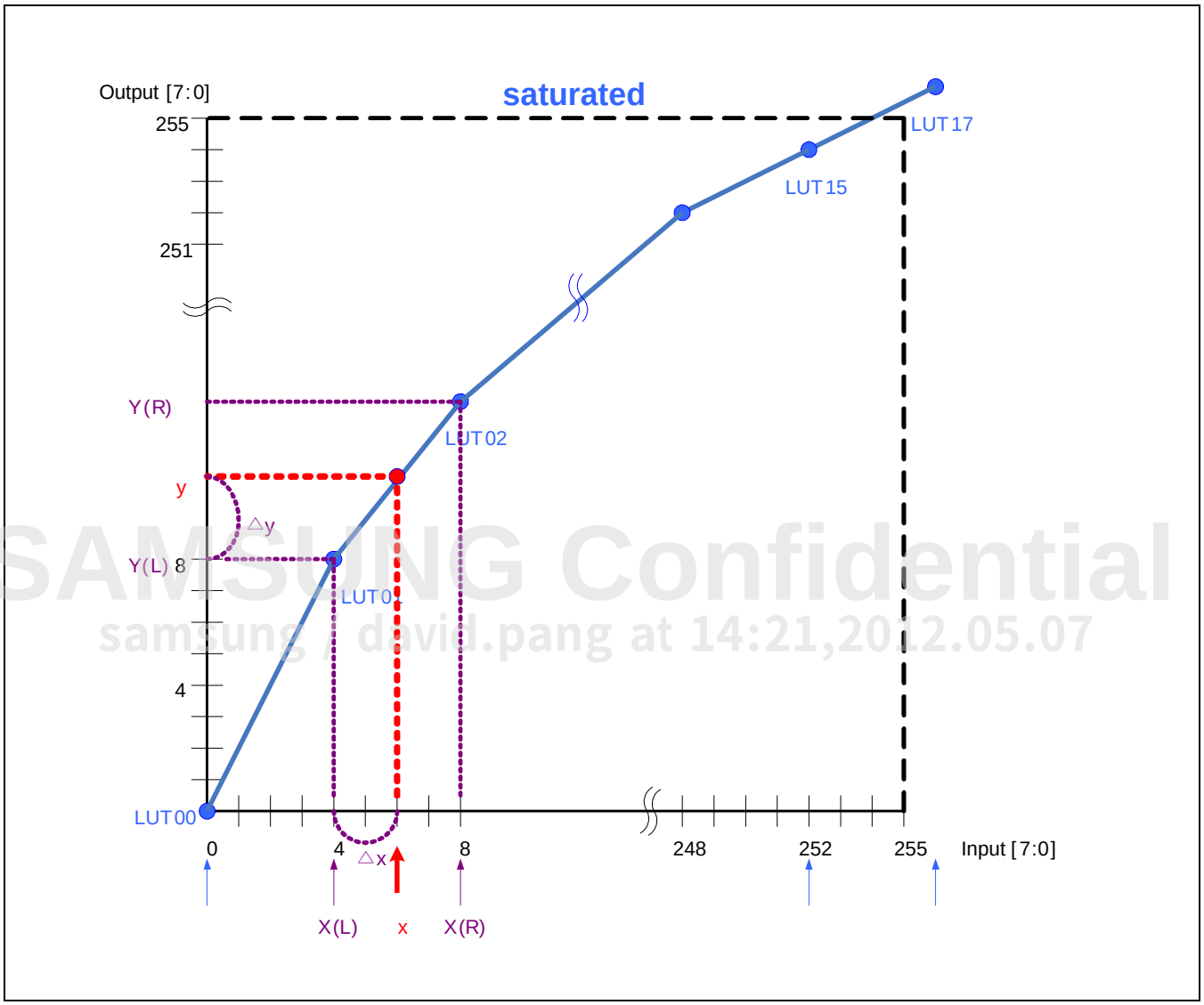


Figure 41-31 Image Enhancement Flow

41.3.7.4 Color Gain Control

The Color Gain Control comprises of three registers for each color: Red, Green, and Blue. The maximum value of Color Gain is 3.99609375 (approximately 4) and it has an 8-bit resolution.

- R (color gain) = R × CG_RGAIN
- G (color gain) = G × CG_GGAIN
- B (color gain) = B × CG_BGAIN

CG_R (G, B) GAIN comprises of 2-bit integer and 8-bit fraction.

The output value saturates at 255 (maximum value is approximately four with 8-bit resolution).

41.3.7.5 Hue Control (TBD)

Hue Control comprises of eight registers for coefficients of Hue matrix.

- Cb<hue> = CBG0 × (Cb + OFFSET_IN) + CBG1 × (Cr + OFFSET_IN) + OFFSET_OUT
- Cr<hue> = CRG0 × (Cb + OFFSET_IN) + CRG1 × (Cr + OFFSET_IN) + OFFSET_OUT
  (Generally, OFFSET_IN is "– 128" and OFFSET_OUT is "+ 128".)
- CBG0 = ((Cb + OFFSET_IN) ≥ 0 & (Cr + OFFSET_IN) ≥ 0) ? CBG0_1:
  ((Cb + OFFSET_IN) ≥ 0 & ! (Cr + OFFSET_IN) ≥ 0) ? CBG0_2:
  (! (Cb + OFFSET_IN) ≥ 0 & ! (Cr + OFFSET_IN) ≥ 0) ? CBG0_3: CBG0_4;
- CBG1 = ((Cb + OFFSET_IN) ≥ 0 & (Cr + OFFSET_IN) ≥ 0) ? CBG1_1:
  ((Cb + OFFSET_IN) ≥ 0 & ! (Cr + OFFSET_IN) ≥ 0) ? CBG1_2:
  (! (Cb + OFFSET_IN) ≥ 0 & ! (Cr + OFFSET_IN) ≥ =0) ? CBG1_3: CBG1_4;
- CRG0 = ((Cb + OFFSET_IN) ≥ 0 & (Cr + OFFSET_IN) ≥ =0) ? CRG0_1:
  ((Cb + OFFSET_IN) ≥ 0 & ! (Cr + OFFSET_IN) ≥ 0) ? CRG0_2:
  (! (Cb + OFFSET_IN) ≥ 0 & ! (Cr + OFFSET_IN) ≥ 0) ? CRG0_3: CRG0_4;
- CRG1 = ((Cb + OFFSET_IN) ≥ 0 & (Cr + OFFSET_IN) ≥ 0) ? CRG1_1:
  ((Cb + OFFSET_IN) ≥ 0 & ! (Cr + OFFSET_IN) ≥ 0) ? CRG1_2:
  (! (Cb + OFFSET_IN) ≥ 0 & ! (Cr + OFFSET_IN) ≥ 0) ? CRG1_3: CRG1_4;

Figure 41-32 illustrates the Hue coefficient decision.

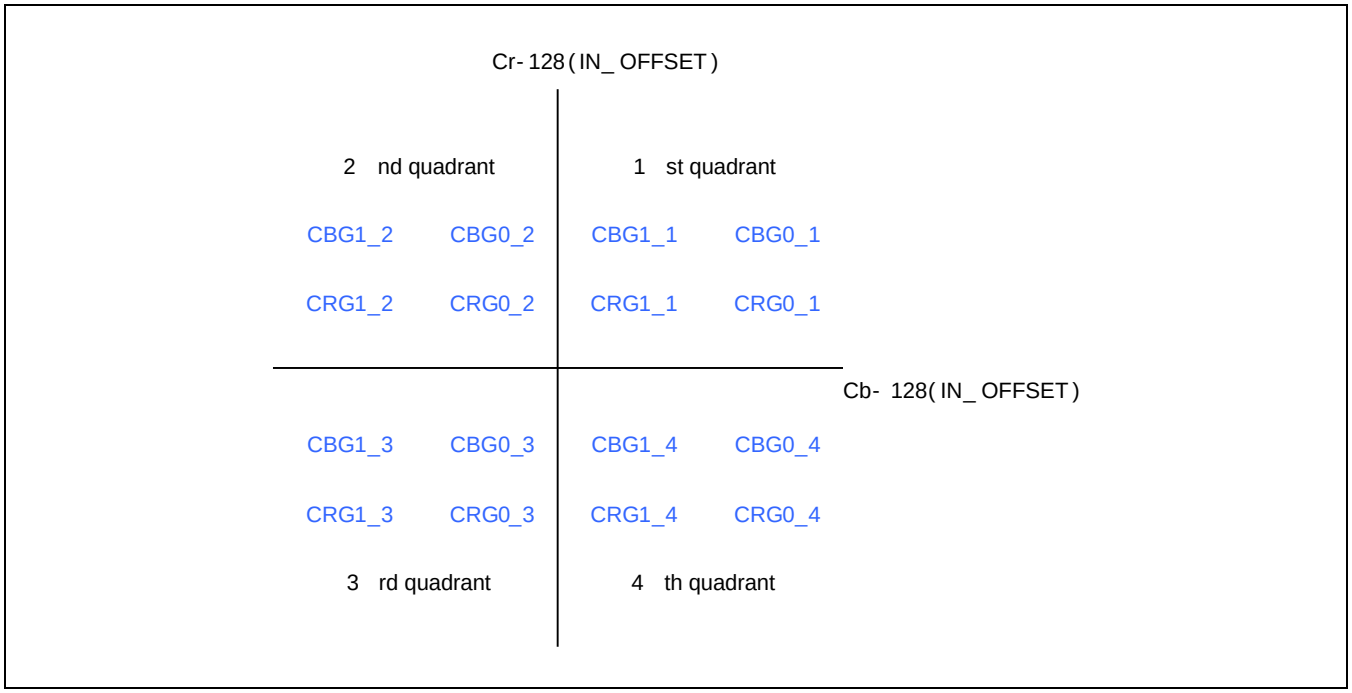


Figure 41-32 Hue Coefficient Decision

Figure 41-33 illustrates the block diagram of Hue Control.

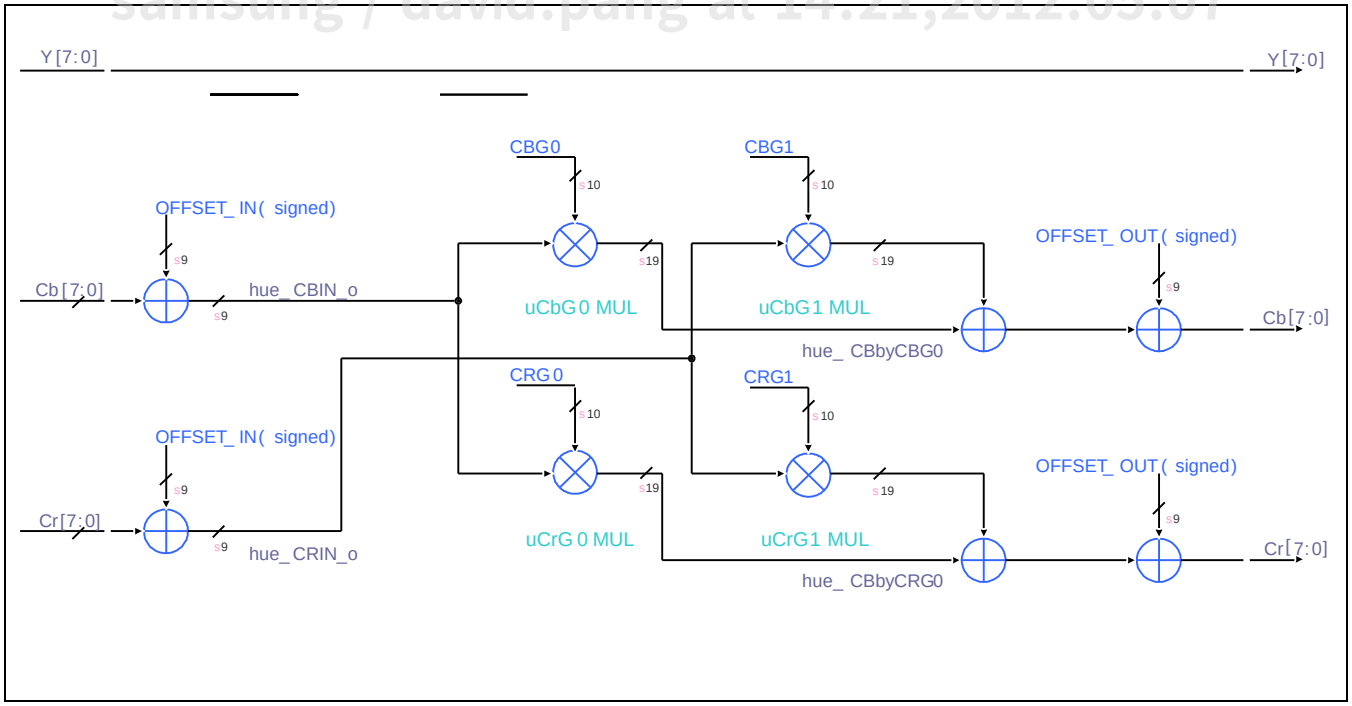


Figure 41-33 Hue Control Block Diagram

41.3.7.6 Pixel Compensation Control

The purpose of Pixel Compensation Control is to compensate data for delta-structure.

Figure 41-34 illustrates the example1. RGBSPSEL == 1'b0, RGB_SKIP == 1'b1, PIXCOMPEN_DIR == 1'b0.

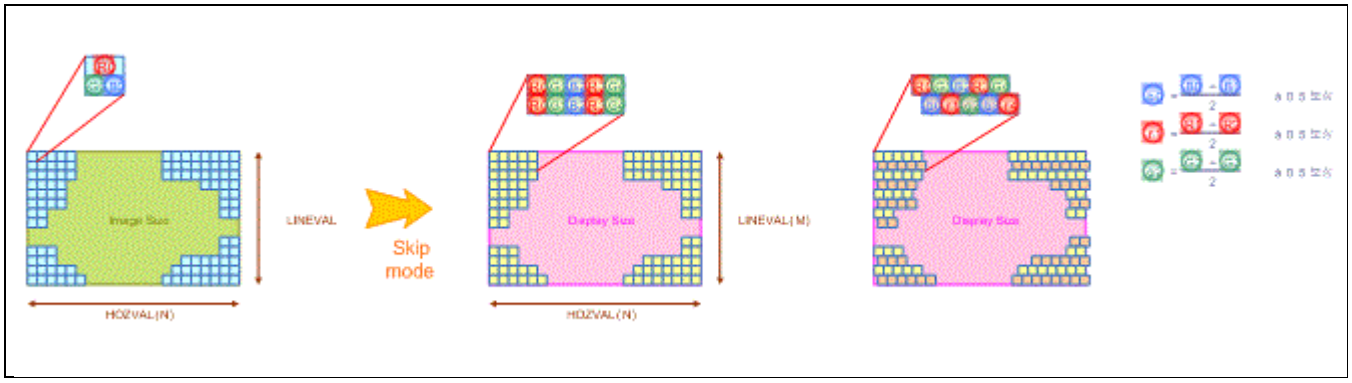


Figure 41-34 Example1. RGBSPSEL == 1'b0, RGB_SKIP == 1'b1, PIXCOMPEN_DIR == 1'b0

Figure 41-35 illustrates the example2. RGBSPSEL == 1'b1, PIXCOMPEN_DIR == 1'b0

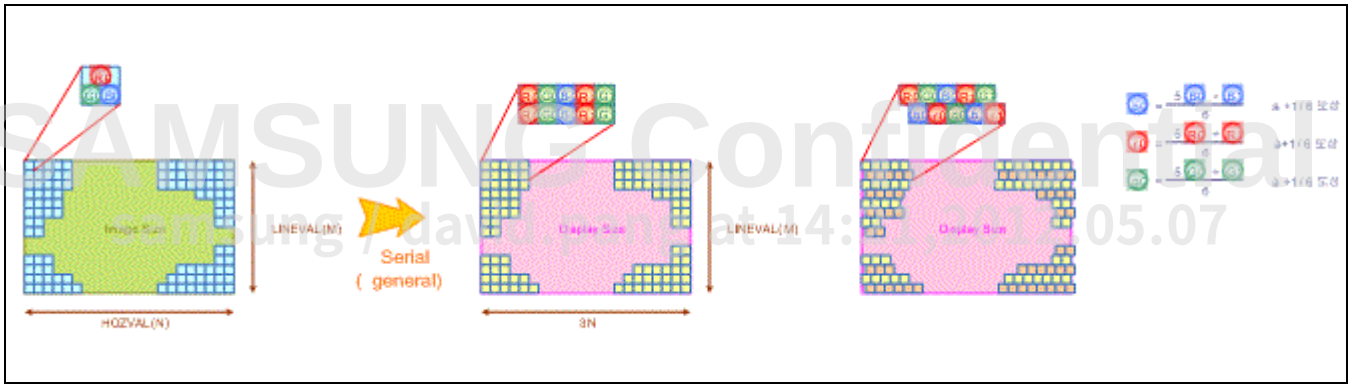


Figure 41-35 Example2. RGBSPSEL == 1'b1, PIXCOMPEN_DIR == 1'b0

Figure 41-36 illustrates the example3. RGBSPSEL == 1'b1, PIXCOMPEN_DIR == 1'b1.

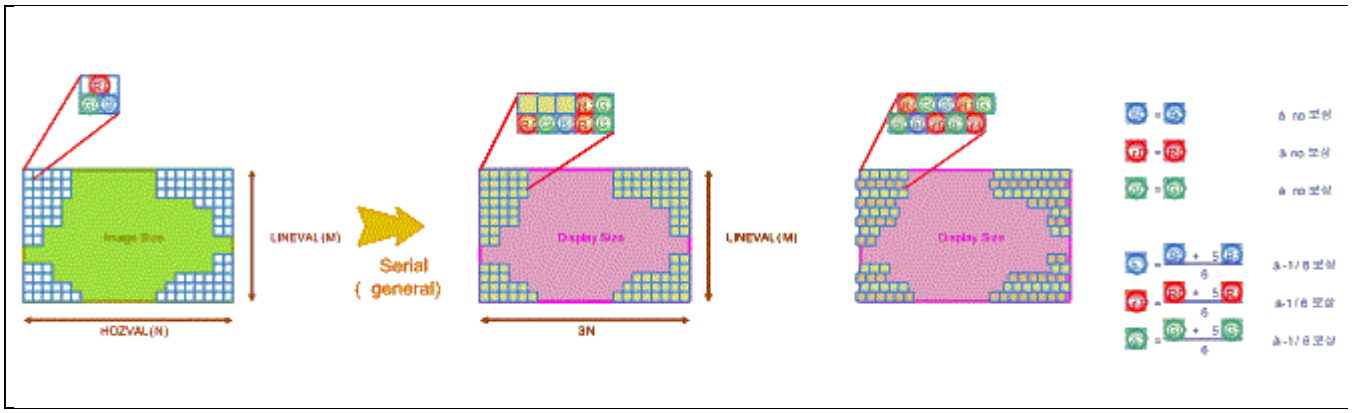


Figure 41-36 Example3. RGBSPSEL == 1'b1, PIXCOMPEN_DIR == 1'b1

41.3.8 VTIME Controller Operation

VTIME comprises of two blocks, namely:

- VTIME_RGB_TV for RGB timing control
- VTIME_i80 for indirect i80 interface timing control

41.3.8.1 RGB Interface Controller

VTIME generates control signals such as RGB_VSYNC, RGB_HSYNC, RGB_VDEN, and RGB_VCLK signal for the RGB interface. You can use these control signals while configuring the VIDTCON0/1/ 2 registers in the VSFR register.

You can program configurations of display control registers in the VSFR. Then, the VTIME module generates programmable control signals that support different types of display devices.

The RGB_VSYNC signal causes the LCD line pointer to begin at the top of display. The configuration of both HOZVAL field and LINEVAL registers control pulse generation of RGB_VSYNC and RGB_HSYNC. Based on these equations, the size of the LCD panel determines HOZVAL and LINEVAL:

- HOZVAL = (Horizontal display size) – 1
- LINEVAL = (Vertical display size) – 1

The CLKVAL field in VIDCON0 register controls the rate of RGB_VCLK signal.
RGB_VCLK (Hz) = SCLK_FIMDx/(CLKVAL + 1), where CLKVAL ≥ 1 where, SCLK_FIMDx (x = 0, 1)

Table 41-1 describes the relationship of RGB_VCLK and CLKVAL. The minimum value of CLKVAL is 1.

- RGB_VCLK (Hz) = SCLK_FIMDx/(CLKVAL + 1), where CLKVAL ≥ 1 where, SCLK_FIMDx (x = 0, 1)

Table 41-1 Relation 16 BPP between VCLK and CLKVAL (TFT, Frequency of Video Clock Source = 60 MHz)

CLKVAL	60 MHz/X	VCLK
2	60 MHz/3	20.0 MHz
3	60 MHz/4	15.0 MHz
:	:	:
63	60 MHz/64	937.5 kHz

VSYNC, VBPD, VFPD, HSYNC, HBPD, HFPD, HOZVAL, and LINEVAL configure RGB_HSYNC and RGB_VSYNC signal. For more information, refer to RGB_VCLK (Hz) = SCLK_FIMDx/(CLKVAL + 1), where CLKVAL ≥ 1 where, SCLK_FIMDx (x = 0, 1)

The frame rate is RGB_VSYNC signal frequency. The frame rate associates with the field of VSYNC, VBPD, VFPD, LINEVAL, HSYNC, HBPD, HFPD, HOZVAL, and CLKVAL registers. Most LCD drivers require their own adequate frame rate.

The equation to calculate frame rate is:

Frame Rate = 1/[{(VSPW + 1) + (VBPD + 1) + (LIINEVAL + 1) + (VFPD + 1)} × {(HSPW + 1) + (HBPD + 1) + (HFPD + 1) + (HOZVAL + 1)} × {(CLKVAL + 1)/(Frequency of Clock source)}]

41.3.8.2 i80 Interface Controller

VTIME_I80 controls display controller for CPU style LCD Driver IC (LDI).

The functions of interface controller are:

- Generates I80 Interface Control Signals
- CPU style LDI Command Control
- Timing Control for VDMA and VDPRCS

41.3.8.3 Output Control Signal Generation

VTIME_I80 generates SYS_CS0, SYS_CS1, SYS_WE, and SYS_RS control signals (for Timing Diagram, refer to RGB_VCLK (Hz) = SCLK_FIMDx/(CLKVAL + 1), where CLKVAL ≥ 1 where, SCLK_FIMDx (x = 0, 1).

SYS_CS0, SYS_CS1, SYS_WE and SYS_RS timing parameters, LCD_CS_SETUP, LCD_WR_SETUP, LCD_WR_ACT, and LCD_WR_HOLD are set through I80IFCONA0 and I80IFCONA1 SFRs.

41.3.8.4 Partial Display Control

Although partial display is the main feature of CPU style LDI, VTIME_I80 does not support this function in hardware logic.

SFR setting (LINEVAL, HOZVAL, OSD_LeftTopX_F, OSD_LeftTopY_F, OSD_RightBotX_F, OSD_RightBotY_F, PAGEWIDTH, and OFFSIZE) implements partial display function.

41.3.8.5 LDI Command Control

LDI receives both command and data. Command specifies an index for selecting the SFR in LDI. In control signal for command and data, only SYS_RS signal has a special function. Usually, SYS_RS has a polarity of '1' for issuing command and vice versa.

Display controller has two kinds of command control:

- Auto command
- Normal command

Auto command is issued automatically, that is, without software control and at a pre-defined rate (rate = 2, 4, 6, ... 30). If the rate is equal to 4, then it implies that auto commands are sent to LDI at the end of every four image frames.

Normal command: The software control issues Normal command.

41.3.9 Setting of Commands

Setting of Commands section describes the settings of Auto and Normal command.

41.3.9.1 Auto Command

If 0x1 (index), 0x32, 0x2 (index), 0x8f, 0x4 (index), or 0x99 requires to be sent to LDI at every 10 frames, then the recommended steps are:

1. LDI_CMD0 ← 0x1, LDI_CMD1 ← 0x32, LDI_CMD2 ← 0x2
2. LDI_CMD3 ← 0x8f, LDI_CMD4 ← 0x4, LDI_CMD5 ← 0x99
3. CMD0_EN ← 0x2, CMD1_EN ← 0x2, CMD2_EN ← 0x2
4. CMD3_EN ← 0x2, CMD4_EN ← 0x2, CMD5_EN ← 0x2
5. CMD0_RS ← 0x1, CMD1_RS ← 0x0, CMD2_RS ← 0x1
6. CMD3_RS ← 0x0, CMD4_RS ← 0x1, CMD5_RS ← 0x0
7. AUTO_CMD_RATE ← 0x5

NOTE:

1. For checking RS polarity, refer to your LDI specification.
2. Do not pack LDI_CMD from LDI_CMD0 to LDI_CMD11 contiguously. For example, it is only possible to use LDI_CMD0, LDI_CMD3, and LDI_CMD11.
3. Maximum 12 auto commands are available.

41.3.9.2 Normal Command

13. The steps to execute Normal command are: Put commands into LDI_CMD0 – 11 (maximum 12 commands).
14. Set CMDx_EN in LDI_CMDCON0 to enable normal command × (For example, if you want to enable command 4, you have to set CMD4_EN to 0x01).
15. Set NORMAL_CMD_ST in I80IFCONB0/1.

- The characteristics that display controller has for the command operations are: Auto, Normal, Auto and Normal command mode is possible for each of the 12 commands.
- Sends 12 maximum commands between frames in its normal operation (Normal operation means ENVID = 1 and displays video data in LCD panel).
- Issues commands in the order of CMD0 → CMD1 → CMD2 → CMD3 → … → CMD10 → CMD11.
- Skips commands that are in disable state. (CMDx_EN = 0x0).
  - Sends over 12 commands (possible in Normal command and system initialization).
  - Sets 12 LDI_CMDx, CMDx_EN, and CMDx_RS.
  - Sets NORMAL_CMD_ST.
  - Reads NORMAL_CMD_ST with polling. If 0, then go to NORMAL_CMD_ST setting.

1.2.7.2.1 Command Setting Example

- CMD0_EN = 2'b10, CMD1_EN = 2'b11, CMD2_EN = 2'b01, CMD3_EN = 2'b11, and CMD4_EN = 2'b01 (Auto Command: CMD0, CMD1, CMD3, Normal Command: CMD1, CMD2, CMD3, and CMD4)
- AUTO_COMMAND_RATE = 4'b0010 (per four frames)
- CMD0_RS = 1, CMD1_RS = 1, CMD2_RS = 0, CMD3_RS = 1, and CMD4_RS = 0
- RSPOL = 0

Figure 41-37 illustrates the sending command.

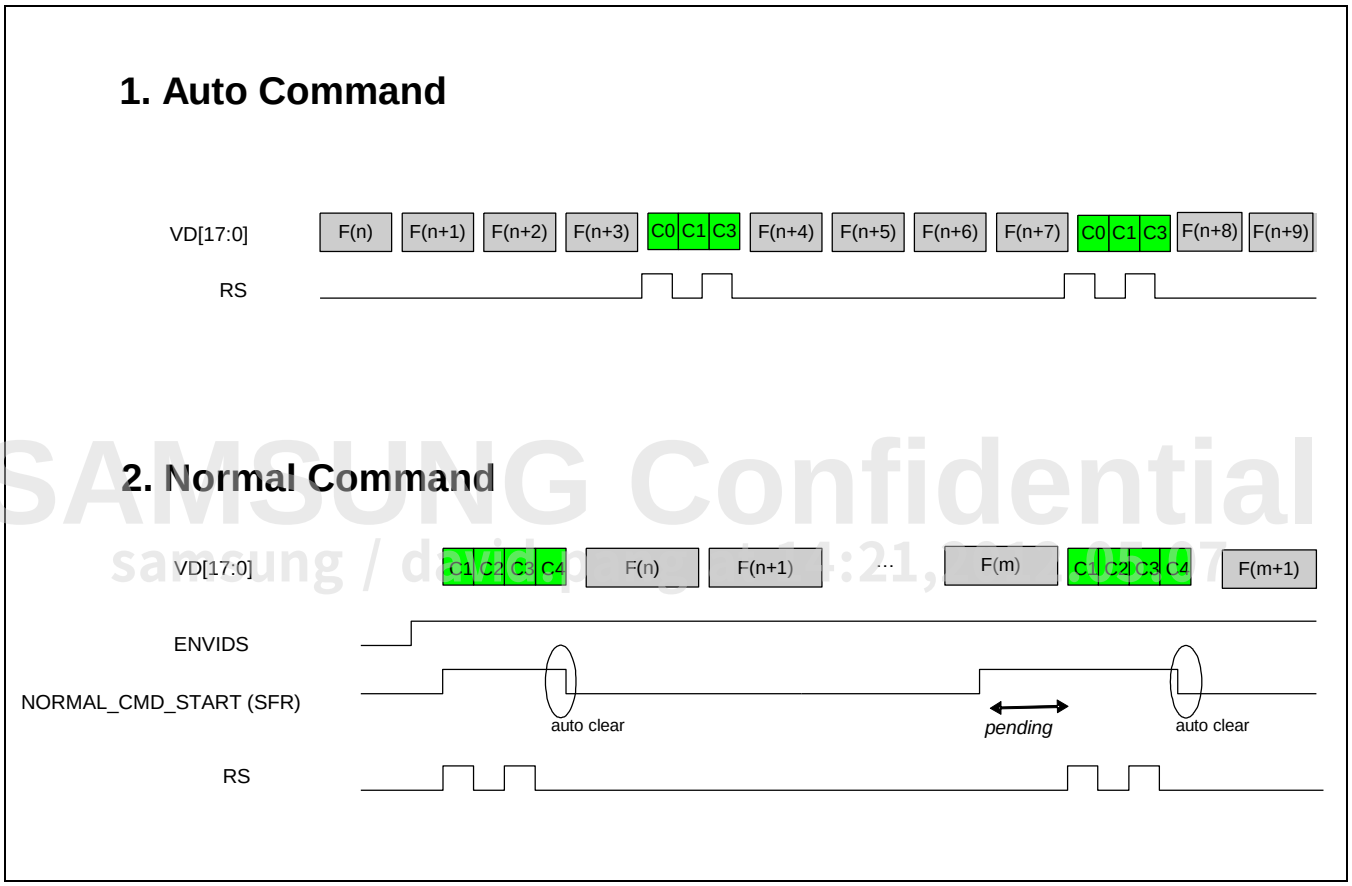


Figure 41-37 Sending Command

1.2.7.2.2 Indirect I80 Interface Trigger

VTIME_I80 starts its operation when a software trigger occurs. There are two kinds of triggers. However, it generates the software trigger by setting TRGCON SFR.

41.3.9.3 Interrupt

Completion of one frame generates Frame Done Interrupt.

41.3.9.3.1 Indirect I80 Interface Output Mode

Table 41-2 describes the output mode of indirect i80 interface based on mode at VIDCON0.

Table 41-2 i80 Output Mode

VIDCON0 Register	Value	BPP	Bus Width	Split	DATA	Command
DSI_EN	1	24	24	X	{R[7:0], G[7:0], B[7:0]}	CMD[23:0]
L0/1_DATA	000	16	16	X	{R[7:3], G[7:2], B[7:3]}	CMD[15:0]
	001	18	16	O (1st) (2nd)	{R[7:2], G[7:2], B[7:4]} {14'b0, B[3:2]}	CMD[15:0] –
	010	18	9	O (1st) (2nd)	{R[7:2], G[7:5]} {G[4:2], B[7:2]}	CMD[17:9] CMD[8:0]
	011	24	16	O (1st) (2nd)	{R[7:0], G[7:0]} {B[7:0], 8'b0}	– –
	100	18	18	X	{R[7:2], G[7:2], B[7:2]}	CMD[17:0]
	101	16	8	O (1st) (2nd)	{R[7:3], G[7:5]} {G[4:2], B[7:3]}	CMD[15:8] CMD[7:0]

41.3.10 Virtual Display

Display controller supports hardware horizontal or vertical scrolling. If the screen scrolls, then it changes the fields of LCDBASEU and LCDBASEL (Refer to *Figure 41-38*), but not the values of PAGEWIDTH and OFFSIZE. The size of video buffer in which you store the image should be larger than the LCD panel screen size.

Figure 41-38 illustrates the example of scrolling in virtual display.

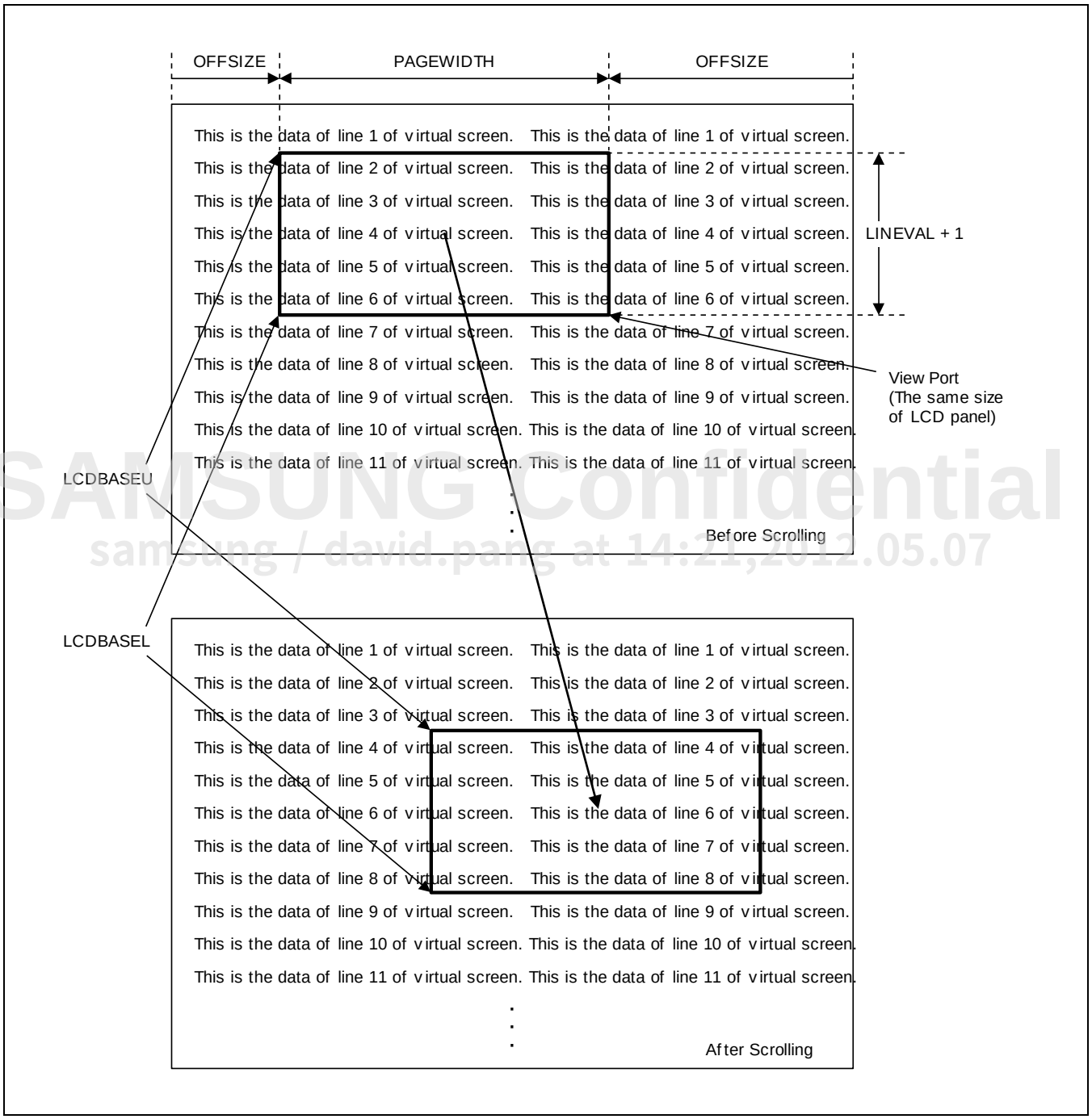


Figure 41-38 Example of Scrolling in Virtual Display

41.3.11 RGB Interface Specification

RGB Interface Spec includes:

- Signals
- LCD RGB Interface Timing
- Parallel Output
- Serial 8-bit Output
- Output Configuration Structure

41.3.11.1 Signals

Table 41-3 describes the signals.

Table 41-3 RGB Interface Signals of Display Controller

Signal	In/Out	Description	Display Controller	
			PADs	GPIO Control
LCD_HSYNC	O	Horizontal Synchronization. Signal	XvHSYNC	GPF0CON[0]
LCD_VSYNC	O	Vertical Synchronization. Signal	XvVSYNC	GPF0CON[1]
LCD_VCLK	O	LCD Video Clock	XvVCLK	GPF0CON[3]
LCD_VDEN	O	Data Enable	XvVDEN	GPF0CON[2]
LCD_VD[0]	O	RGB data output	XvVD_0	GPF0CON[4]
LCD_VD[1]	O	RGB data output	XvVD_1	GPF0CON[5]
LCD_VD[2]	O	RGB data output	XvVD_2	GPF0CON[6]
LCD_VD[3]	O	RGB data output	XvVD_3	GPF0CON[7]
LCD_VD[4]	O	RGB data output	XvVD_4	GPF1CON[0]
LCD_VD[5]	O	RGB data output	XvVD_5	GPF1CON[1]
LCD_VD[6]	O	RGB data output	XvVD_6	GPF1CON[2]
LCD_VD[7]	O	RGB data output	XvVD_7	GPF1CON[3]
LCD_VD[8]	O	RGB data output	XvVD_8	GPF1CON[4]
LCD_VD[9]	O	RGB data output	XvVD_9	GPF1CON[5]
LCD_VD[10]	O	RGB data output	XvVD_10	GPF1CON[6]
LCD_VD[11]	O	RGB data output	XvVD_11	GPF1CON[7]
LCD_VD[12]	O	RGB data output	XvVD_12	GPF2CON[0]
LCD_VD[13]	O	RGB data output	XvVD_13	GPF2CON[1]
LCD_VD[14]	O	RGB data output	XvVD_14	GPF2CON[2]
LCD_VD[15]	O	RGB data output	XvVD_15	GPF2CON[3]
LCD_VD[16]	O	RGB data output	XvVD_16	GPF2CON[4]
LCD_VD[17]	O	RGB data output	XvVD_17	GPF2CON[5]
LCD_VD[18]	O	RGB data output	XvVD_18	GPF2CON[6]

Signal	In/Out	Description	Display Controller	
			PADs	GPIO Control
LCD_VD[19]	O	RGB data output	XvVD_19	GPF2CON[7]
LCD_VD[20]	O	RGB data output	XvVD_20	GPF3CON[0]
LCD_VD[21]	O	RGB data output	XvVD_21	GPF3CON[1]
LCD_VD[22]	O	RGB data output	XvVD_22	GPF3CON[2]
LCD_VD[23]	O	RGB data output	XvVD_23	GPF3CON[3]

While using RGB interface, the VT_LBLKx bit fields in LCDBLKC_CFG (0x1001_0210) register should be set to RGB Interface out (2'b00), even though you use DSI Video Mode.

41.3.11.2 LCD RGB Interface Timing

Figure 41-39 illustrates the LCD RGB interface timing.

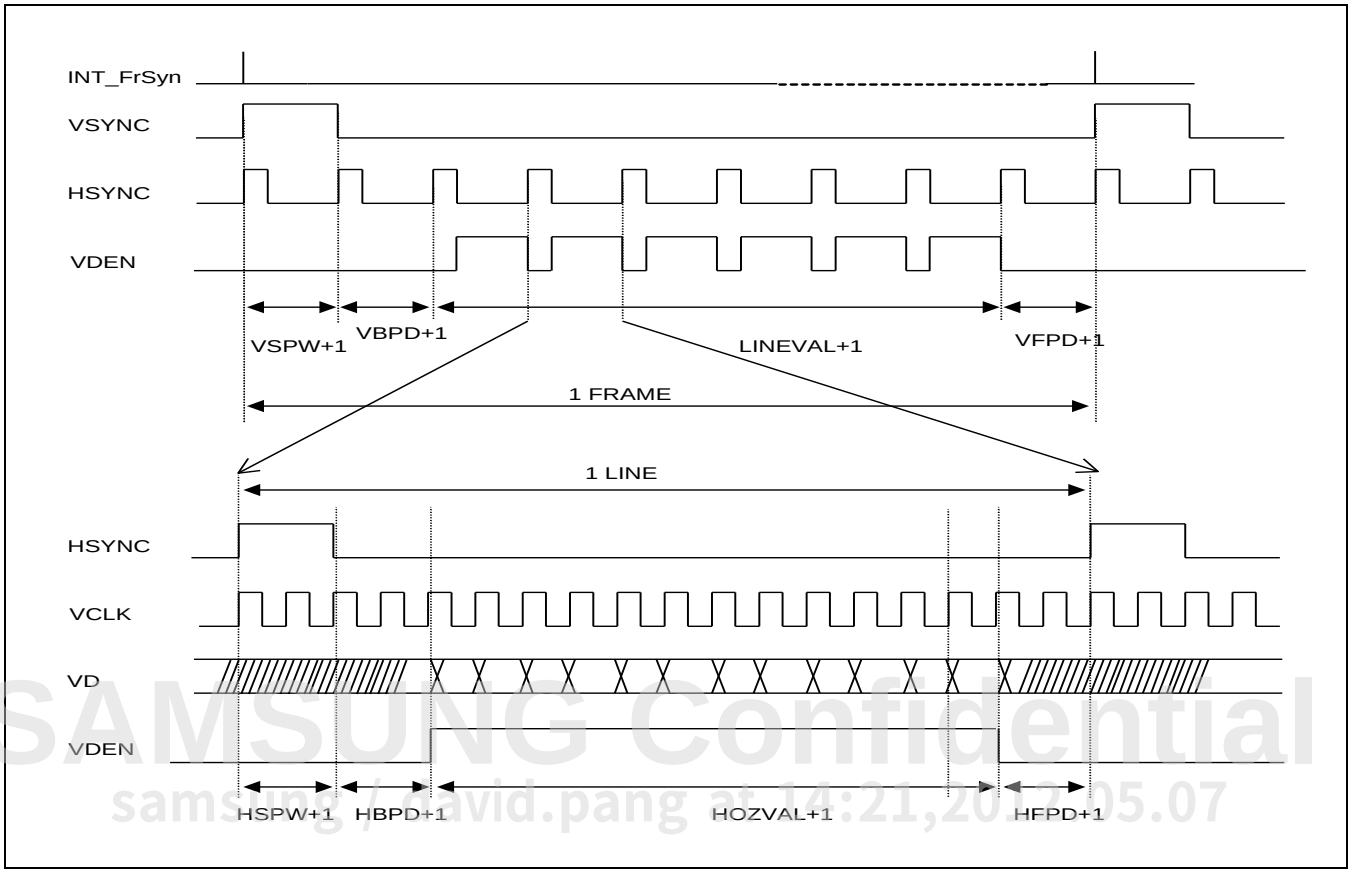


Figure 41-39 LCD RGB Interface Timing

41.3.11.3 Parallel Output

Parallel Output section includes:

- General 24-bit Output
- RGB Skip 8-bit Output

41.3.11.3.1 General 24-bit Output (RGBSPSEL = 0, RGB_SKIP_EN = 0)

Figure 41-40 illustrates the LCD RGB interface timing (RGB parallel).

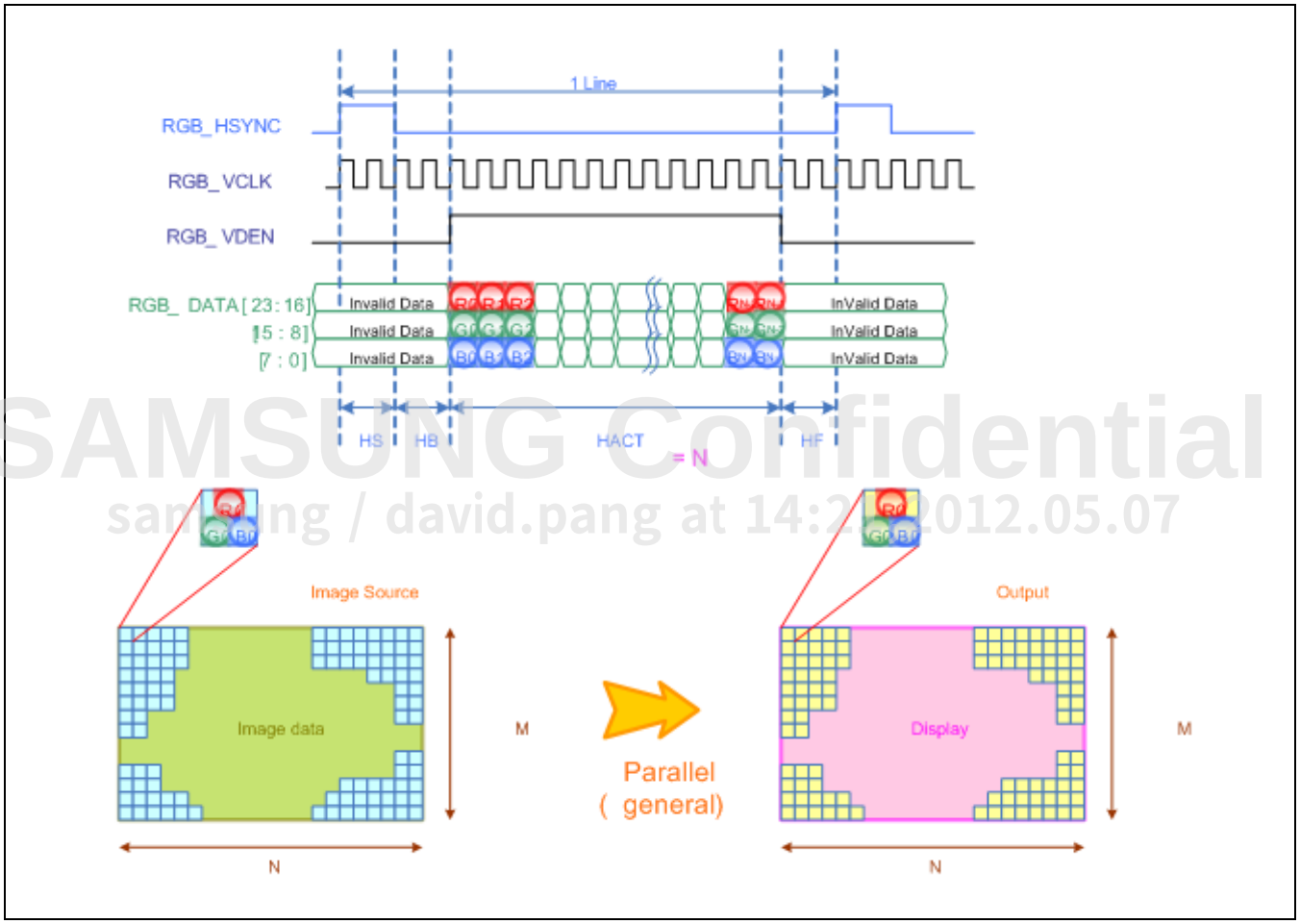


Figure 41-40 LCD RGB Interface Timing (RGB Parallel)

41.3.11.3.2 RGB Skip 8-bit Output (Color Sub Sampling), (RGBSPSEL = 0, RGB_SKIP_EN = 1)

Figure 41-41 illustrates the LCD RGB interface timing (RGB Skip)

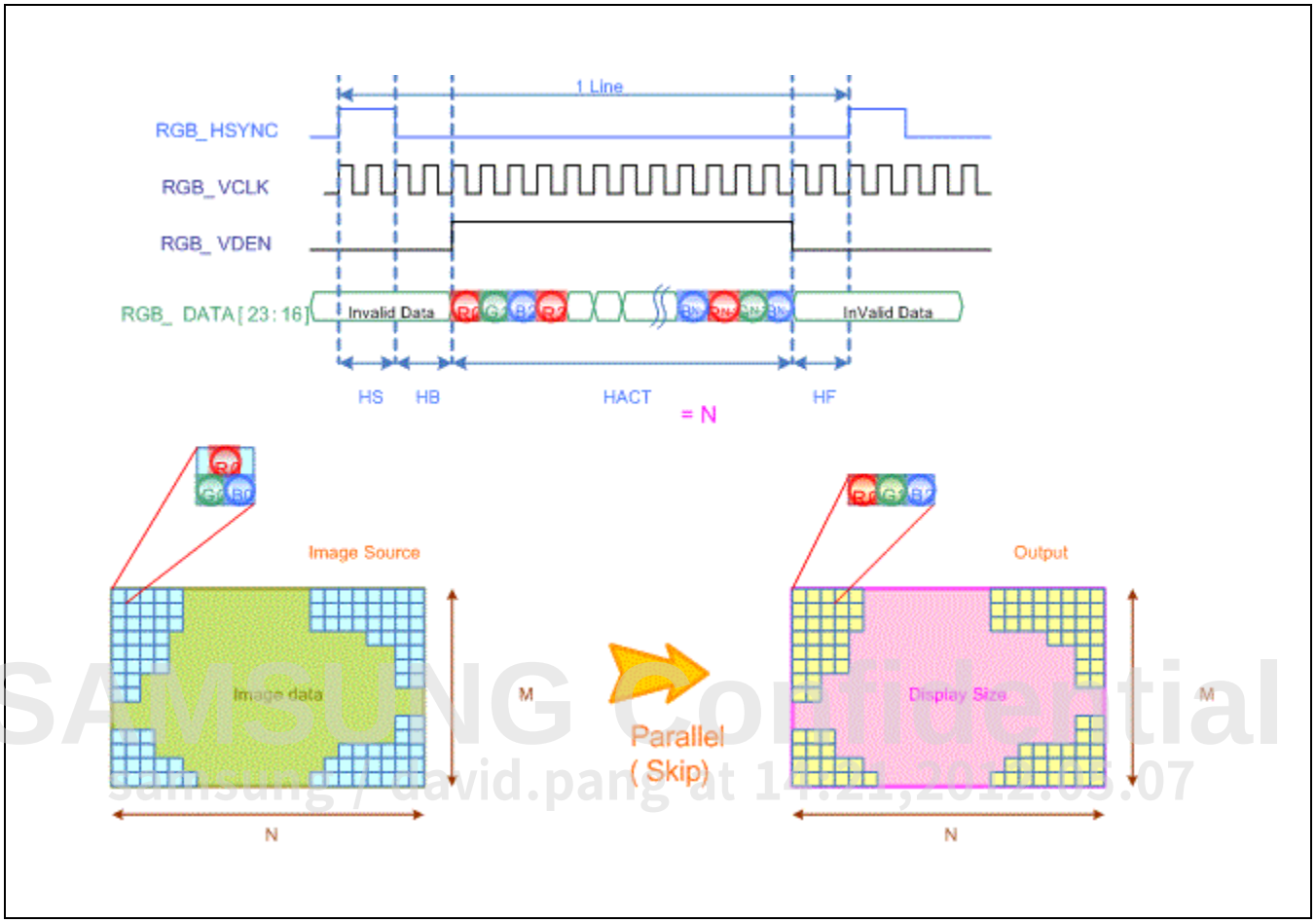


Figure 41-41 LCD RGB Interface Timing (RGB Skip)

41.3.11.4 Serial 8-bit Output

Serial 8-bit Output section includes:

- General 8-bit Output
- Dummy Insertion Output

41.3.11.4.1 General 8-bit Output (RGBSPSEL = 1, RGB_DUMMY_EN = 0)

Figure 41-42 illustrates the LCD RGB interface timing (RGB serial, dummy disable).

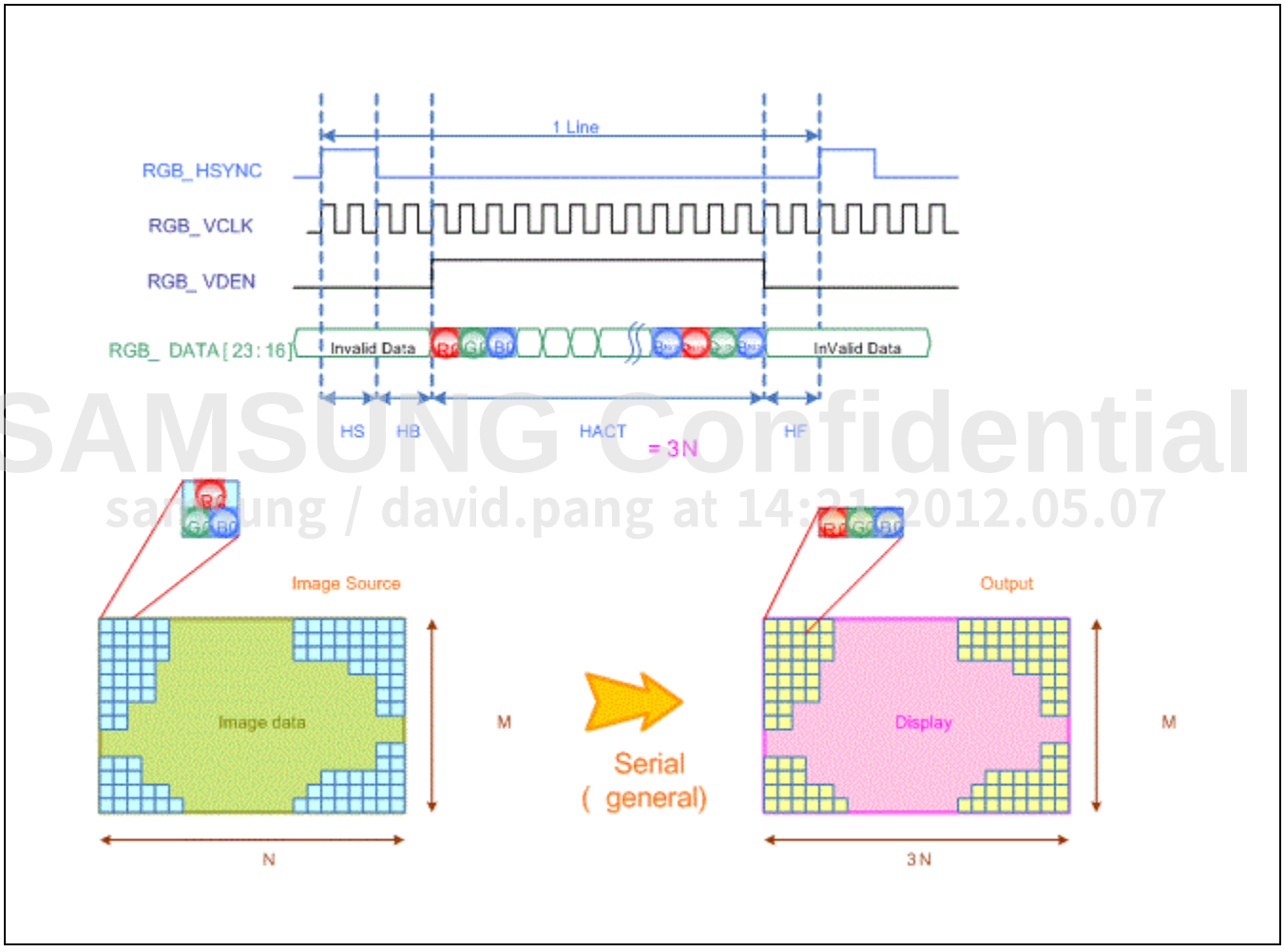


Figure 41-42 LCD RGB Interface Timing (RGB Serial, Dummy Disable)

41.3.11.4.2 Dummy Insertion Output (RGBSPSEL = 1, RGB_DUMMY_EN = 1)

Figure 41-43 illustrates the LCD RGB interface timing (RGB serial, dummy insertion).

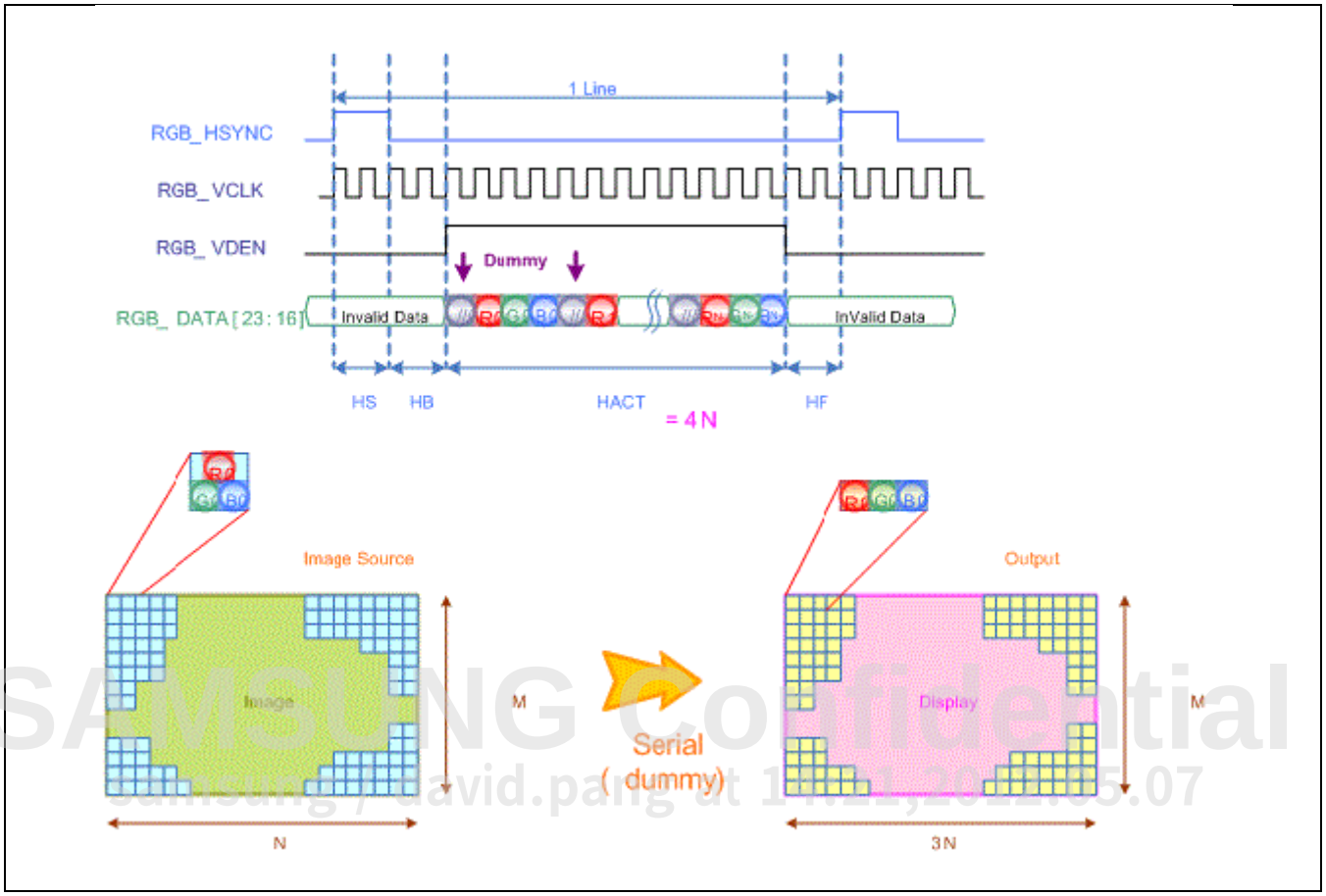


Figure 41-43 LCD RGB Interface Timing (RGB serial, Dummy Insertion)

41.3.11.5 Output Configuration Structure

Output Configuration Structure includes:

- Color Order Control
- Example of Delta Structure

41.3.11.5.1 Color Order Control

"RGB_ORDER_O" controls odd line color structure. On the other hand, "RGB_ORDER_E" at VIDCON2 controls even line color structure.

Figure 41-44 illustrates the LCD RGB output order.

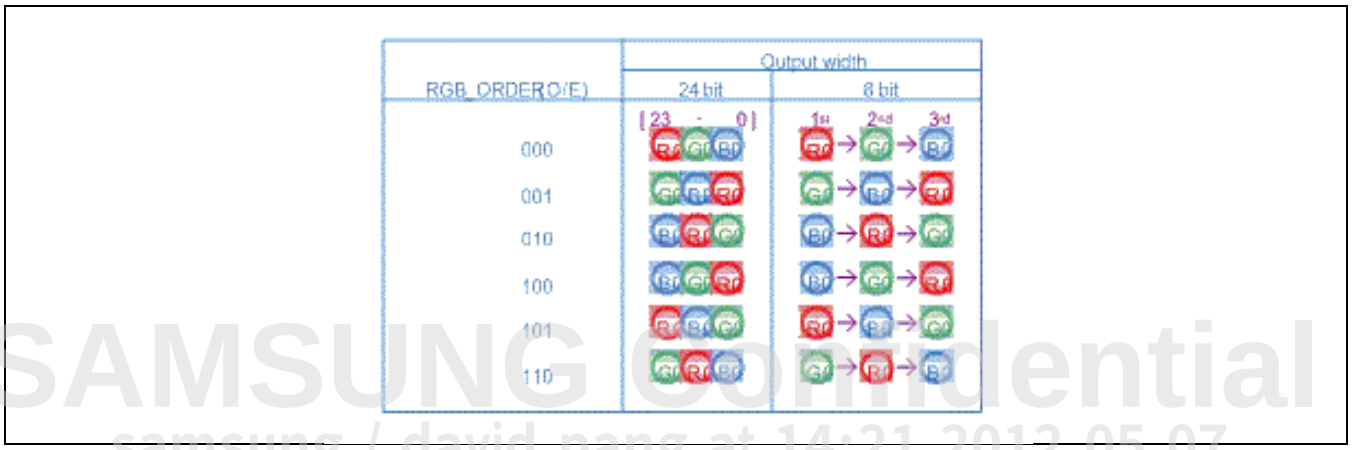


Figure 41-44 LCD RGB Output Order

41.3.11.5.2 Example of Delta Structure

For more information, refer to register "RGB_ORDER_O, E".

Example 41-6 Delta Structure and RGB Interface Output Order

```
RGB_ORDER_O = 000, RGB_ORDER_E = 001
```

Figure 41-45 illustrates the delta structure and LCD RGB interface timing.

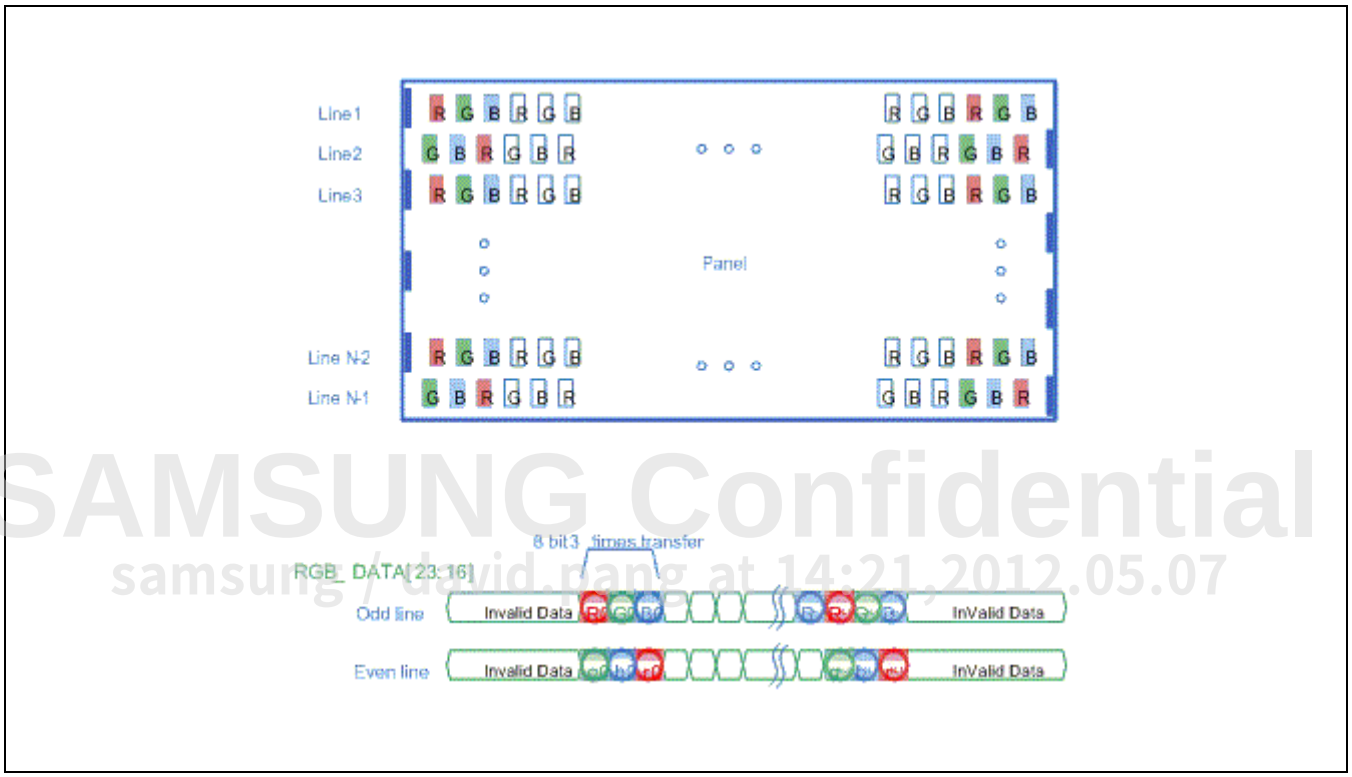


Figure 41-45 Delta Structure and LCD RGB Interface Timing

41.3.12 LCD Indirect i80 System Interface

LCD Indirect i80 System Interface includes:

- Signals
- Indirect i80 System Interface Write Cycle Timing

41.3.12.1 Signals

Table 41-4 describes the signals.

Table 41-4 LCD Indirect i80 System Interface Signals of Display Controller

Signal	In/Out	Description	Display Controller	
			PAD	GPIO Control
SYS_VD[0]	I/O	Data bit[0]	XvVD_0	GPF0CON[4]
SYS_VD[1]	I/O	Data bit[1]	XvVD_1	GPF0CON[5]
SYS_VD[2]	I/O	Data bit[2]	XvVD_2	GPF0CON[6]
SYS_VD[3]	I/O	Data bit[3]	XvVD_3	GPF1CON[7]
SYS_VD[4]	I/O	Data bit[4]	XvVD_4	GPF1CON[0]
SYS_VD[5]	I/O	Data bit[5]	XvVD_5	GPF1CON[1]
SYS_VD[6]	I/O	Data bit[6]	XvVD_6	GPF1CON[2]
SYS_VD[7]	I/O	Data bit[7]	XvVD_7	GPF1CON[3]
SYS_VD[8]	I/O	Data bit[8]	XvVD_8	GPF1CON[4]
SYS_VD[9]	I/O	Data bit[9]	XvVD_9	GPF1CON[5]
SYS_VD[10]	I/O	Data bit[10]	XvVD_10	GPF1CON[6]
SYS_VD[11]	I/O	Data bit[11]	XvVD_11	GPF1CON[7]
SYS_VD[12]	I/O	Data bit[12]	XvVD_12	GPF2CON[0]
SYS_VD[13]	I/O	Data bit[13]	XvVD_13	GPF2CON[1]
SYS_VD[14]	I/O	Data bit[14]	XvVD_14	GPF2CON[2]
SYS_VD[15]	I/O	Data bit[15]	XvVD_15	GPF2CON[3]
SYS_VD[16]	I/O	Data bit[16]	XvVD_16	GPF2CON[4]
SYS_VD[17]	I/O	Data bit[17]	XvVD_17	GPF2CON[5]
SYS_CS0	O	Chip select for LCD0	XvHSYNC	GPF0CON[0]
SYS_CS1	O	Chip select for LCD1	XvVSYNC	GPF0CON[1]
SYS_WE	O	Write enable	XvVCLK	GPF0CON[3]
SYS_OE	O	Output enable	XvSYS_OE	GPF3CON[5]
SYS_RS/SYS_ADD[0]	O	Address Output[0]	XvVDEN	GPF0CON[2]
SYS_ST/SYS_ADD[1]	O	Address Output[1]	–	Internal Connection

NOTE:

1. SYS_ST/SYS_ADD[1] is valid in DSI Mode (VIDCON0 [30] = 1)
   SYS_ADD[1] = SYS_ST: 0 when VDOUT is from Frame
   SYS_ADD[1] = SYS_ST: 1 when VDOUT is from Command
2. While using RGB interface, set the VT_LBLKx bit fields in LCDBLKC_CFG (0x1001_0210) register to i80 interface out (2'b01), even though you use DSI Command Mode.

41.3.12.2 Indirect i80 System Interface Write Cycle Timing

Figure 41-46 illustrates the indirect i80 system interface write cycle timing.

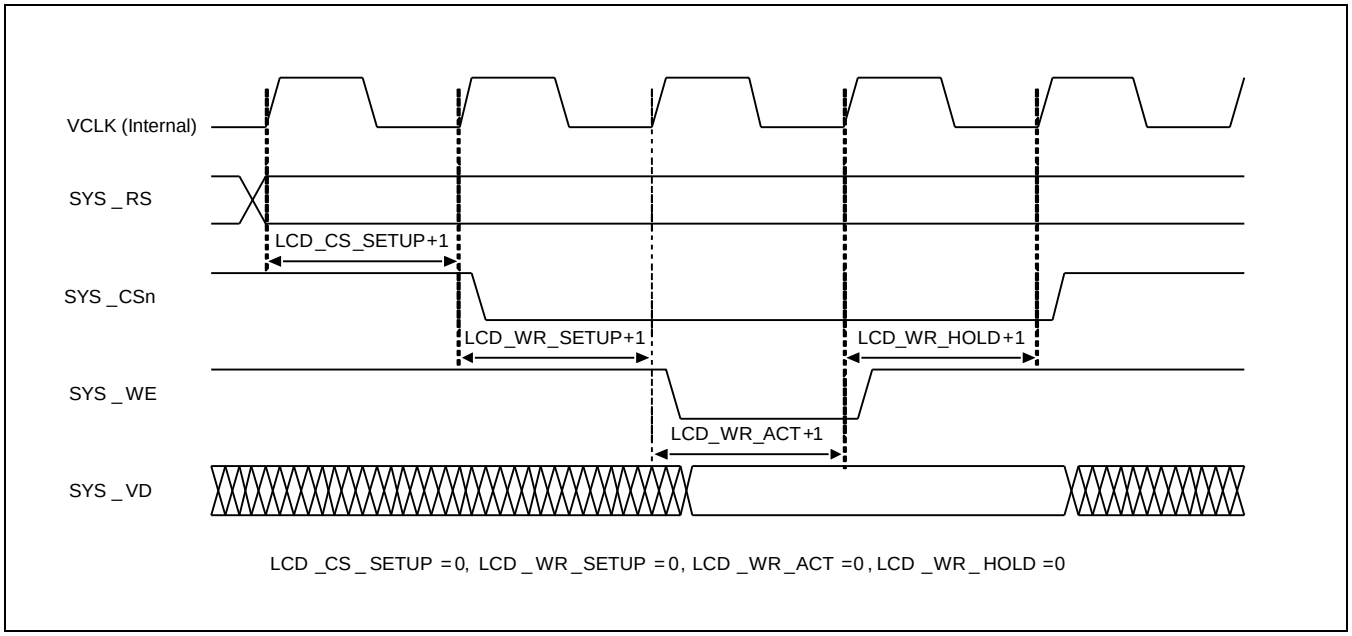


Figure 41-46 Indirect i80 System Interface Write Cycle Timing

Table 41-5 describes the timing reference code (XY Definition).

Table 41-5 Timing Reference Code (XY Definition)

–	Parallel RGB			Serial RGB	
	24 BPP (888)	18 BPP (666)	16 BPP (565)	24 BPP (888)	18 BPP (666)
VD[23]	R[7]	R[5]	R[4]	D[7]	D[5]
VD[22]	R[6]	R[4]	R[3]	D[6]	D[4]
VD[21]	R[5]	R[3]	R[2]	D[5]	D[3]
VD[20]	R[4]	R[2]	R[1]	D[4]	D[2]
VD[19]	R[3]	R[1]	R[0]	D[3]	D[1]
VD[18]	R[2]	R[0]	–	D[2]	D[0]
VD[17]	R[1]	–	–	D[1]	–
VD[16]	R[0]	–	–	D[0]	–
VD[15]	G[7]	G[5]	G[5]	–	–
VD[14]	G[6]	G[4]	G[4]	–	–
VD[13]	G[5]	G[3]	G[3]	–	–
VD[12]	G[4]	G[2]	G[2]	–	–
VD[11]	G[3]	G[1]	G[1]	–	–
VD[10]	G[2]	G[0]	G[0]	–	–
VD[9]	G[1]	–	–	–	–
VD[8]	G[0]	–	–	–	–
VD[7]	B[7]	B[5]	B[4]	–	–
VD[6]	B[6]	B[4]	B[3]	–	–
VD[5]	B[5]	B[3]	B[2]	–	–
VD[4]	B[4]	B[2]	B[1]	–	–
VD[3]	B[3]	B[1]	B[0]	–	–
VD[2]	B[2]	B[0]	–	–	–
VD[1]	B[1]	–	–	–	–
VD[0]	B[0]	–	–	–	–

41.4 I/O Description

Table 41-6 describes the I/O.

Table 41-6 I/O Signals of Display Controller

Signal	In/Out	Description	PAD for Display Controller	Type (1)
LCD_HSYNC	Out	Horizontal Synchronization. Signal	XvHSYNC	Muxed
LCD_VSYNC	Out	Vertical Synchronization. Signal	XvVSYNC	Muxed
LCD_VCLK	Out	LCD Video Clock	XvVCLK	Muxed
LCD_VDEN	Out	Data Enable	XvVDEN	Muxed
LCD_VD[23:0]	Out	RGB Data Output	XvVD_23 to XvVD_0	Muxed
SYS_VD[17:0]	In/Out	Data to/from Display Controller from/to Display Module	XvVD_17 to XvVD_0	Muxed
SYS_CS0	Out	Chip select for LCD0	XvHSYNC	Muxed
SYS_CS1	Out	Chip select for LCD1	XvVSYNC	Muxed
SYS_WE	Out	Write Enable	XvVCLK	Muxed
SYS_OE	Out	Output Enable	XvSYS_OE	Muxed
SYS_RS/ SYS_ADD[0]	Out	Address Output SYS_ADD[0] is Register/State select	XvVDEN	Muxed
VSYNC_LDI	Out	VSYNC signal for Vsync Interface (2)	XvVSYNC_LDI	Muxed
LCD_FRM	Out	Frame Synchronization Signal for general use (3)	XpwmTout_0	Muxed

NOTE:

1. Type field indicates type (or kind) of the pad whether it is dedicated to a signal or connected to multiplexed signals.
2. VSYNCEN register controls VSYNC_LD signal.
3. FRMEN control register controls LCD_FRM signal.

41.5 Register Description

Overview

The registers you can use to configure display controller are:

1. VIDCON0: Configures video output format and displays enable/disable.
2. VIDCON1: Specifies RGB I/F control signal.
3. VIDCON2: Specifies output data format control.
4. VIDCON3: Specifies image enhancement control.
5. I80IFCONx: Specifies CPU interface control signal.
6. VIDTCONx: Configures video output timing and determines the size of display.
7. WINCONx: Specifies each window feature setting.
8. VIDOSDxA, VIDOSDxB: Specifies window position setting.
9. VIDOSDxC, D: Specifies On Screen Display (OSD) size setting.
10. VIDWxALPHA0/1: Specifies alpha value setting.
11. BLENDEQx: Specifies blending equation setting.
12. VIDWxxADDx: Specifies source image address setting.
13. WxKEYCONx: Specifies color key setting register.
14. WxKEYALPHA: Specifies color key alpha value setting.
15. WINxMAP: Specifies window color control.
16. GAMMALUT_xx: Specifies gamma value setting.
17. COLORGAINCON: Specifies color gain value setting.
18. HUExxx: Specifies Hue coefficient and offset value setting.
19. WPALCON: Specifies palette control register.
20. WxRTQOSCON: Specifies RTQoS control register.
21. WxPDATAxx: Specifies window palette data of each index.
22. SHDOWCON: Specifies shadow control register.
23. WxRTQOSCON: Specifies QoS control register.

41.5.1 Register Map Summary

- Base Address = 0x11C0_0000.

Register	Offset	Description	Reset Value
Control Register			
VIDCON0	0x0000	Specifies video control 0 register.	0x0000_0000
VIDCON1	0x0004	Specifies video control 1 register.	0x0000_0000
VIDCON2	0x0008	Specifies video control 2 register.	0x0000_0000
VIDCON3	0x000C	Specifies video control 3 register.	0x0000_0000
VIDTCON0	0x0010	Specifies video time control 0 register.	0x0000_0000
VIDTCON1	0x0014	Specifies video time control 1 register.	0x0000_0000
VIDTCON2	0x0018	Specifies video time control 2 register.	0x0000_0000
VIDTCON3	0x001C	Specifies video time control 3 register.	0x0000_0000
WINCON0	0x0020	Specifies window control 0 register.	0x0000_0000
WINCON1	0x0024	Specifies window control 1 register.	0x0000_0000
WINCON2	0x0028	Specifies window control 2 register.	0x0000_0000
WINCON3	0x002C	Specifies window control 3 register.	0x0000_0000
WINCON4	0x0030	Specifies window control 4 register.	0x0000_0000
SHADOWCON	0x0034	Specifies window shadow control register.	0x0000_0000
WINCHMAP2	0x003C	Specifies window and channel mapping control register.	0x7D51_7D51
VIDOSD0A	0x0040	Specifies video window 0's position control register.	0x0000_0000
VIDOSD0B	0x0044	Specifies video window 0's position control register.	0x0000_0000
VIDOSD0C	0x0048	Specifies video window 0's size control register.	0x0000_0000
VIDOSD1A	0x0050	Specifies video window 1's position control register.	0x0000_0000
VIDOSD1B	0x0054	Specifies video window 1's position control register	0x0000_0000
VIDOSD1C	0x0058	Specifies video window 1's alpha control register.	0x0000_0000
VIDOSD1D	0x005C	Specifies video window 1's size control register.	0x0000_0000
VIDOSD2A	0x0060	Specifies video window 2's position control register.	0x0000_0000
VIDOSD2B	0x0064	Specifies video window 2's position control register.	0x0000_0000
VIDOSD2C	0x0068	Specifies video window 2's alpha control register.	0x0000_0000
VIDOSD2D	0x006C	Specifies video window 2's size control register.	0x0000_0000
VIDOSD3A	0x0070	Specifies video window 3's position control register.	0x0000_0000
VIDOSD3B	0x0074	Specifies video window 3's position control register.	0x0000_0000
VIDOSD3C	0x0078	Specifies video window 3's alpha control register.	0x0000_0000
VIDOSD4A	0x0080	Specifies video window 4's position control register.	0x0000_0000
VIDOSD4B	0x0084	Specifies video window 4's position control register.	0x0000_0000
VIDOSD4C	0x0088	Specifies video window 4's alpha control register.	0x0000_0000
VIDW00ADD0B0	0x00A0	Specifies window 0's buffer start address register, buffer 0.	0x0000_0000

Register	Offset	Description	Reset Value
VIDW00ADD0B1	0x00A4	Specifies window 0's buffer start address register, buffer 1.	0x0000_0000
VIDW00ADD0B2	0x20A0	Specifies window 0's buffer start address register, buffer 2.	0x0000_0000
VIDW01ADD0B0	0x00A8	Specifies window 1's buffer start address register, buffer 0.	0x0000_0000
VIDW01ADD0B1	0x00AC	Specifies window 1's buffer start address register, buffer 1.	0x0000_0000
VIDW01ADD0B2	0x20A8	Specifies window 1's buffer start address register, buffer 2.	0x0000_0000
VIDW02ADD0B0	0x00B0	Specifies window 2's buffer start address register, buffer 0.	0x0000_0000
VIDW02ADD0B1	0x00B4	Specifies window 2's buffer start address register, buffer 1.	0x0000_0000
VIDW02ADD0B2	0x20B0	Specifies window 2's buffer start address register, buffer 2.	0x0000_0000
VIDW03ADD0B0	0x00B8	Specifies window 3's buffer start address register, buffer 0.	0x0000_0000
VIDW03ADD0B1	0x00BC	Specifies window 3's buffer start address register, buffer 1.	0x0000_0000
VIDW03ADD0B2	0x20B8	Specifies window 3's buffer start address register, buffer 2.	0x0000_0000
VIDW04ADD0B0	0x00C0	Specifies window 4's buffer start address register, buffer 0.	0x0000_0000
VIDW04ADD0B1	0x00C4	Specifies window 4's buffer start address register, buffer 1.	0x0000_0000
VIDW04ADD0B2	0x20C0	Specifies window 4's buffer start address register, buffer 2.	0x0000_0000
VIDW00ADD1B0	0x00D0	Specifies window 0's buffer end address register, buffer 0.	0x0000_0000
VIDW00ADD1B1	0x00D4	Specifies window 0's buffer end address register, buffer 1.	0x0000_0000
VIDW00ADD1B2	0x20D0	Specifies window 0's buffer end address register, buffer 2.	0x0000_0000
VIDW01ADD1B0	0x00D8	Specifies window 1's buffer end address register, buffer 0.	0x0000_0000
VIDW01ADD1B1	0x00DC	Specifies window 1's buffer end address register, buffer 1.	0x0000_0000
VIDW01ADD1B2	0x20D8	Specifies window 1's buffer end address register, buffer 2.	0x0000_0000
VIDW02ADD1B0	0x00E0	Specifies window 2's buffer end address register, buffer 0.	0x0000_0000
VIDW02ADD1B1	0x00E4	Specifies window 2's buffer end address register, buffer 1.	0x0000_0000
VIDW02ADD1B2	0x20E0	Specifies window 2's buffer end address register, buffer 2.	0x0000_0000
VIDW03ADD1B0	0x00E8	Specifies window 3's buffer end address register, buffer 0.	0x0000_0000
VIDW03ADD1B1	0x00EC	Specifies window 3's buffer end address register, buffer 1.	0x0000_0000
VIDW03ADD1B2	0x20E8	Specifies window 3's buffer end address register, buffer 2.	0x0000_0000
VIDW04ADD1B0	0x00F0	Specifies window 4's buffer end address register, buffer 0.	0x0000_0000
VIDW04ADD1B1	0x00F4	Specifies window 4's buffer end address register, buffer 1.	0x0000_0000
VIDW04ADD1B2	0x20F0	Specifies window 4's buffer end address register, buffer 2.	0x0000_0000
VIDW00ADD2	0x0100	Specifies window 0's buffer size register.	0x0000_0000
VIDW01ADD2	0x0104	Specifies window 1's buffer size register.	0x0000_0000
VIDW02ADD2	0x0108	Specifies window 2's buffer size register.	0x0000_0000
VIDW03ADD2	0x010C	Specifies window 3's buffer size register.	0x0000_0000
VIDW04ADD2	0x0110	Specifies window 4's buffer size register.	0x0000_0000
VIDINTCON0	0x0130	Specifies video interrupt control register.	0x0000_0000
VIDINTCON1	0x0134	Specifies video interrupt pending register.	0x0000_0000
W1KEYCON0	0x0140	Specifies color key control register.	0x0000_0000

Register	Offset	Description	Reset Value
W1KEYCON1	0x0144	Specifies color key value (transparent value) register.	0x0000_0000
W2KEYCON0	0x0148	Specifies color key control register.	0x0000_0000
W2KEYCON1	0x014C	Specifies color key value (transparent value) register.	0x0000_0000
W3KEYCON0	0x0150	Specifies color key control register.	0x0000_0000
W3KEYCON1	0x0154	Specifies color key value (transparent value) register.	0x0000_0000
W4KEYCON0	0x0158	Specifies color key control register.	0x0000_0000
W4KEYCON1	0x015C	Specifies color key value (transparent value) register.	0x0000_0000
W1KEYALPHA	0x0160	Specifies color key alpha value register.	0x0000_0000
W2KEYALPHA	0x0164	Specifies color key alpha value register.	0x0000_0000
W3KEYALPHA	0x0168	Specifies color key alpha value register.	0x0000_0000
W4KEYALPHA	0x016C	Specifies color key alpha value register.	0x0000_0000
DITHMODE	0x0170	Specifies dithering mode register.	0x0000_0000
WIN0MAP	0x0180	Specifies window 0's color control.	0x0000_0000
WIN1MAP	0x0184	Specifies window 1's color control.	0x0000_0000
WIN2MAP	0x0188	Specifies window 2's color control.	0x0000_0000
WIN3MAP	0x018C	Specifies window 3's color control.	0x0000_0000
WIN4MAP	0x0190	Specifies window 4's color control.	0x0000_0000
WPALCON_H	0x019C	Specifies window palette control register.	0x0000_0000
WPALCON_L	0x01A0	Specifies window palette control register.	0x0000_0000
TRIGCON	0x01A4	Specifies i80/ RGB trigger control register.	0x0000_0000
I80IFCONA0	0x01B0	Specifies i80 interface control 0 for main LDI.	0x0000_0000
I80IFCONA1	0x01B4	Specifies i80 interface control 0 for sub LDI.	0x0000_0000
I80IFCONB0	0x01B8	Specifies i80 interface control 1 for main LDI.	0x0000_0000
I80IFCONB1	0x01BC	Specifies i80 interface control 1 for sub LDI.	0x0000_0000
COLORGAINCON	0x01C0	Specifies color gain control register.	0x1004_0100
LDI_CMDCON0	0x01D0	Specifies i80 interface LDI command control 0.	0x0000_0000
LDI_CMDCON1	0x01D4	Specifies i80 interface LDI command control 1.	0x0000_0000
SIFCCON0	0x01E0	Specifies LCD i80 system interface command control 0.	0x0000_0000
SIFCCON1	0x01E4	Specifies LCD i80 system interface command control 1.	0x0000_0000
SIFCCON2	0x01E8	Specifies LCD i80 system interface command control 2.	0x????_????
HUECOEF_CR_1	0x01EC	Specifies hue coefficient control register.	0x0100_0100
HUECOEF_CR_2	0x01F0	Specifies hue coefficient control register.	0x0000_0000
HUECOEF_CR_3	0x01F4	Specifies hue coefficient control register.	0x0000_0000
HUECOEF_CR_4	0x01F8	Specifies hue coefficient control register.	0x0100_0100
HUECOEF_CB_1	0x01FC	Specifies hue coefficient control register.	0x0100_0100
HUECOEF_CB_2	0x0200	Specifies hue coefficient control register.	0x0000_0000
HUECOEF_CB_3	0x0204	Specifies hue coefficient control register.	0x0000_0000

Register	Offset	Description	Reset Value
HUECOEF_CB_4	0x0208	Specifies hue coefficient control register.	0x0100_0100
HUEOFFSET	0x020C	Specifies hue offset control register.	0x0180_0080
VIDW0ALPHA0	0x021C	Specifies window 0's alpha value 0 register.	0x0000_0000
VIDW0ALPHA1	0x0220	Specifies window 0's alpha value 1 register.	0x0000_0000
VIDW1ALPHA0	0x0224	Specifies window 1's alpha value 0 register.	0x0000_0000
VIDW1ALPHA1	0x0228	Specifies window 1's alpha value 1 register.	0x0000_0000
VIDW2ALPHA0	0x022C	Specifies window 2's alpha value 0 register.	0x0000_0000
VIDW2ALPHA1	0x0230	Specifies window 2's alpha value 1 register.	0x0000_0000
VIDW3ALPHA0	0x0234	Specifies window 3's alpha value 0 register.	0x0000_0000
VIDW3ALPHA1	0x0238	Specifies window 3's alpha value 1 register.	0x0000_0000
VIDW4ALPHA0	0x023C	Specifies window 4's alpha value 0 register.	0x0000_0000
VIDW4ALPHA1	0x0240	Specifies window 4's alpha value 1 register.	0x0000_0000
BLENDEQ1	0x0244	Specifies window 1's blending equation control register.	0x0000_00c2
BLENDEQ2	0x0248	Specifies window 2's blending equation control register.	0x0000_00c2
BLENDEQ3	0x024C	Specifies window 3's blending equation control register.	0x0000_00c2
BLENDEQ4	0x0250	Specifies window 4's blending equation control register.	0x0000_00c2
BLENDCON	0x0260	Specifies blending control register.	0x0000_0000
W0RTQOSCON	0x0264	Specifies window 0's RTQOS control register.	0x0000_0000
W1RTQOSCON	0x0268	Specifies window 1's RTQOS control register.	0x0000_0000
W2RTQOSCON	0x026C	Specifies window 2's RTQOS control register.	0x0000_0000
W3RTQOSCON	0x0270	Specifies window 3's RTQOS control register.	0x0000_0000
W4RTQOSCON	0x0274	Specifies window 4's RTQOS control register.	0x0000_0000
LDI_CMD0	0x0280	Specifies i80 interface LDI command 0.	0x0000_0000
LDI_CMD1	0x0284	Specifies i80 interface LDI command 1.	0x0000_0000
LDI_CMD2	0x0288	Specifies i80 interface LDI command 2.	0x0000_0000
LDI_CMD3	0x028C	Specifies i80 interface LDI command 3.	0x0000_0000
LDI_CMD4	0x0290	Specifies i80 interface LDI command 4.	0x0000_0000
LDI_CMD5	0x0294	Specifies i80 interface LDI command 5.	0x0000_0000
LDI_CMD6	0x0298	Specifies i80 interface LDI command 6.	0x0000_0000
LDI_CMD7	0x029C	Specifies i80 interface LDI command 7.	0x0000_0000
LDI_CMD8	0x02A0	Specifies i80 interface LDI command 8.	0x0000_0000
LDI_CMD9	0x02A4	Specifies i80 interface LDI command 9.	0x0000_0000
LDI_CMD10	0x02A8	Specifies i80 interface LDI command 10.	0x0000_0000
LDI_CMD11	0x02AC	Specifies i80 interface LDI command 11.	0x0000_0000
Gamma LUT Data for 64 Step Mode			
GAMMALUT_01_00	0x037C	Specifies gamma LUT data of the index 0, 1.	0x0010_0000
GAMMALUT_03_02	0x0380	Specifies gamma LUT data of the index 2, 3.	0x0030_0020

Register	Offset	Description	Reset Value
GAMMALUT_05_04	0x0384	Specifies gamma LUT data of the index 4, 5.	0x0050_0040
GAMMALUT_07_06	0x0388	Specifies gamma LUT data of the index 6, 7.	0x0070_0060
GAMMALUT_09_08	0x038C	Specifies gamma LUT data of the index 8, 9.	0x0090_0080
GAMMALUT_11_10	0x0390	Specifies gamma LUT data of the index 10, 11.	0x00B0_00A0
GAMMALUT_13_12	0x0394	Specifies gamma LUT data of the index 12, 13.	0x00D0_00C0
GAMMALUT_15_14	0x0398	Specifies gamma LUT data of the index 14, 15.	0x00F0_00E0
GAMMALUT_17_16	0x039C	Specifies gamma LUT data of the index 16, 17.	0x0110_0100
GAMMALUT_19_18	0x03A0	Specifies gamma LUT data of the index 18, 19.	0x0130_0120
GAMMALUT_21_20	0x03A4	Specifies gamma LUT data of the index 20, 21.	0x0150_0140
GAMMALUT_23_22	0x03A8	Specifies gamma LUT data of the index 22, 23.	0x0170_0160
GAMMALUT_25_24	0x03AC	Specifies gamma LUT data of the index 24, 25.	0x0190_0180
GAMMALUT_27_26	0x03B0	Specifies gamma LUT data of the index 26, 27.	0x01B0_01A0
GAMMALUT_29_28	0x03B4	Specifies gamma LUT data of the index 28, 29.	0x01F0_01C0
GAMMALUT_31_30	0x03B8	Specifies gamma LUT data of the index 30, 31.	0x01F0_01E0
GAMMALUT_33_32	0x03BC	Specifies gamma LUT data of the index 32, 33.	0x0210_0200
GAMMALUT_35_34	0x03C0	Specifies gamma LUT data of the index 34, 35.	0x0230_0220
GAMMALUT_37_36	0x03C4	Specifies gamma LUT data of the index 36, 37.	0x0250_0240
GAMMALUT_39_38	0x03C8	Specifies gamma LUT data of the index 38, 39.	0x0270_0260
GAMMALUT_41_40	0x03CC	Specifies gamma LUT data of the index 40, 41.	0x0290_0280
GAMMALUT_43_42	0x03D0	Specifies gamma LUT data of the index 42, 43.	0x02B0_02A0
GAMMALUT_45_44	0x03D4	Specifies gamma LUT data of the index 44, 45.	0x02D0_02C0
GAMMALUT_47_46	0x03D8	Specifies gamma LUT data of the index 46, 47.	0x02F0_02E0
GAMMALUT_49_48	0x03DC	Specifies gamma LUT data of the index 48, 49.	0x0310_0300
GAMMALUT_51_50	0x03E0	Specifies gamma LUT data of the index 50, 51.	0x0330_0320
GAMMALUT_53_52	0x03E4	Specifies gamma LUT data of the index 52, 53.	0x0350_0340
GAMMALUT_55_54	0x03E8	Specifies gamma LUT data of the index 54, 55.	0x0370_0360
GAMMALUT_57_56	0x03EC	Specifies gamma LUT data of the index 56, 57.	0x0390_0380
GAMMALUT_59_58	0x03F0	Specifies gamma LUT data of the index 58, 59.	0x03B0_03A0
GAMMALUT_61_60	0x03F4	Specifies gamma LUT data of the index 60, 61.	0x03D0_03C0
GAMMALUT_63_62	0x03F8	Specifies gamma LUT data of the index 62, 63.	0x03F0_03E0
GAMMALUT_xx_64	0x03FC	Specifies gamma LUT data of the index 64.	0x0000_0400
Gamma LUT Data for 16 Step Mode			
GAMMALUT_R_1_0	0X037C	Specifies gamma RED LUT data of the index 0, 1.	0X0010_0000
GAMMALUT_R_3_2	0X0380	Specifies gamma RED data of the index 2, 3.	0X0030_0020
GAMMALUT_R_5_4	0X0384	Specifies gamma RED data of the index 4, 5.	0X0050_0040
GAMMALUT_R_7_6	0X0388	Specifies gamma RED data of the index 6, 7.	0X0070_0060
GAMMALUT_R_9_8	0X038C	Specifies gamma RED data of the index 8, 9.	0X0090_0080

Register	Offset	Description	Reset Value
GAMMALUT _R_11_10	0X0390	Specifies gamma RED data of the index 10, 11.	0X00B0_00A0
GAMMALUT _R_13_12	0X0394	Specifies gamma RED data of the index 12, 13.	0X00D0_00C0
GAMMALUT _R_15_14	0X0398	Specifies gamma RED data of the index 14, 15.	0X00F0_00E0
GAMMALUT_R_16	0X039C	Specifies gamma RED data of the index 16.	0X0110_0100
GAMMALUT_R_1_0	0X03A0	Specifies gamma GREEN LUT data of the index 0, 1.	0X0130_0120
GAMMALUT_R_3_2	0X03A4	Specifies gamma GREEN data of the index 2, 3.	0X0150_0140
GAMMALUT_R_5_4	0X03A8	Specifies gamma GREEN data of the index 4, 5.	0X0170_0160
GAMMALUT_R_7_6	0X03AC	Specifies gamma GREEN data of the index 6, 7.	0X0190_0180
GAMMALUT_R_9_8	0X03B0	Specifies gamma GREEN data of the index 8, 9.	0X01B0_01A0
GAMMALUT _R_11_10	0X03B4	Specifies gamma GREEN data of the index 10, 11.	0X01D0_01C0
GAMMALUT _R_13_12	0X03B8	Specifies gamma GREEN data of the index 12, 13.	0X01F0_01E0
GAMMALUT _R_15_14	0X03BC	Specifies gamma GREEN data of the index 14, 15.	0X0210_0200
GAMMALUT_R_16	0X03C0	Specifies gamma GREEN data of the index 16.	0X0230_0220
GAMMALUT_R_1_0	0X03C4	Specifies gamma BLUE data of the index 0, 1.	0X0250_0240
GAMMALUT_R_3_2	0X03C8	Specifies gamma BLUE data of the index 2, 3.	0X0270_0260
GAMMALUT_R_5_4	0X03CC	Specifies gamma BLUE data of the index 4, 5.	0X0290_0280
GAMMALUT_R_7_6	0X03D0	Specifies gamma BLUE data of the index 6, 7.	0X02B0_02A0
GAMMALUT_R_9_8	0X03D4	Specifies gamma BLUE data of the index 8, 9.	0X02D0_02C0
GAMMALUT _R_11_10	0X03D8	Specifies gamma BLUE data of the index 10, 11.	0X02F0_02E0
GAMMALUT _R_13_12	0X03DC	Specifies gamma BLUE data of the index 12, 13.	0X0310_0300
GAMMALUT _R_15_14	0X03E0	Specifies gamma BLUE data of the index 14, 15.	0X0330_0320
GAMMALUT _R_16	0X03E4	Specifies gamma BLUE data of the index 16	0X0350_0340
RSVD	0X03E8	Does not use.	0X0370_0360
RSVD	0X03EC	Does not use.	0X0390_0380
RSVD	0X03F0	Does not use.	0X03B0_03A0
RSVD	0X03F4	Does not use.	0X03D0_03C0
RSVD	0X03F8	Does not use.	0X03F0_03E0
RSVD	0X03FC	Does not use.	0X0000_0400
Shadow Windows Control			

Register	Offset	Description	Reset Value
SHD_VIDW00ADD0	0x40A0	Specifies window 0's buffer start address register (shadow).	0x0000_0000
SHD_VIDW01ADD0	0x40A8	Specifies window 1's buffer start address register (shadow).	0x0000_0000
SHD_VIDW02ADD0	0x40B0	Specifies window 2's buffer start address register (shadow).	0x0000_0000
SHD_VIDW03ADD0	0x40B8	Specifies window 3's buffer start address register (shadow).	0x0000_0000
SHD_VIDW04ADD0	0x40C0	Specifies window 4's buffer start address register (shadow).	0x0000_0000
SHD_VIDW00ADD1	0x40D0	Specifies window 0's buffer end address register (shadow)	0x0000_0000
SHD_VIDW01ADD1	0x40D8	Specifies window 1's buffer end address register (shadow)	0x0000_0000
SHD_VIDW02ADD1	0x40E0	Specifies window 2's buffer end address register (shadow).	0x0000_0000
SHD_VIDW03ADD1	0x40E8	Specifies window 3's buffer end address register (shadow).	0x0000_0000
SHD_VIDW04ADD1	0x40F0	Specifies window 4's buffer end address register (shadow).	0x0000_0000
SHD_VIDW00ADD2	0x4100	Specifies window 0's buffer size register (shadow).	0x0000_0000
SHD_VIDW01ADD2	0x4104	Specifies window 1's buffer size register (shadow).	0x0000_0000
SHD_VIDW02ADD2	0x4108	Specifies window 2's buffer size register (shadow).	0x0000_0000
SHD_VIDW03ADD2	0x410C	Specifies window 3's buffer size register (shadow).	0x0000_0000
SHD_VIDW04ADD2	0x4110	Specifies window 4's buffer size register (shadow).	0x0000_0000

41.5.2 Palette Memory

- Base Address = 0x11C0_0000

Register	Start Address	End Address	Description	Reset Value
Win0 PalRam	0x2400 (0x0400)	0x27FC (0x07FC)	Specifies 0 to 255 entry palette data.	Undefined
Win1 PalRam	0x2800 (0x0800)	0x2BFC (0x0BFC)	Specifies 0 to 255 entry palette data.	Undefined
Win2 PalRam	0x2C00	0x2FFC	Specifies 0 to 255 entry palette data.	Undefined
Win3 PalRam	0x3000	0x33FC	Specifies 0 to 255 entry palette data.	Undefined
Win4 PalRam	0x3400	0x37FC	Specifies 0 to 255 entry palette data.	Undefined

41.5.3 Control Register

41.5.3.1 VIDCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved NOTE: This bit should be set to 0.	0
DSI_EN	[30]	RW	Enables MIPI DSI. 0 = Disables 1 = Enables (i80 24-bit data interface, SYS_ADD[1])	0
RSVD	[29]	–	Reserve NOTE: This bit should be set to 0.	0
VIDOUT	[28:26]	RW	Determines output format of Video Controller. 000 = RGB interface 001 = Reserved 010 = Indirect i80 interface for LDI0 011 = Indirect i80 interface for LDI1 100 = Write-Back interface and RGB interface 101 = Reserved 110 = WB Interface and i80 interface for LDI0 111 = WB Interface and i80 interface for LDI1	000
L1_DATA16	[25:23]	RW	Selects output data format mode of indirect i80 interface (LDI1). (VIDOUT[1:0] == 2'b11) 000 = 16-bit mode (16 BPP) 001 = 16 + 2-bit mode (18 BPP) 010 = 9 + 9-bit mode (18 BPP) 011 = 16 + 8-bit mode (24 BPP) 100 = 18-bit mode (18 BPP) 101 = 8 + 8-bit mode (16 BPP)	000
L0_DATA16	[22:20]	RW	Selects output data format mode of indirect i80 interface (LDI0). (VIDOUT[1:0] == 2'b10) 000 = 16-bit mode (16 BPP) 001 = 16 + 2-bit mode (18 BPP) 010 = 9 + 9-bit mode (18 BPP) 011 = 16 + 8-bit mode (24 BPP) 100 = 18-bit mode (18 BPP) 101 = 8 + 8-bit mode (16BPP)	000
RSVD	[19]	–	Reserved NOTE: This bit should be set to 0.	0
RGSPSEL	[18]	RW	Selects display mode (VIDOUT[1:0] == 2'b00). 0 = RGB parallel format 1 = RGB serial format Selects the display mode (VIDOUT[1:0] ! = 2'b00).	0

Name	Bit	Type	Description	Reset Value
			0 = RGB parallel format	
PNRMODE	[17]	RW	Controls inverting RGB_ORDER (atVIDCON3). 0 = Normal: RGBORDER[2] atVIDCON3 1 = Invert: to RGBORDER[2] atVIDCON3 NOTE: You can use this bit for the previous version of FIMD. You do not have to use this bit if you use RGB_ORDER atVIDCON3 register.	00
CLKVALUP	[16]	RW	Selects CLKVAL_F update timing control. 0 = Always 1 = Start of a frame (only once per frame)	0
RSVD	[15:14]	–	Reserved	0
CLKVAL_F	[13:6]	RW	Determines rates of VCLK and CLKVAL[7:0]. VCLK = FIMD × SCLK/(CLKVAL+1), where CLKVAL ≥ 1 NOTE: The maximum frequency of VCLK is 80 MHz. (80 MHz for Display Controller)	0
VCLKFREE	[5]	RW	Controls VCLK Free Run (only valid at RGB IF mode). 0 = Normal mode (controls using ENVID) 1 = Free-run mode	0
RSVD	[4:2]	–	Reserved NOTE: This bit should be set to 0.	0x0
ENVID	[1]	RW	Enables/disables video output and logic immediately. 0 = Disables the video output and display control signal 1 = Enables the video output and display control signal	0
ENVID_F	[0]	RW	Enables/disables video output and logic at current frame end. 0 = Disables the video output and display control signal 1 = Enables the video output and display control signal If this bit is set to "on" and "off", then "H" is Read and enables the video controller until the end of current frame. (NOTE)	0

NOTE: Display On: ENVID and ENVID_F are set to "1".
Direct Off: ENVID and ENVID_F are set to "0" simultaneously.
Per Frame Off: ENVID_F is set to "0" and ENVID is set to "1".

Caution: 1 = If VIDCON0 is set for Per Frame Off in interlace mode, then the value of INTERLACE_F should be set to "0" in the same time.
2 = If display controller is off using direct off, then it is impossible to turn on the display controller without reset.

41.5.3.2 VIDCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
LINECNT (read only)	[26:16]	RW	Provides status of the line counter (Read only). Up count from 0 to LINEVAL.	0
FSTATUS	[15]	RW	Specifies Field Status (Read only). 0 = ODD Field 1 = EVEN Field	0
VSTATUS	[14:13]	RW	Specifies Vertical Status (Read only). 00 = VSYNC 01 = BACK Porch 10 = ACTIVE 11 = FRONT Porch	0
RSVD	[12:11]	–	Reserved	0
FIXVCLK	[10:9]	RW	Specifies VCLK hold scheme at data under-flow. 00 = VCLK hold 01 = VCLK running 11 = VCLK running and disables VDEN	0
RSVD	[8]	–	Reserved	0
IVCLK	[7]	RW	Controls polarity of the VCLK active edge. 0 = Fetches video data at VCLK falling edge 1 = Fetches video data at VCLK rising edge	0
IHSYNC	[6]	RW	Specifies HSYNC pulse polarity. 0 = Normal 1 = Inverted	0
IVSYNC	[5]	RW	Specifies VSYNC pulse polarity. 0 = Normal 1 = Inverted	0
IVDEN	[4]	RW	Specifies VDEN signal polarity. 0 = Normal 1 = Inverted	0
RSVD	[3:0]	RW	Reserved	0x0

41.5.3.3 VIDCON2

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0
RGB_SKIP_EN	[27]	RW	Enables RGB skip mode. (Only where RGBSPSEL == 1'b0). 0 = Disables 1 = Enables	0
RSVD	[26]	–	Reserved	0
RGB_DUMMY_LOC	[25]	RW	Controls RGB dummy insertion location. (Only where RGBSPSEL == 1'b1 and RGB_DUMMY_EN == 1'b1) 0 = Last (fourth) position 1 = First position	0
RGB_DUMMY_EN	[24]	RW	Enables RGB dummy insertion mode. (Only where RGBSPSEL == 1'b1) 0 = Disables 1 = Enables	0
RSVD	[23:22]	–	Reserved NOTE: This bit should be set to 0.	0
RGB_ORDER_E	[21:19]	RW	Controls RGB interface output order. (Even line, line number 2, 4, 6, 8.), where, RGBSPSEL== 1'b0 000 = RGB 001 = GBR 010 = BRG 100 = BGR 101 = RBG 110 = GRB where, (RGBSPSEL == 1'b1) or (RGBSPSEL == 1'b0 and RGB_SKIP_EN = 1'b1) 000 = R → G → B 001 = G → B → R 010 = B → R → G 100 = B → G → R 101 = R → B → G 110 = G → R→ B NOTE: PNR0[0] atVIDCON0 should be set to 0, when you use RGB_ORDER_O[2:0] at VIDCON3 register.	0
RGB_ORDER_O	[18:16]	RW	Controls RGB interface output order (Odd Line, line number 1, 3, 5, 7.), where, RGBSPSEL == 1'b0 000 = RGB 001 = GBR 010 = BRG 100 = BGR	0

Name	Bit	Type	Description	Reset Value
			101 = RBG 110 = GRB where, (RGBSPSEL == 1'b1) or (RGBSPSEL == 1'b0 and RGB_SKIP_EN = 1'b1) 000 = R → G→ B 001 = G → B→ R 010 = B → R → G 100 = B → G → R 101 = R → B → G 110 = G → R → B NOTE: PNR0[0] at VIDCON0 should be set to 0, when you use RGB_ORDER_E[2:0] at VIDCON3 register.	
RSVD	[15:14]	–	Reserved NOTE: This bit should be set to 1.	0
TVFORMATSEL	[13:12]	RW	Specifies output format of YUV data. 00 = Reserved 01 = YUV422 1x = YUV444	0
RSVD	[11:9]	–	Reserved	0
OrgYCbCr	[8]	RW	Specifies order of YUV data. 0 = Y – CbCr 1 = CbCr – Y	0
YUVOrd	[7]	RW	Specifies order of Chroma data. 0 = Cb – Cr 1 = Cr – Cb	0
RSVD	[6:5]	–	Reserved	0
WB_FRAME_SKIP	[4:0]	RW	Controls WB frame skip rate. The maximum rate is up to 1:30 [only where (VIDOUT[2:0] == 3'b001 or 3'b100 TV encoder interface), (INTERLACE_F == 1'b0) and (TV422 or TVRGB output)]. 00000 = No skip = 1:1 00001 = Skip rate = 1:2 00010 = Skip rate = 1:3 … 11101 = Skip rate = 1: 0 1111x = Reserved	0

41.5.3.4 VIDCON3

- Base Address = 0x11C0_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:21]	–	Reserved NOTE: This bit should be set to 0.	0
RSVD	[20:19]	–	Reserved	0
CG_ON	[18]	RW	Enables Control Color Gain. 0 = Disables (bypass) 1 = Enables	0
RSVD	[17]	–	Reserved	0
GM_ON	[16]	RW	Enables Control Gamma. 0 = Disables (bypass) 1 = Enables	0
GM_MODE	[15]	RW	Gamma mode selection 0 = Applies 64 step identical value to all R, G, B data 1 = Applies 16 step independent value to each R, G, B data	0
HUE_CSC_F_Narrow	[14]	RW	Controls HUE CSC_F Narrow/ Wide. 0 = Wide 1 = Narrow	0
HUE_CSC_F_EQ709	[13]	RW	Controls HUE_CSC_F parameter. 0 = Equation. 601 1 = Equation. 709	0
HUE_CSC_F_ON	[12]	RW	Enables HUE_CSC_F. 0 = Disables 1 = Enables (when HUE_ON == 1'b1)	0
RSVD	[11]	–	Reserved.	0
HUE_CSC_B_Narrow	[10]	RW	Controls HUE CSC_B Narrow/ Wide. 0 = Wide 1 = Narrow	0
HUE_CSC_B_EQ709	[9]	RW	Controls HUE_CSC_B parameter. 0 = Equation 601 1 = Equation 709	0
HUE_CSC_B_ON	[8]	RW	Enables HUE_CSC_B. 0 = Disables 1 = Enables (when HUE_ON == 1'b1)	0
HUE_ON	[7]	RW	Enables Control Hue. 0 = Disables (bypass) 1 = Enables	0
RSVD	[6:2]	–	Reserved NOTE: This bit should be set to 0.	0
PC_DIR	[1]	RW	Controls Pixel Compensation direction.	0

Name	Bit	Type	Description	Reset Value
			0 = + 0.5 (positive) 1 = – 0.5 (negative)	
PC_ON	[3:0]	RW	Enables Pixel Compensation. 0 = Disables 1 = Enables NOTE: PC_ON == 1'b1 compensates the TV output data.	0x0

41.5.3.5 VIDTCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
VBPDE	[31:24]	RW	Vertical back porch specifies the number of inactive lines at the start of a frame after vertical synchronization period. (only for even field of YVU interface)	0x00
VBPD	[23:16]	RW	Vertical back porch specifies the number of inactive lines at the start of a frame after vertical synchronization period.	0x00
VFPD	[15:8]	RW	Vertical front porch specifies the number of inactive lines at the end of a frame before vertical synchronization period.	0x00
VSPW	[7:0]	RW	Vertical synchronization pulse width determines the high-level width of VSYNC pulse by counting the number of inactive lines.	0x00

41.5.3.6 VIDTCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
VFPDE	[31:24]	RW	Vertical front porch specifies the number of inactive lines at the end of a frame before vertical synchronization period. (only for the even field of YVU interface).	0
HBPD	[23:16]	RW	Horizontal back porch specifies the number of VCLK periods between the falling edge of HSYNC and start of active data.	0x00
HFPD	[15:8]	RW	Horizontal front porch specifies the number of VCLK periods between the end of active data and rising edge of HSYNC.	0x00
HSPW	[7:0]	RW	Horizontal synchronization pulse width determines the high-level width of HSYNC pulse by counting the number of VCLK.	0x00

41.5.3.7 VIDTCON2

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
LINEVAL	[21:11]	RW	Determines vertical size of display. In the Interlace mode, (LINEVAL + 1) should be even.	0
HOZVAL	[10:0]	RW	Determines horizontal size of display.	0

NOTE: HOZVAL = (Horizontal display size) – 1 and LINEVAL = (Vertical display size) – 1.

41.5.3.8 VIDTCON3

- Base Address = 0x11C0_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
VSYNCEN	[31]	RW	Enables VSYNC Signal Output. 0 = Disables 1 = Enables VBPD (VFPD, VSPW) + 1 < LINEVAL (when VSYNCEN = 1)	0
RSVD	[30]	–	Reserved NOTE: This bit should be set to 0.	0
FRMEN	[29]	RW	Enables FRM signal output. 0 = Disables 1 = Enables	0
INVFRM	[28]	RW	Controls polarity of FRM pulse. 0 = Active HIGH 1 = Active LOW	0
FRMVRATE	[27:24]	RW	Controls FRM issue rate (maximum rate up to 1:16).	0x00
RSVD	[23:16]	RW	Reserved	0x00
FRMVFPD	[15:8]	RW	Specifies number of line between data active and FRM signal.	0x00
FRMVSPW	[7:0]	RW	Specifies number of line of FRM signal width. (FRMVFPD + 1) + (FRMVSPW + 1) < LINEVAL + 1 (in RGB)	0x00

41.5.3.9 WINCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
BUFSTATUS_H	[31]	RW	Specifies Buffer Status (read only). NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L} 00 = Buffer set to 0 01 = Buffer set to 1 10 = Buffer set to 2	0
BUFSEL_H	[30]	RW	Selects Buffer set. NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L} 00 = Buffer set to 0 01 = Buffer set to 1 10 = Buffer set to 2 (only available where BUF_MODE == 1'b1)	0
LIMIT_ON	[29]	RW	Enables CSC source limiter (for clamping xvYCC source). 0 = Disables 1 = Enables (when local SRC data has xvYCC color space, InRGB = 1)	0
EQ709	[28]	RW	Controls CSC parameter. 0 = Equation. 601 1 = Equation. 709 (when local SRC data has HD (709) color gamut)	0
nWide/Narrow	[27:26]	RW	Chooses color space conversion equation from YCbCr to RGB according to input value range (2'00 for YCbCr w wide range and 2'11 for YCbCr narrow range) Wide Range: Y/Cb/Cr: 255-0 Narrow Range: Y:235-16, Cb/Cr:240-16	00
TRGSTATUS	[25]	RW	Specifies Trigger Status (read only). 0 = Does not issue trigger 1 = Issues trigger	0
RSVD	[24:23]	–	Reserved.	00
ENLOCAL_F	[22]	RW	Selects Data access method. 0 = Dedicated DMA 1 = Local Path	0
BUFSTATUS_L	[21]	RW	Specifies Buffer Status (read only). NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}	0
BUFSEL_L	[20]	RW	Selects Buffer set. NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}	0
BUFAUTOEN	[19]	RW	Specifies Double Buffer Auto control bit. 0 = Fixed by BUFSEL 1 = Auto Changed by Trigger Input	0
BITSWP_F	[18]	RW	Specifies Bit swap control bit. 0 = Disables swap	0

Name	Bit	Type	Description	Reset Value
			1 = Enables swap NOTE: It should be set to 0 when ENLOCAL is 1.	
BYTSWP_F	[17]	RW	Specifies Byte swaps control bit. 0 = Disables swap 1 = Enables swap NOTE: It should be set to 0 when ENLOCAL is 1.	0
HAWSWP_F	[16]	RW	Specifies Half-Word swap control bit. 0 = Disables swap 1 = Enables swap NOTE: It should be set to 0 when ENLOCAL is 1.	0
WSWP_F	[15]	RW	Specifies Word swap control bit. 0 = Disables swap 1 = Enables swap NOTE: It should be set to 0 when ENLOCAL is 1.	0
BUF_MODE	[14]	RW	Selects auto-buffering mode. 0 = Double 1 = Triple	0
InRGB	[13]	RW	Specifies input color space of source image. (Only for "ENLOCAL" enable). 0 = RGB 1 = YCbCr	0
RSVD	[12:11]	–	Reserved NOTE: This bit should be set to 0.	0
BURSTLEN	[10:9]	RW	Selects DMA Burst Maximum Length. 00 = 16 word-burst 01 = 8 word-burst 10 = 4 word-burst	0
RSVD	[8:7]	–	Reserved NOTE: This bit should be set to 0.	0
BLD_PIX_F	[6]	RW	Selects blending category (In case of window0, this is required only for deciding window 0's blending factor.) 0 = Per plane blending 1 = Per pixel blending	0
BPPMODE_F	[5:2]	RW	Selects Bits Per Pixel (BPP) mode for Window image. 0000 = 1 BPP 0001 = 2 BPP 0010 = 4 BPP 0011 = 8 BPP (palletized) 0100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 BPP (non-palletized, R:5-G:6-B:5) 0110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 1000 = Unpacked 18 BPP (non-palletized, R:6-G:6-B:6) 1001 = Unpacked 18 BPP (non-palletized, A:1-R:6-G:6-B:5)	0

Name	Bit	Type	Description	Reset Value
			1010 = Unpacked 19 BPP (non-palletized, A:1-R:6-G:6-B:6) 1011 = Unpacked 24 BPP (non-palletized R:8-G:8-B:8) 1100 = Unpacked 24 BPP (non-palletized A:1-R:8-G:8-B:7) 1101 = Unpacked 25 BPP (non-palletized A:1-R:8-G:8-B:8) 1110 = Unpacked 13 BPP (non-palletized A:1-R:4-G:4-B:4) 1111 = Unpacked 15 BPP (non-palletized R:5-G:5-B:5) NOTE: 1. 1101 = Supports unpacked 32 BPP (non-palletized A:8-R:8-G:8-B:8) for per pixel blending. 2. 1110 = Supports 16 BPP (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending)	
ALPHA_SEL_F	[1]	RW	Selects Alpha value. When per plane blending case (BLD_PIX == 0): 0 = Uses ALPHA0_R/G/B values 1 = Uses ALPHA1_R/G/B values When per pixel blending (BLD_PIX == 1): 0 = Selected by AEN (A value) 1 = Using DATA[31:24] data in word boundary (only when BPPMODE_F = 4'b1101) DATA[31:28], [15:12] data in word boundary (only when BPPMODE_F = 4'b1110)	0
ENWIN_F	[0]	RW	Enables/disables video output and logic immediately. 0 = Disables the video output and video control signal 1 = Enables the video output and video control signal	0

41.5.3.10 WINCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
BUFSTATUS_H	[31]	RW	Specifies Buffer Status (read only). 00 = Buffer set to 0 01 = Buffer set to 1 10 = Buffer set to 2 NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}	0
BUFSEL_H	[30]	RW	Select Buffer set. 00 = Buffer set to 0 01 = Buffer set to 1 10 = Buffer set to 2 (only available when BUF_MODE == 1'b1) NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}	0
LIMIT_ON	[29]	RW	Enables Control CSC source limiter (for clamping xvYCC source). 0 = Disables 1 = Enables (when local SRC data has xvYCC color space, InRGB = 1)	0
EQ709	[28]	RW	Controls CSC parameter. 0 = Equation. 601 1 = Equation. 709 (when local SRC data has HD (709) color gamut)	0
nWide/Narrow	[27:26]	RW	Chooses color space conversion equation from YCbCr to RGB based on input value range (2'00 for YCbCr wide range and 2'11 for YCbCr narrow range). Wide Range: Y/Cb/Cr: 255-0 Narrow Range: Y:235-16, Cb/Cr: 240-16	00
TRGSTATUS	[25]	RW	Specifies Window 0 Software Trigger Update Status (read only). 0 = Updates 1 = Does not update If the Software Trigger in window 1 occurs, then this bit is automatically set to "1". It clears this value only after updating the shadow register sets.	0
RSVD	[24:23]	–	Reserved NOTE: This bit should be set to 0.	0
ENLOCAL_F	[22]	RW	Selects Data access method. 0 = Dedicated DMA 1 = Local Path	0
BUFSTATUS_L	[21]	RW	Specifies Buffer Status (Read only). NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}	0
BUFSEL_L	[20]	RW	Selects Buffer set.	0

Name	Bit	Type	Description	Reset Value
			NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}	
BUFAUTOEN	[19]	RW	Specifies Double Buffer Auto control bit. 0 = Fixed by BUFSEL 1 = Auto changed by Trigger Input	0
BITSWP_F	[18]	RW	Specifies Bit swap control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1.	0
BYTSWP_F	[17]	RW	Specifies Byte swaps control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1.	0
HAWSWP_F	[16]	RW	Specifies Half-Word swap control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1.	0
WSWP_F	[15]	RW	Specifies Word swap control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1.	0
BUF_MODE	[14]	RW	Selects auto-buffering mode. 0 = Double 1 = Triple	0
InRGB	[13]	RW	Indicates input color space of source image. (Only for "EnLcal" enable). 0 = RGB 1 = YCbCr	0
RSVD	[12:11]	–	Reserved NOTE: This bit should be set to 0.	0
BURSTLEN	[10:9]	RW	Specifies DMA's Burst Maximum Length selection. 00 = 16word-burst 01 = 8word-burst 10 = 4word-burst	0
RSVD	[8]	–	Reserved NOTE: This bit should be set to 0.	0
ALPHA_MUL_F	[7]	RW	Specifies Multiplied Alpha value mode. 0 = Disables multiplied mode 1 = Enables multiplied mode When ALPHA_MUL is 1, set BLD_PIX = 1, ALPHA_SEL = 1, and BPPMODE_F[5:2] = 4'b1101 or 4'b1110. NOTE: Alpha value = alpha_pixel (from data) × ALPHA0_R/G/B	0

Name	Bit	Type	Description	Reset Value
BLD_PIX_F	[6]	RW	Selects blending category. 0 = Per plane blending 1 = Per pixel blending	0
BPPMODE_F	[5:2]	RW	Selects Bits Per Pixel (BPP) mode in Window image. 0000 = 1 BPP 0001 = 2 BPP 0010 = 4 BPP 0011 = 8 BPP (palletized) 0100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 BPP (non-palletized, R:5-G:6-B:5) 0110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 1000 = Unpacked 18 BPP (non-palletized, R:6-G:6-B:6) 1001 = Unpacked 18 BPP (non-palletized, A:1-R:6-G:6-B:5) 1010 = Unpacked 19 BPP (non-palletized, A:1-R:6-G:6-B:6) 1011 = Unpacked 24 BPP (non-palletized R:8-G:8-B:8) 1100 = Unpacked 24 BPP (non-palletized A:1-R:8-G:8-B:7) 1101 = Unpacked 25 BPP (non-palletized A:1-R:8-G:8-B:8) 1110 = Unpacked 13 BPP (non-palletized A:1-R:4-G:4-B:4) 1111 = Unpacked 15 BPP (non-palletized R:5-G:5-B:5) NOTE: 1. 1101 = Supports unpacked 32 BPP (non-palletized A:8-R:8-G:8-B:8) for per pixel blending. 2. 1110 = Supports 16 BPP (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending)	0
ALPHA_SEL_F	[1]	RW	Selects Alpha value. When per plane blending case (BLD_PIX == 0) 0 = Using ALPHA0_R/G/B values 1 = Using ALPHA1_R/G/B values When per pixel blending (BLD_PIX == 1) 0 = Selected by AEN (A value) 1 = Using DATA[31:24] data in word boundary (only when BPPMODE_F = 4'b1101) DATA[31:28], [15:12] data in word boundary (only when BPPMODE_F = 4'b1110)	0
ENWIN_F	[0]	RW	Enables/disables video output and logic immediately. 0 = Disables the video output and video control signal 1 = Enables the video output and video control signal	0

41.5.3.11 WINCON2

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
BUFSTATUS_H	[31]	RW	Specifies Buffer Status (Read only). 00 = Buffer set to 0 01 = Buffer set to 1 10 = Buffer set to 2 NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}	0
BUFSEL_H	[30]	RW	Selects Buffer set. 00 = Buffer set to 0 01 = Buffer set to 1 10 = Buffer set to 2 (only available when BUF_MODE == 1'b1) NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}	0
LIMIT_ON	[29]	RW	Enables CSC source limiter (for clamping xvYCC source). 0 = Disables 1 = Enables (when local SRC data has xvYCC color space, InRGB = 1)	0
EQ709	[28]	RW	Controls CSC parameter. 0 = Equation.601 1 = Equation 709 (when local SRC data has HD (709) color gamut)	0
nWide/Narrow	[27:26]	RW	Chooses color space conversion equation from YCbCr to RGB based on the input value range (2'00 for YCbCr wide range and 2'11 for YCbCr narrow range). Wide Range: Y/Cb/Cr: 255-0 Narrow Range: Y: 235-16, Cb/Cr: 240-16	00
RSVD	[25:24]	–	Reserved	00
LOCALSEL_F	[23]	RW	Selects local path source. 0 = CAMIF2 1 = CAMIF3	0
ENLOCAL_F	[22]	RW	Selects Data access method. 0 = Dedicated DMA 1 = Local Path	0
BUFSTATUS_L	[21]	RW	Specifies Buffer Status (read only). NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}	
BUFSEL_L	[20]	RW	Selects Buffer set. NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}	0
BUFAUTOEN	[19]	RW	Specifies Double Buffer Auto control bit. 0 = Fixed by BUFSEL 1 = Auto changed by Trigger Input	0
BITSWP_F	[18]	RW	Specifies the Bit swap control bit. 0 = Disables swap	0

Name	Bit	Type	Description	Reset Value
			1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1.	
BYTSWP_F	[17]	RW	Specifies Byte swaps control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1.	0
HAWSWP_F	[16]	RW	Specifies Half-Word swap control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 0 when ENLOCAL is 1.	0
WSWP_F	[15]	RW	Specifies Word swap control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1.	0
BUF_MODE	[14]	RW	Selects auto-buffering mode. 0 = Double 1 = Triple	0
InRGB	[13]	RW	Specifies input color space of source image (only for "EnLcal" enable). 0 = RGB 1 = YCbCr	0
RSVD	[12:11]	–	Reserved NOTE: This bit should be set to 0.	0
BURSTLEN	[10:9]	RW	Selects the DMA's Burst Maximum Length. 00 = 16 word-burst 01 = 8 word-burst 10 = 4 word-burst	0
RSVD	[8]	–	Reserved (should be 0).	0
ALPHA_MUL_F	[7]	RW	Specifies Multiplied Alpha value mode. 0 = Disables multiplied mode 1 = Enables multiplied mode When ALPHA_MUL is 1, set BLD_PIX = 1, ALPHA_SEL = 1, and BPPMODE_F[5:2] = 4'b1101 or 4'b1110. NOTE: Alpha value = alpha_pixel (from data) × ALPHA0_R/G/B	0
BLD_PIX_F	[6]	RW	Selects blending category. 0 = Per plane blending 1 = Per pixel blending	0
BPPMODE_F	[5:2]	RW	Selects Bits Per Pixel (BPP) mode in Window image. 0000 = 1 BPP 0001 = 2 BPP 0010 = 4 BPP 0011 = 8 BPP (palletized)	0

Name	Bit	Type	Description	Reset Value
			0100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 BPP (non-palletized, R:5-G:6-B:5) 0110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 1000 = Unpacked 18 BPP (non-palletized, R:6-G:6-B:6) 1001 = Unpacked 18 BPP (non-palletized, A:1-R:6-G:6-B:5) 1010 = Unpacked 19 BPP (non-palletized, A:1-R:6-G:6-B:6) 1011 = Unpacked 24 BPP (non-palletized R:8-G:8-B:8) 1100 = Unpacked 24 BPP (non-palletized A:1-R:8-G:8-B:7) 1101 = Unpacked 25 BPP (non-palletized A:1-R:8-G:8-B:8) 1110 = Unpacked 13 BPP (non-palletized A:1-R:4-G:4-B:4) 1111 = Unpacked 15 BPP (non-palletized R:5-G:5-B:5) NOTE: 1. 1101 = Supports unpacked 32 BPP (non-palletized A:8-R:8-G:8-B:8) for per pixel blending. 2. 1110 = Supports 16 BPP (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending)	
ALPHA_SEL_F	[1]	RW	Selects Alpha value. When Per plane blending case BLD_PIX == 0: 0 = Using ALPHA0_R/G/B values 1 = Using ALPHA1_R/G/B values When Per pixel blending BLD_PIX == 1: 0 = Selected by AEN (A value) 1 = Using DATA[31:24] data in word boundary (only when BPPMODE_F = 4'b1101) DATA[31:28], [15:12] data in word boundary (only when BPPMODE_F = 4'b1110)	0
ENWIN_F	[0]	RW	Enables/disables the video output and logic immediately. 0 = Disables the video output and video control signal 1 = Enables the video output and video control signal	0

41.5.3.12 WINCON3

- Base Address = 0x11C0_0000
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
BUFSTATUS_H	[31]	RW	Specifies Buffer Status (read only). 00 = Buffer is set to 0 01 = Buffer is set to 1 10 = Buffer is set to 2 NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}	–
BUFSEL_H	[30]	RW	Selects Buffer set 00 = Buffer is set to 0 01 = Buffer is set to 1 10 = Buffer is set to 2 (only available where BUF_MODE == 1'b1) NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}	–
RSVD	[29:26]	–	Reserved NOTE: This bit should be set to 0.	–
TRIGSTATUS	[25]	RW	Specifies Trigger Status (read only) 0 = No trigger is issued 1 = Trigger is issued	–
RSVD	[24:22]	–	Reserved NOTE: This bit should be set to 0.	–
BUFSTATUS_L	[21]	RW	Specifies Buffer Status (read only). NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}	–
BUFSEL_L	[20]	RW	Selects Buffer set. NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}	–
BUFAUTOEN	[19]	RW	Specifies Double Buffer Auto control bit. 0 = Fixed by BUFSEL 1 = Auto changed by Trigger Input	–
BITSWP_F	[18]	RW	Specifies Bit swap control bit. 0 = Disables swap 1 = Enables swap	0
BYTSWP_F	[17]	RW	Specifies Byte swaps control bit. 0 = Disables swap 1 = Enables swap	0
HAWSWP_F	[16]	RW	Specifies Half-Word swap control bit. 0 = Disables swap 1 = Enables swap	0
WSWP_F	[15]	RW	Specifies Word swap control bit. 0 = Disables swap 1 = Enables swap	
BUF_MODE	[14]	RW	Selects auto-buffering mode. 0 = Double	0

Name	Bit	Type	Description	Reset Value
			1 = Triple	
RSVD	[13:11]	–	Reserved NOTE: This bit should be set to 0.	0
BURSTLEN	[10:9]	RW	Selects DMA Burst Maximum Length. 00 = 16 word- burst 01 = 8 word- burst 10 = 4 word- burst	0
RSVD	[8]	–	Reserved NOTE: This bit should be set to 0.	0
ALPHA_MUL_F	[7]	RW	Specifies Multiplied Alpha value mode. 0 = Disables multiplied mode 1 = Enables multiplied mode When ALPHA_MUL is 1, set BLD_PIX = 1, ALPHA_SEL = 1, and BPPMODE_F[5:2] = 4'b1101 or 4'b1110. NOTE. Alpha value = alpha_pixel (from data) × ALPHA0_R/G/B	0
BLD_PIX_F	[6]	RW	Selects blending category. 0 = Per plane blending 1 = Per pixel blending	
BPPMODE_F	[5:2]	RW	Selects Bits Per Pixel (BPP) mode in Window image. 0000 = 1 BPP 0001 = 2 BPP 0010 = 4 BPP 0011 = 8 BPP (palletized) 0100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 BPP (non-palletized, R:5-G:6-B:5) 0110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 1000 = Unpacked 18 BPP (non-palletized, R:6-G:6-B:6) 1001 = Unpacked 18 BPP (non-palletized, A:1-R:6-G:6-B:5) 1010 = Unpacked 19 BPP (non-palletized, A:1-R:6-G:6-B:6) 1011 = Unpacked 24 BPP (non-palletized R:8-G:8-B:8) 1100 = Unpacked 24 BPP (non-palletized A:1-R:8-G:8-B:7) 1101 = Unpacked 25 BPP (non-palletized A:1-R:8-G:8-B:8) 1110 = Unpacked 13 BPP (non-palletized A:1-R:4-G:4-B:4) 1111 = Unpacked 15 BPP (non-palletized R:5-G:5-B:5) NOTE: 1. 1101 = Supports unpacked 32 BPP (non-palletized A:8-R:8-G:8-B:8) for per pixel blending. 2. 1110 = Supports 16 BPP (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending)	0

Name	Bit	Type	Description	Reset Value
ALPHA_SEL_F	[1]	RW	Selects Alpha value. When Per plane blending case BLD_PIX == 0: 0 = Uses ALPHA0_R/G/B values 1 = Uses ALPHA1_R/G/B values When Per pixel blending BLD_PIX == 1: 0 = Selected by AEN (A value) 1 = Uses DATA[31:24] data in word boundary (only when BPPMODE_F = 4'b1101) DATA[31:28], [15:12] data in word boundary (only when BPPMODE_F = 4'b1110)	0
ENWIN_F	[0]	RW	Enables/disables video output and logic immediately. 0 = Disables the video output and video control signal 1 = Enables the video output and video control signal	0

41.5.3.13 WINCON4

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
BUFSTATUS_H	[31]	RW	Specifies Buffer Status (read only). 00 = Buffer is set to 0 01 = Buffer is set to 1 10 = Buffer is set to 2 NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}	0
BUFSEL_H	[30]	RW	Selects Buffer set. 00 = Buffer is set to 0 01 = Buffer is set to 1 10 = Buffer is set to 2 (only available where BUF_MODE == 1'b1) NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}	0
RSVD	[29:26]	–	Reserved NOTE: This bit should be set to 0.	0
TRIGSTATUS	[25]	RW	Specifies Trigger Status (read only). 0 = Does not issue trigger 1 = Issues trigger	0
RSVD	[24:22]	–	Reserved NOTE: This bit should be set o 0.	0
BUFSTATUS_L	[21]	RW	Specifies Buffer Status (read only). NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}	0
BUFSEL_L	[20]	RW	Selects Buffer set. NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}	0
BUFAUTOEN	[19]	RW	Specifies Double Buffer Auto control bit. 0 = Fixed by BUFSEL 1 = Auto changed by Trigger Input	0
BITSWP_F	[18]	RW	Specifies Bit swap control bit. 0 = Disables swap 1 = Enables swap	0
BYTSWP_F	[17]	RW	Specifies Byte swap control bit. 0 = Disables swap 1 = Enables swap	0
HAWSWP_F	[16]	RW	Specifies Half-Word swap control bit. 0 = Disables swap 1 = Enables swap	0
WSWP_F	[15]	RW	Specifies Word swap control bit. 0 = Disables swap 1 = Enables swap	0
BUF_MODE	[14]	RW	Selects auto-buffering mode. 0 = Double	0

Name	Bit	Type	Description	Reset Value
			1 = Triple	
RSVD	[13:11]	–	Reserved NOTE: This bit should be set to 0	0
BURSTLEN	[10:9]	RW	Selects DMA Burst Maximum Length. 00 = 16 word-burst 01 = 8 word-burst 10 = 4 word-burst	0
RSVD	[8]	–	Reserved NOTE: This bit should be set to 0.	0
ALPHA_MUL_F	[7]	RW	Specifies Multiplied Alpha value mode. 0 = Disables multiplied mode 1 = Enables multiplied mode When ALPHA_MUL is 1, set BLD_PIX = 1, ALPHA_SEL = 1, and BPPMODE_F[5:2] = 4'b1101 or 4'b1110. NOTE: Alpha value = alpha_pixel (from data) × ALPHA0_R/G/B	0
BLD_PIX_F	[6]	RW	Selects blending category. 0 = Per plane blending 1 = Per pixel blending	0
BPPMODE_F	[5:2]	RW	Selects Bits Per Pixel (BPP) mode in Window image. 0000 = 1 BPP 0001 = 2 BPP 0010 = 4 BPP 0011 = 8 BPP (palletized) 0100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 BPP (non-palletized, R:5-G:6-B:5) 0110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 1000 = Unpacked 18 BPP (non-palletized, R:6-G:6-B:6) 1001 = Unpacked 18 BPP (non-palletized, A:1-R:6-G:6-B:5) 1010 = Unpacked 19 BPP (non-palletized, A:1-R:6-G:6-B:6) 1011 = Unpacked 24 BPP (non-palletized, R:8-G:8-B:8) 1100 = Unpacked 24 BPP (non-palletized, A:1-R:8-G:8-B:7) 1101 = Unpacked 25 BPP (non-palletized, A:1-R:8-G:8-B:8) 1110 = Unpacked 13 BPP (non-palletized, A:1-R:4-G:4-B:4) 1111 = Unpacked 15 BPP (non-palletized, R:5-G:5-B:5) NOTE: 1. 1101 = Support unpacked 32 BPP (non-palletized, A:8-R:8-G:8-B:8) for per pixel blending. 2. 1110 = Support 16 BPP (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending)	0

Name	Bit	Type	Description	Reset Value
ALPHA_SEL_F	[1]	RW	Selects Alpha value. When Per plane blending case BLD_PIX == 0 : 0 = Uses ALPHA0_R/G/B values 1 = Uses ALPHA1_R/G/B values When Per pixel blending BLD_PIX == 1 : 0 = Selected by AEN (A value) 1 = Uses DATA[31:24] data in word boundary (only when BPPMODE_F = 4'b1101) DATA[31:28], [15:12] data in word boundary (only when BPPMODE_F = 4'b1110)	0
ENWIN_F	[0]	RW	Enables/disables video output and logic immediately. 0 = Disables the video output and video control signal 1 = Enables the video output and video control signal	0

41.5.3.14 SHADOWCON

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:15]	–	Reserved NOTE: This bit should be set to 0	0
W4_SHADOW_PROTECT	[14]	RW	Protects to update window 4's shadow register (xxx_F). 0 = Updates shadow register per frame 1 = Protects to update (updates shadow register at next frame after "SHADOW_PROTECT" turns to be 1'b0)	0
W3_SHADOW_PROTECT	[13]	RW	Protects to update window 3's shadow register (xxx_F) 0 = Updates shadow register per frame 1 = Protects to update (updates shadow register at next frame after "SHADOW_PROTECT" turns to be 1'b0)	0
W2_SHADOW_PROTECT	[12]	RW	Protects to update window 2's shadow register (xxx_F) 0 = Updates shadow register per frame 1 = Protects to update (updates shadow register at next frame after "SHADOW_PROTECT" turns to be 1'b0)	0
W1_SHADOW_PROTECT	[11]	RW	Protects to update window 1's shadow register (xxx_F) 0 = Updates shadow register per frame 1 = Protects to update (update shadow register at next frame after "SHADOW_PROTECT" turns to be 1'b0)	0
W0_SHADOW_PROTECT	[10]	RW	Protects to update window 0's shadow register (xxx_F) 0 = Updates shadow register per frame 1 = Protects to update (update shadow register at next frame after "SHADOW_PROTECT" turns to be 1'b0)	0
RSVD	[9:8]	–	Reserved	0
C2_ENLOCAL_F	[7]	RW	Enables Channel 2 Local Path. 0 = Disables 1 = Enables	0
C1_ENLOCAL_F	[6]	RW	Enables Channel 1 Local Path. 0 = Disables 1 = Enables	0
C0_ENLOCAL_F	[5]	RW	Enables Channel 0 Local Path. 0 = Disables 1 = Enables	0
C4_EN_F	[4]	RW	Enables Channel 4. 0 = Disables 1 = Enables	0
C3_EN_F	[3]	RW	Enables Channel 3. 0 = Disables 1 = Enables	0
C2_EN_F	[2]	RW	Enables Channel 2. 0 = Disables	0

Name	Bit	Type	Description	Reset Value
			1 = Enables	
C1_EN_F	[1]	RW	Enables Channel 1. 0 = Disables 1 = Enables	0
C0_EN_F	[0]	RW	Enables Channel 0. 0 = Disables 1 = Enables	0

41.5.3.15 WINCHMAP2

- Base Address = 0x11C0_0000
- Address = Base Address + 0x003C, Reset Value = 0x7D51_7D51

Name	Bit	Type	Description	Reset Value
CH4FISEL	[30:28]	RW	Selects Channel 4's channel. 001 = Window 0 010 = Window 1 101 = Window 2 110 = Window 3 111 = Window 4	111
CH3FISEL	[27:25]	RW	Selects Channel 3's channel. 001 = Window 0 010 = Window 1 101 = Window 2 110 = Window 3 111 = Window 4	110
CH2FISEL	[24:22]	RW	Selects Channel 2's channel. 001 = Window 0 010 = Window 1 101 = Window 2 110 = Window 3 111 = Window 4	101
CH1FISEL	[21:19]	RW	Selects Channel 1's channel. 001 = Window 0 010 = Window 1 101 = Window 2 110 = Window 3 111 = Window 4	010
CH0FISEL	[18:16]	RW	Selects Channel 0's channel. 001 = Window 0 010 = Window 1 101 = Window 2 110 = Window 3 111 = Window 4	001
W4FISEL	[14:12]	RW	Selects Window 4's channel. 001 = Channel 0 010 = Channel 1 101 = Channel 2 110 = Channel 3 111 = Channel 4	111
W3FISEL	[11:9]	RW	Selects Window 3's channel. 001 = Channel 0 010 = Channel 1 101 = Channel 2 110 = Channel 3 111 = Channel 4	110

Name	Bit	Type	Description	Reset Value
W2FISEL	[8:6]	RW	Selects Window 2's channel. 001 = Channel 0 010 = Channel 1 101 = Channel 2 110 = Channel 3 111 = Channel 4	101
W1FISEL	[5:3]	RW	Selects Window 1's channel. 001 = Channel 0 010 = Channel 1 101 = Channel 2 110 = Channel 3 111 = Channel 4	010
W0FISEL	[2:0]	RW	Selects Window 0's channel. 001 = Channel 0 010 = Channel 1 101 = Channel 2 110 = Channel 3 111 = Channel 4	001

41.5.3.16 VIDOSD0A

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OSD_LeftTopX_F	[21:11]	RW	Specifies the horizontal screen coordinate for left top pixel of OSD image.	0
OSD_LeftTopY_F	[10:0]	RW	Specifies the vertical screen coordinate for left top pixel of OSD image. (For interlace TV output, this value should be set to half of the original screen y coordinate. The original screen y coordinate should be even.)	0

41.5.3.17 VIDOSD0B

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OSD_RightBotX_F	[21:11]	RW	Specifies horizontal screen coordinate for right bottom pixel of OSD image.	0
OSD_RightBotY_F	[10:0]	RW	Specifies vertical screen coordinate for right bottom pixel of OSD image. (For interlace TV output, this value should be set to half of the original screen y coordinate. The original screen y coordinate should be odd value.)	0

NOTE: Registers should have word boundary X position.
Therefore, 24 BPP mode should have X position by 1 pixel. (For example, X = 0, 1, 2, 3….)
16 BPP mode should have X position by 2 pixel. (For example, X = 0, 2, 4, 6….)
8 BPP mode should have X position by 4 pixel. (For example, X = 0, 4, 8, 12….)

41.5.3.18 VIDOSD0C

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0048, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[25:24]	–	Reserved NOTE: This bit should be set to 0.	0
OSDSIZE	[23:0]	RW	Specifies the Window Size For example, Height × Width (number of word)	0

41.5.3.19 VIDOSD0C

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OSD_LeftTopX_F	[21:11]	RW	Specifies Horizontal screen coordinate for left top pixel of OSD image.	0
OSD_LeftTopY_F	[10:0]	RW	Specifies Vertical screen coordinate for left top pixel of OSD image. (For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be even.)	0

41.5.3.20 VIDOSD1B

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OSD_RightBotX_F	[21:11]	RW	Specifies horizontal screen coordinate for right bottom pixel of OSD image.	0
OSD_RightBotY_F	[10:0]	RW	Specifies vertical screen coordinate for right bottom pixel of OSD image. (For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be odd value.)	0

NOTE: Registers should have word boundary X position.
Therefore, 24 BPP mode should have X position by 1 pixel. (For example, X = 0, 1, 2, 3….)
16 BPP mode should have X position by 2pixel. (For example, X = 0, 2, 4, 6….)
8 BPP mode should have X position by 4pixel. (For example, X = 0, 4, 8, 12….)

41.5.3.21 VIDOSD1C

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0058, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
ALPHA0_R_H_F	[23:20]	RW	Specifies Red Alpha upper value (case AEN == 0)	0
ALPHA0_G_H_F	[19:16]	RW	Specifies Green Alpha upper value (case AEN == 0)	0
ALPHA0_B_H_F	[15:12]	RW	Specifies Blue Alpha upper value (case AEN == 0)	0
ALPHA1_R_H_F	[11:8]	RW	Specifies Red Alpha upper value (case AEN == 1)	0
ALPHA1_G_H_F	[7:4]	RW	Specifies Green Alpha upper value (case AEN == 1)	0
ALPHA1_B_H_F	[3:0]	RW	Specifies Blue Alpha upper value (case AEN == 1)	0

NOTE: For more information, refer to VIDW1ALPHA0, 1 register.

41.5.3.22 VIDOSD1D

- Base Address = 0x11C0_0000
- Address = Base Address + 0x005C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[25:24]	–	Reserved NOTE: This bit should be set to 0.	0
OSDSIZE	[23:0]	RW	Specifies Window Size. For example, Height × Width(number of word)	0

41.5.3.23 VIDOSD2A

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OSD_LeftTopX_F	[21:11]	RW	Specifies horizontal screen coordinate for left top pixel of OSD image.	0
OSD_LeftTopY_F	[10:0]	RW	Specifies vertical screen coordinate for left top pixel of OSD image. For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be even value.	0

41.5.3.24 VIDOSD2B

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0064, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OSD_RightBotX_F	[21:11]	RW	Specifies horizontal screen coordinate for right bottom pixel of OSD image.	0
OSD_RightBotY_F	[10:0]	RW	Specifies vertical screen coordinate for right bottom pixel of OSD image. (For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be odd value.)	0

NOTE: Registers should have word boundary X position.
Therefore, 24 BPP mode should have X position by 1 pixel. (For example, X = 0, 1, 2, 3….)
16 BPP mode should have X position by 2 pixel. (For example, X = 0, 2, 4, 6….)
8 BPP mode should have X position by 4 pixel. (For example, X = 0, 4, 8, 12….)

41.5.3.25 VIDOSD2C

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0068, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
ALPHA0_R_H_F	[23:20]	RW	Specifies the Red Alpha upper value (case AEN == 0).	0
ALPHA0_G_H_F	[19:16]	RW	Specifies the Green Alpha upper value (case AEN == 0).	0
ALPHA0_B_H_F	[15:12]	RW	Specifies the Blue Alpha upper value (case AEN == 0).	0
ALPHA1_R_H_F	[11:8]	RW	Specifies the Red Alpha upper value (case AEN == 1).	0
ALPHA1_G_H_F	[7:4]	RW	Specifies the Green Alpha upper value (case AEN == 1).	0
ALPHA1_B_H_F	[3:0]	RW	Specifies the Blue Alpha upper value (case AEN == 1).	0

NOTE: For more information, refer to VIDW2ALPHA0, 1 register.

41.5.3.26 VIDOSD2D

- Base Address = 0x11C0_0000
- Address = Base Address + 0x006C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[25:24]	–	Reserved NOTE: This bit should be set to 0.	0
OSDSIZE	[23:0]	RW	Specifies Window Size For example, Height × Width(Number of Word)	0

41.5.3.27 VIDOSD3A

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0070, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OSD_LeftTopX_F	[21:11]	RW	Specifies Horizontal screen coordinate for left top pixel of OSD image.	0
OSD_LeftTopY_F	[10:0]	RW	Specifies Vertical screen coordinate for left top pixel of OSD image. For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be even value.	0

41.5.3.28 VIDOSD3B

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0074, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OSD_RightBotX_F	[21:11]	RW	Specifies Horizontal screen coordinate for right bottom pixel of OSD image.	0
OSD_RightBotY_F	[10:0]	RW	Specifies Vertical screen coordinate for right bottom pixel of OSD image. (For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be odd value.)	0

NOTE: Registers should have word boundary X position.
Therefore, 24 BPP mode should have X position by 1 pixel. (For example, X = 0, 1, 2, 3….)
16 BPP mode should have X position by 2 pixel. (For example, X = 0, 2, 4, 6….)
8 BPP mode should have X position by 4 pixel. (For example, X = 0, 4, 8, 12….)

41.5.3.29 VIDOSD3C

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0078, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
ALPHA0_R_H_F	[23:20]	RW	Specifies the Red Alpha upper value (case AEN == 0).	0
ALPHA0_G_H_F	[19:16]	RW	Specifies the Green Alpha upper value (case AEN == 0).	0
ALPHA0_B_H_F	[15:12]	RW	Specifies the Blue Alpha upper value (case AEN == 0).	0
ALPHA1_R_H_F	[11:8]	RW	Specifies the Red Alpha upper value (case AEN == 1).	0
ALPHA1_G_H_F	[7:4]	RW	Specifies the Green Alpha upper value (case AEN == 1).	0
ALPHA1_B_H_F	[3:0]	RW	Specifies the Blue Alpha upper value (case AEN == 1).	0

NOTE: For more information, Refer to *41.5.3.76* *41.5.3.77* VIDW3ALPHA0, 1 register.

41.5.3.30 VIDOSD4A

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OSD_LeftTopX_F	[21:11]	RW	Specifies the Horizontal screen coordinate for left top pixel of OSD image.	0
OSD_LeftTopY_F	[10:0]	RW	Specifies the Vertical screen coordinate for left top pixel of OSD image. For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be even value.	0

41.5.3.31 VIDOSD4B

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0084, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OSD_RightBotX_F	[21:11]	RW	Specifies Horizontal screen coordinate for right bottom pixel of OSD image.	0
OSD_RightBotY_F	[10:0]	RW	Specifies Vertical screen coordinate for right bottom pixel of OSD image. For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be odd value.	0

NOTE: Registers should have word boundary X position.
Therefore, 24 BPP mode should have X position by1 pixel. (For example, X = 0, 1, 2, 3….)
16 BPP mode should have X position by 2 pixel. (For example, X = 0, 2, 4, 6….)
8 BPP mode should have X position by 4 pixel. (For example, X = 0, 4, 8, 12….)

41.5.3.32 VIDOSD4C

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0088, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
ALPHA0_R_H_F	[23:20]	RW	Specifies the Red Alpha upper value (case AEN == 0).	0
ALPHA0_G_H_F	[19:16]	RW	Specifies the Green Alpha upper value (case AEN == 0).	0
ALPHA0_B_H_F	[15:12]	RW	Specifies the Blue Alpha upper value (case AEN == 0).	0
ALPHA1_R_H_F	[11:8]	RW	Specifies the Red Alpha upper value (case AEN == 1).	0
ALPHA1_G_H_F	[7:4]	RW	Specifies the Green Alpha upper value (case AEN == 1).	0
ALPHA1_B_H_F	[3:0]	RW	Specifies the Blue Alpha upper value (case AEN == 1).	0

NOTE: For more information, Refer to *41.5.3.78* *41.5.3.79* VIDW4ALPHA0, 1 register

41.5.3.33 VIDW0nADD0Bn

- Base Address = 0x11C0_0000
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000 (VIDW00ADD0B0)
- Address = Base Address + 0x00A4, Reset Value = 0x0000_0000 (VIDW00ADD0B1)
- Address = Base Address + 0x20A0, Reset Value = 0x0000_0000 (VIDW00ADD0B2)
- Address = Base Address + 0x00A8, Reset Value = 0x0000_0000 (VIDW01ADD0B0)
- Address = Base Address + 0x00AC, Reset Value = 0x0000_0000 (VIDW01ADD0B1)
- Address = Base Address + 0x20A8, Reset Value = 0x0000_0000 (VIDW01ADD0B2)
- Address = Base Address + 0x00B0, Reset Value = 0x0000_0000 (VIDW02ADD0B0)
- Address = Base Address + 0x00B4, Reset Value = 0x0000_0000 (VIDW02ADD0B1)
- Address = Base Address + 0x20B0, Reset Value = 0x0000_0000 (VIDW02ADD0B2)
- Address = Base Address + 0x00B8, Reset Value = 0x0000_0000 (VIDW03ADD0B0)
- Address = Base Address + 0x00BC, Reset Value = 0x0000_0000 (VIDW03ADD0B1)
- Address = Base Address + 0x20B8, Reset Value = 0x0000_0000 (VIDW03ADD0B2)
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000 (VIDW04ADD0B0)
- Address = Base Address + 0x00C4, Reset Value = 0x0000_0000 (VIDW04ADD0B1)
- Address = Base Address + 0x20C0, Reset Value = 0x0000_0000 (VIDW04ADD0B2)

Name	Bit	Type	Description	Reset Value
VBASEU_F	[31:0]	RW	Specifies A[31:0] of the start address for video frame buffer.	0

41.5.3.34 VIDW0nADD1Bn

- Base Address = 0x11C0_0000
- Address = Base Address + 0x00D0, Reset Value = 0x0000_0000 (VIDW00ADD1B0)
- Address = Base Address + 0x00D4, Reset Value = 0x0000_0000 (VIDW00ADD1B1)
- Address = Base Address + 0x20D0, Reset Value = 0x0000_0000 (VIDW00ADD1B2)
- Address = Base Address + 0x00D8, Reset Value = 0x0000_0000 (VIDW01ADD1B0)
- Address = Base Address + 0x00DC, Reset Value = 0x0000_0000 (VIDW01ADD1B1)
- Address = Base Address + 0x20D8, Reset Value = 0x0000_0000 (VIDW01ADD1B2)
- Address = Base Address + 0x00E0, Reset Value = 0x0000_0000 (VIDW02ADD1B0)
- Address = Base Address + 0x00E4, Reset Value = 0x0000_0000 (VIDW02ADD1B1)
- Address = Base Address + 0x20E0, Reset Value = 0x0000_0000 (VIDW02ADD1B2)
- Address = Base Address + 0x00E8, Reset Value = 0x0000_0000 (VIDW03ADD1B0)
- Address = Base Address + 0x00EC, Reset Value = 0x0000_0000 (VIDW03ADD1B1)
- Address = Base Address + 0x20E8, Reset Value = 0x0000_0000 (VIDW03ADD1B2)
- Address = Base Address + 0x00F0, Reset Value = 0x0000_0000 (VIDW04ADD1B0)
- Address = Base Address + 0x00F4, Reset Value = 0x0000_0000 (VIDW04ADD1B1)
- Address = Base Address + 0x20F0, Reset Value = 0x0000_0000 (VIDW04ADD1B2)

Name	Bit	Type	Description	Reset Value
VBASEL_F	[31:0]	RW	Specifies A[31:0] of the end address for video frame buffer. VBASEL = VBASEU + (PAGEWIDTH + OFFSIZE) × (LINEVAL + 1)	0x0

41.5.3.35 VIDW0nADD2 (n = 0 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0100, + 0x0104, + 0x0108, + 0x010C, + 0x0110, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OFFSIZE_F	[25:13]	RW	Specifies virtual screen offset size (number of byte). This value defines the difference between address of last byte which displays on the previous video line and address of first byte which will display in the new video line. OFFSIZE_F should have value that is multiple of 4byte size or 0.	0
PAGEWIDTH_F	[12:0]	RW	Specifies virtual screen page width (number of byte). This value defines the width of view port in the frame. PAGEWIDTH should have bigger value than the burst size and you should align the size word boundary.	0

NOTE: You should align the sum of PAGEWIDTH_F and OFFSIZE_F double-word (8 byte) boundary.

41.5.3.36 VIDINTCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0130, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	0
FIFOINTERVAL	[25:20]	RW	Controls interval of the FIFO interrupt.	0
SYSMAINCON	[19]	RW	Sends complete interrupt enable bit to Main LCD 0 = Disables Interrupt 1 = Enables Interrupt NOTE: This bit is valid if both INTEN and I80IFDONE are high.	0
SYSSUBCON	[18]	RW	Sends complete interrupt enable bit to Sub LCD 0 = Disables Interrupt 1 = Enables Interrupt NOTE: This bit is valid l if both INTEN and I80IFDONE are high.	0
I80IFDONE	[17]	RW	Enables i80 Interface Interrupt (only for I80 Interface mode). 0 = Disables Interrupt1 = Enables Interrupt NOTE: This bit is valid if INTEN is high.	0
FRAMESEL0	[16:15]	RW	Specifies Video Frame Interrupt 0 at start of: 00 = BACK Porch 01 = VSYNC 10 = ACTIVE 11 = FRONT Porch	0
FRAMESEL1	[14:13]	RW	Specifies Video Frame Interrupt 1 at start of: 00 = None 01 = BACK Porch 10 = VSYNC 11 = FRONT Porch	0
INTFRMEN	[12]	RW	Specifies Video Frame Interrupt Enable Control Bit. 0 = Disables Video Frame Interrupt 1 = Enables Video Frame Interrupt NOTE: This bit is valid l when INTEN is high.	0
FIFOSEL	[11:5]	RW	Specifies FIFO Interrupt control bit. Each bit has a special significance: [11] Window 4 control 0 = Disables 1 = Enables [10] Window 3 control 0 = Disables 1 = Enables [9] Window 2 control 0 = Disables	0

Name	Bit	Type	Description	Reset Value
			1 = Enables [8] Reserved [7] Reserved [6] Window 1 control 0 = Disables 1 = Enables [5] Window 0 control 0 = Disables 1 = Enables NOTE: This bit is valid l if both INTEN and INTFIFOEN are high.	
FIFOLEVEL	[4:2]	RW	Selects Video FIFO Interrupt Level. 000 = 0 – 25 % 001 = 0 – 50 % 010 = 0 – 75 % 011 = 0 % (empty) 100 = 100 % (full)	0
INTFIFOEN	[1]	RW	Specifies Video FIFO Interrupt Enable Control Bit. 0 = Disables video FIFO level interrupt 1 = Enables video FIFO level interrupt NOTE: This bit is valid if INTEN is high.	0
INTEN	[0]	RW	Specifies Video Interrupt Enable Control Bit. 0 = Disables video interrupt 1 = Enables video interrupt	0

NOTE:

1. If video frame interrupt occurs, then you can select maximum two points by setting FRAMESEL0 and FRAMESEL1. For example, in case of FRAMESEL0 = 00 and FRAMESEL1 = 11, it triggers video frame interrupt both at the start of back porch and front porch.
2. Interrupt controller has three interrupt sources related to display controller, namely, LCD[0], LCD[1], and LCD[2]. (For more information, refer to Chapter 9 interrupt controller"). LCD[0] specifies FIFO Level interrupt, LCD[1] specifies video frame synchronization interrupt and LCD[2] specifies i80 done interface interrupt.

41.5.3.37 VIDINTCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0134, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0
RSVD	[4:3]	–	Reserved NOTE: This bit should be set to 0.	0
INTI80PEND	[2]	RW	Specifies i80 done interrupt. Writes "1" to clear this bit. 0 = Does not request interrupt 1 = i80 done status asserts the interrupt request	0
INTFRMPEND	[1]	RW	Specifies frame synchronization interrupt. Writes "1" to clear this bit. 0 = Does not request interrupt 1 = Frame synchronization status asserts the interrupt request	0
INTFIFOPEND	[0]	RW	Specifies FIFO Level interrupt. Writes "1" to clear this bit. 0 = Does not request interrupt 1 = FIFO empty status asserts the interrupt request.	0

41.5.3.38 W1KEYCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0140, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
KEYBLEN_F	[26]	RW	Enables blending. 0 = Disables blending 1 = Enables blending using original alpha for non-key area and KEY_ALPHA for key area	0
KEYEN_F	[25]	RW	Enables/Disables Color Key (Chroma key). 0 = Disables color key 1 = Enables color key	0
DIRCON_F	[24]	RW	Controls color key (Chroma key) direction. 0 = If the pixel value matches foreground image with COLVAL, then it displays the pixel from background image (only in OSD area) 1 = If the pixel value matches background image with COLVAL, then it displays the pixel from foreground image (only in OSD area)	0
COMPKEY_F	[23:0]	RW	Each bit corresponds to COLVAL [23:0]. If some position bit is set, then it disables the position bit of COLVAL.	0

NOTE: Set BLD_PIX = 1, ALPHA_SEL = 0, A_FUNC = 0x2, and B_FUNC = 0x3 to enable alpha blending using color key.

41.5.3.39 W1KEYCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0144, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
COLVAL_F	[23:0]	RW	Specifies color key value for transparent pixel effect.	0

41.5.3.40 W2KEYCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0148, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
KEYBLEN_F	[26]	RW	Enables blending. 0 = Disables blending 1 = Enables blending using original alpha for non-key area and KEY_ALPHA for key area	0
KEYEN_F	[25]	RW	Enables color key (Chroma key). 0 = Disables color key 1 = Enables color key	0
DIRCON_F	[24]	RW	Controls color key (Chroma key) direction. 0 = If the pixel value matches foreground image with COLVAL, then it displays the pixel from background image (only in OSD area) 1 = If the pixel value matches background image with COLVAL, then it displays the pixel from foreground image (only in OSD area)	0
COMPKEY_F	[23:0]	RW	Each bit corresponds to COLVAL [23:0]. If some position bit is set, then it disables the position bit of COLVAL.	0

NOTE: Set BLD_PIX = 1, ALPHA_SEL = 0, A_FUNC = 0x2, and B_FUNC = 0x3 to enable alpha blending using color key.

41.5.3.41 W2KEYCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x014C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
COLVAL_F	[23:0]	RW	Specifies color key value for transparent pixel effect.	0

41.5.3.42 W3KEYCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0150, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
KEYBLEN_F	[26]	RW	Enables blending. 0 = Disables blending 1 = Enables blending using original alpha for non-key area and KEY_ALPHA for key area	0
KEYEN_F	[25]	RW	Enables Color Key (Chroma key). 0 = Disables color key 1 = Enables color key	0
DIRCON_F	[24]	RW	Controls Color key (Chroma key) direction. 0 = If the pixel value matches foreground image with COLVAL, then it displays the pixel from background image (only in OSD area) 1 = If the pixel value matches background image with COLVAL, then it displays the pixel from foreground image (only in OSD area)	0
COMPKEY_F	[23:0]	RW	Each bit corresponds to COLVAL [23:0]. If some position bit is set, then it disables the position bit of COLVAL.	0

NOTE: Set BLD_PIX = 1, ALPHA_SEL = 0, A_FUNC = 0x2, and B_FUNC = 0x3 to enable alpha blending using color key.

41.5.3.43 W3KEYCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0154, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
COLVAL_F	[23:0]	RW	Specifies color key value for transparent pixel effect.	0

41.5.3.44 W4KEYCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0158, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
KEYBLEN_F	[26]	RW	Enables blending. 0 = Disables blending 1 = Enables blending using original alpha for non-key area and KEY_ALPHA for key area	0
KEYEN_F	[25]	RW	Enables color Key (Chroma key). 0 = Disables color key 1 = Enables color key	0
DIRCON_F	[24]	RW	Controls color key (Chroma key) direction. 0 = If the pixel value matches foreground image with COLVAL, then it displays the pixel from background image (only in OSD area) 1 = If the pixel value matches background image with COLVAL, then it displays the pixel from foreground image (only in OSD area)	0
COMPKEY_F	[23:0]	RW	Each bit corresponds to COLVAL [23:0]. If some position bit is set, then it disables the COLVAL position bit.	0

NOTE: Set BLD_PIX = 1, ALPHA_SEL = 0, A_FUNC = 0x2, and B_FUNC = 0x3 to enable alpha blending using color key.

41.5.3.45 W4KEYCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x015C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
COLVAL_F	[23:0]	RW	Specifies color key value for transparent pixel effect.	0

NOTE: Both COLVAL and COMPKEY use 24-bit color data in all BPP modes.

At 24 BPP Mode: 24-bit color value is valid.

- A. COLVAL
  - Red: COLVAL[23:17]
  - Green: COLVAL[15: 8]
  - Blue: COLVAL[7:0]
- B. COMPKEY
  - Red: COMPKEY[23:17]
  - Green: COMPKEY[15: 8]
  - Blue: COMPKEY[7:0]

At 16 BPP (5:6:5) mode: 16-bit color value is valid.

- A. COLVAL
  - Red: COLVAL[23:19]
  - Green: COLVAL[15: 10]
  - Blue: COLVAL[7:3]
- B. COMPKEY
  - Red: COMPKEY[23:19]
  - Green: COMPKEY[15: 10]
  - Blue: COMPKEY[7:3]
  - COMPKEY[18:16] should be 0x7.
  - COMPKEY[9: 8] should be 0x3.
  - COMPKEY[2:0] should be 0x7.

NOTE: COMPKEY register should be set properly for each BPP mode.

41.5.3.46 W1KEYALPHA

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0160, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	–	Reserved	0
KEYALPHA_R_F	[23:0]	RW	Specifies Key alpha R value.	0
KEYALPHA_G_F	[15:8]	RW	Specifies Key alpha G value.	0
KEYALPHA_B_F	[7:0]	RW	Specifies Key alpha B value.	0

41.5.3.47 W2KEYALPHA

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0164, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	–	Reserved	0
KEYALPHA_R_F	[23:0]	RW	Specifies Key alpha R value.	0
KEYALPHA_G_F	[15:8]	RW	Specifies Key alpha G value.	0
KEYALPHA_B_F	[7:0]	RW	Specifies Key alpha B value.	0

41.5.3.48 W3KEYALPHA

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0168, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	–	Reserved	0
KEYALPHA_R_F	[23:0]	RW	Specifies Key alpha R value.	0
KEYALPHA_G_F	[15:8]	RW	Specifies Key alpha G value.	0
KEYALPHA_B_F	[7:0]	RW	Specifies Key alpha B value.	0

41.5.3.49 W4KEYALPHA

- Base Address = 0x11C0_0000
- Address = Base Address + 0x016C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	–	Reserved.	0
KEYALPHA_R_F	[23:0]	RW	Specifies Key alpha R value.	0
KEYALPHA_G_F	[15:8]	RW	Specifies Key alpha G value.	0
KEYALPHA_B_F	[7:0]	RW	Specifies Key alpha B value.	0

41.5.3.50 DITHMODE

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0170, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Does not use for normal access (writing not-zero values to these registers results in abnormal behavior.)	0
RDithPos	[6:5]	RW	Controls Red Dither bit. 00 = 8-bit 01 = 6-bit 10 = 5-bit	0
GDithPos	[4:3]	RW	Controls Green Dither bit. 00 = 8-bit 01 = 6-bit 10 = 5-bit	0
BDithPos	[2:1]	RW	Controls Blue Dither bit. 00 = 8-bit 01 = 6-bit 10 = 5-bit	0
DITHEN_F	[0]	RW	Enables Dithering bit. 0 = Disables dithering 1 = Enables dithering	0

41.5.3.51 WIN0MAP

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0180, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MAPCOLEN_F	[24]	RW	Specifies color mapping of window control bit. If it enables this bit, then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables	0
MAPCOLOR	[23:0]	RW	Specifies color value.	0

41.5.3.52 WIN1MAP

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0184, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MAPCOLEN_F	[24]	RW	Specifies the color mapping of window control bit. If it enables this bit, then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables	0
MAPCOLOR	[23:0]	RW	Specifies the color value.	0

41.5.3.53 WIN2MAP

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0188, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MAPCOLEN_F	[24]	RW	Specifies the color mapping of window control bit. If it enables this bit, then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables	0
MAPCOLOR	[23:0]	RW	Specifies color value.	0

41.5.3.54 WIN3MAP

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0_018C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MAPCOLEN_F	[24]	RW	Specifies color mapping of window control bit. If it enables this bit, then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables	0
MAPCOLOR	[23:0]	RW	Specifies color value.	0

41.5.3.55 WIN4MAP

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0190, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MAPCOLEN_F	[24]	RW	Specifies color mapping of window control bit. If it enables this bit, then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables	0
MAPCOLOR	[23:0]	RW	Specifies color value.	0

41.5.3.56 WPALCON_H

- Base Address = 0x11C0_0000
- Address = Base Address + 0x019C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:19]	–	Reserved	0
W4PAL_H	[18:17]	RW	W4PAL[2:1]	0
RSVD	[16:15]	RW	Reserved	0
W3PAL_H	[14:13]	RW	W3PAL[2:1]	0
RSVD	[12:11]	RW	Reserved	0
W2PAL_H	[10: 9]	RW	W2PAL[2:1]	0
RSVD	[8: 0]	RW	Reserved	0

41.5.3.57 WPALCON_L

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:23]	–	Reserved	0
PALUPDATEEN	[9]	RW	0 = Normal Mode 1 = Enables (Palette Update)	0
W4PAL_L	[8]	RW	W4PAL[0]	0
W3PAL_L	[7]	RW	W3PAL[0]	0
W2PAL_L	[6]	RW	W2PAL[0]	0
W1PAL_L	[5: 3]	RW	W1PAL[2:0]	0
W0PAL_L	[2: 0]	RW	W0PAL[2:0]	0

NOTE:

1. WPALCON = {WPALCON_H,WPALCON_L}

Name	Description	Reset Value
PALUPDATEEN	0 = Normal Mode 1 = Enables (Palette Update)	0
W4PAL[3:0]	Specifies size of palette data format of Window 4. 000 = 16-bit (5:6:5) 001 = 16-bit (A:5:5:5) 010 = 18-bit (6:6:6) 011 = 18-bit (A:6:6:5) 100 = 19-bit (A:6:6:6) 101 = 24-bit (8:8:8) 110 = 25-bit (A:8:8:8) 111 = 32-bit (8:8:8:8) (A: 8-bit)	0
W3PAL[2:0]	Specifies size of palette data format of Window 3. 000 = 16-bit (5:6:5) 001 = 16-bit (A:5:5:5) 010 = 18-bit (6:6:6) 011 = 18-bit (A:6:6:5) 100 = 19-bit (A:6:6:6) 101 = 24-bit (8:8:8) 110 = 25-bit (A:8:8:8) 111 = 32-bit (8:8:8:8) (A: 8-bit)	0
W2PAL[2:0]	Specifies size of palette data format of Window 2. 000 = 16-bit (5:6:5) 001 = 16-bit (A:5:5:5) 010 = 18-bit (6:6:6) 011 = 18-bit (A:6:6:5) 100 = 19-bit (A:6:6:6) 101 = 24-bit (8:8:8) 110 = 25-bit (A:8:8:8) 111 = 32-bit (8:8:8:8) (A: 8-bit)	0

Name	Description	Reset Value
W1PAL[2:0]	Specifies size of palette data format of Window 1. 000 = 25-bit (A:8:8:8) 001 = 24-bit (8:8:8) 010 = 19-bit (A:6:6:6) 011 = 18-bit (A:6:6:5) 100 = 18-bit (6:6:6) 101 = 16-bit (A:5:5:5) 110 = 16-bit (5:6:5) 111 = 32-bit (8:8:8:8) (A: 8-bit)	0
W0PAL[2:0]	Specifies size of palette data format of Window 0. 000 = 25-bit (A:8:8:8) 001 = 24-bit (8:8:8) 010 = 19-bit (A:6:6:6) 011 = 18-bit (A:6:6:5) 100 = 18-bit (6:6:6) 101 = 16-bit (A:5:5:5) 110 = 16-bit (5:6:5) 111 = 32-bit (8:8:8:8) (A: 8-bit)	0

2. The bit map for W0/ W1 is different from W2/W3/W4.

41.5.3.58 TRIGCON

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01A4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0
SWTRGCMD_W4BUF	[26]	RW	Specifies Window 4 double buffer trigger. 1 = Enables Software Trigger Command (write only) *Only when TRGMODE_W4BUF is set to "1"	0
TRGMODE_W4BUF	[25]	RW	Specifies Window 4 double buffer trigger. 0 = Disables trigger 1 = Enables trigger	0
RSVD	[24:22]	–	Reserved	0
SWTRGCMD_W3BUF	[21]	RW	Specifies Window 3 double buffer trigger. 1 = Enables Software Trigger Command (write only) *Only when TRGMODE_W3BUF is set to "1"	0
TRGMODE_W3BUF	[20]	RW	Specifies Window 3 double buffer trigger. 0 = Disables trigger 1 = Enables trigger	0
RSVD	[19:17]	–	Reserved	0
SWTRGCMD_W2BUF	[16]	RW	Specifies Window 2 double buffer trigger. 1 = Enables Software Trigger Command (write only) *Only when TRGMODE_W2BUF is set to "1"	0
TRGMODE_W2BUF	[15]	RW	Specifies Window 2 double buffer trigger. 0 = Disables trigger 1 = Enables trigger	0
RSVD	[14:12]	–	Reserved	0
SWTRGCMD_W1BUF	[11]	RW	Specifies Window 1 double buffer trigger. 1 = Enables Software Trigger Command (write only) *Only when TRGMODE_W1BUF is set to "1"	0
TRGMODE_W1BUF	[10]	RW	Specifies Window 1 double buffer trigger. 0 = Disables trigger 1 = Enables trigger	0
RSVD	[9:7]	–	Reserved	0
SWTRGCMD_W0BUF	[6]	RW	Specifies Window 0 double buffer trigger. 1 = Enables Software Trigger Command (write only) *Only when TRGMODE_W0BUF is set to "1"	0
TRGMODE_W0BUF	[5]	RW	Specifies Window 0 double buffer trigger. 0 = Disables trigger 1 = Enables trigger	0
RSVD	[4:3]	–	Reserved	0
SWFRSTATUS	[2]	RW	Specifies Frame Done Status (read only; i80 start	0

Name	Bit	Type	Description	Reset Value
_I80			trigger) 0 = Does not request 1 = Requests *Clear Condition: Read or New Frame Start *Only when TRGMODE is set to "1"	
SWTRGCMD _I80	[1]	RW	Enables i80 start trigger. 1 = Software Triggering Command (write only) *Only when TRGMODE is set to"1"	0
TRGMODE_I80	[0]	RW	Enables i80 start trigger. 0 = Disables i80 Software Trigger 1 = Enables i80 Software Trigger	0

NOTE: Generates two continuous software trigger inputs in some video clocks (VCLK) recognizes as one.

41.5.3.59 I80IFCONAn (n = 0 to 1)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01B0, + 0x01B4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[22:20]	–	Reserved	0
LCD_CS_SETUP	[19:16]	RW	Specifies number of clock cycles for the active period of address signal enable to chip select enable.	0
LCD_WR_SETUP	[15:12]	RW	Specifies number of clock cycles for the active period of CS signal enable to write signal enable.	0
LCD_WR_ACT	[11:8]	RW	Specifies number of clock cycles for the active period of chip select enable.	0
LCD_WR_HOLD	[7:4]	RW	Specifies number of clock cycles for the active period of chip select disable to write signal disable.	0
RSVD	[3]	–	Reserved	
RSPOL	[2]	RW	Specifies polarity of RS Signal 0 = Low 1 = High	0
RSVD	[1]	–	Reserved	0
I80IFEN	[0]	RW	Controls the LCD i80 interface. 0 = Disables 1 = Enables	0

41.5.3.60 I80IFCONBn (n = 0 to 1)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01B8, + 0x01BC , Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[11:10]	–	Reserved	0
NORMAL_CMD_ST	[9]	RW	1 = Normal Command Start * Auto clears after sending out one set of commands	0
RSVD	[8:7]	–	Reserved	
FRAME_SKIP	[6:5]	RW	Specifies i80 Interface Output Frame Decimation Factor. 00 = 1 (Does not Skip) 01 = 2 10 = 3	00
RSVD	[4]	–	Reserved	0
AUTO_CMD_RATE	[3:0]	RW	0000 = Disables auto command (if you do not use any auto-command, then you should set AUTO_CMD_RATE as "0000".) 0001 = per 2 Frames 0010 = per 4 Frames 0011 = per 6 Frames … 1111 = per 30 Frames	0000

41.5.3.61 COLORGAINCON

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01C0, Reset Value = 0x1004_0100

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
CG_RGAIN	[29:20]	RW	Specifies color gain value of R data (maximum 4, 8-bit resolution). 0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256) … 0h0FF = 0.99609375 (255/256) 0h100 = 1.0 … 0x3FF = 3.99609375 (maximum)	0x100
CG_GGAIN	[19:10]	RW	Specifies color gain value of G data (maximum 4, 8-bit resolution). 0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256) … 0h0FF = 0.99609375 (255/256) 0h100 = 1.0 … 0x3FF = 3.99609375 (maximum)	0x100
CG_BGAIN	[9:0]	RW	Specifies color gain value of B data (maximum 4, 8-bit resolution). 0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256) … 0h0FF = 0.99609375 (255/256) 0h100 = 1.0 … 0x3FF = 3.99609375 (maximum)	0x100

41.5.3.62 LDI_CMDCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01D0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	–
CMD11_EN	[23:22]	RW	Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD10_EN	[21:20]	RW	Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD9_EN	[19:18]	RW	Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD8_EN	[17:16]	RW	Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD7_EN	[15:14]	RW	Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD6_EN	[13:12]	RW	Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD5_EN	[11:10]	RW	Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD4_EN	[9:8]	RW	Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00

Name	Bit	Type	Description	Reset Value
CMD3_EN	[7:6]	RW	Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD2_EN	[5:4]	RW	Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Normal and Auto Command Enable	00
CMD1_EN	[3:2]	RW	Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD0_EN	[1:0]	RW	Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00

41.5.3.63 LDI_CMDCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01D4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved.	0
CMD11_RS	[11]	RW	Controls Command 11 RS	0
CMD10_RS	[10]	RW	Controls Command 10 RS	0
CMD9_RS	[9]	RW	Controls Command 9 RS	0
CMD8_RS	[8]	RW	Controls Command 8 RS	0
CMD7_RS	[7]	RW	Controls Command 7 RS	0
CMD6_RS	[6]	RW	Controls Command 6 RS	0
CMD5_RS	[5]	RW	Controls Command 5 RS	0
CMD4_RS	[4]	RW	Controls Command 4 RS	0
CMD3_RS	[3]	RW	Controls Command 3 RS	0
CMD2_RS	[2]	RW	Controls Command 2 RS	0
CMD1_RS	[1]	RW	Controls Command 1 RS	0
CMD0_RS	[0]	RW	Controls Command 0 RS	0

41.5.3.64 SIFCCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01E0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[7]	–	Reserved NOTE: This bit should be set to 0.	0
SYS_ST_CON	[6]	RW	Controls LCD i80 System Interface ST Signal. 0 = Low 1 = High	0
SYS_RS_CON	[5]	RW	Controls LCD i80 System Interface RS Signal. 0 = Low 1 = High	0
SYS_nCS0_CON	[4]	RW	Controls LCD i80 System Interface nCS0 (main) Signal. 0 = Disables (High) 1 = Enables (Low)	0
SYS_nCS1_CON	[3]	RW	Controls LCD i80 System Interface nCS1 (sub) Signal. 0 = Disables (High) 1 = Enables (Low)	0
SYS_nOE_CON	[2]	RW	Controls LCD i80 System Interface nOE Signal. 0 = Disables (High) 1 = Enables (Low)	0
SYS_nWE_CON	[1]	RW	Controls LCD i80 System Interface nWE Signal. 0 = Disables (High) 1 = Enables (Low)	0
SCOMEN	[0]	RW	Enables LCD i80 System Interface Command Mode. 0 = Disables (Normal Mode) 1 = Enables (Manual Command Mode)	

41.5.3.65 SIFCCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01E4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SYS_WDATA	[23:0]	RW	Controls LCD i80 System Interface Write Data.	0

41.5.3.66 SIFCCON2

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01E8, Reset Value = 0x????_????

Name	Bit	Type	Description	Reset Value
SYS_RDATA	[23:0]	R	Controls LCD i80 System Interface Read Data.	0

41.5.3.67 HUECOEF_CR_n (n = 1 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01EC, Reset Value = 0x0100_0100
- Address = Base Address + 0x01F0, Reset Value = 0x0000_0000
- Address = Base Address + 0x01F4, Reset Value = 0x0000_0000
- Address = Base Address + 0x01F8, Reset Value = 0x0100_0100

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	0
CRG0_x	[25:16]	RW	Specifies Hue matrix coefficient 00 (when "cb + In_offset" is positive). (Signed) 0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256) … 0h0FF = 0.99609375 (255/256) 0h100 = 1.0 (256/256) 0h300 = – 1.0 (– 256/256) 0h301 = – 0.99609375 (– 255/256) … 0h3FF = – 0.00390625 (– 1/256) 0h101 to 2FF = Reserved (do not use)	0x100
RSVD	[15:10]	–	Reserved	0
CRG1_x	[9:0]	RW	Specifies Hue matrix coefficient 00 (when "cb + In_offset" is negative). (Signed) 0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256) … 0h0FF = 0.99609375 (255/256) 0h100 = 1.0 (256/256) 0h300 = – 1.0 (– 256/256) 0h301 = – 0.99609375 (– 255/256) … 0h3FF = – 0.00390625 (– 1/256) 0h101 to 2FF = Reserved (do not use)	0x100

41.5.3.68 HUECOEF_CB_n (n = 1 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01FC, Reset Value = 0x0100_0100
- Address = Base Address + 0x0200, Reset Value = 0x0000_0000
- Address = Base Address + 0x0204, Reset Value = 0x0000_0000
- Address = Base Address + 0x0208, Reset Value = 0x0100_0100

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	0
CBG0_x	[25:16]	RW	Specifies Hue matrix coefficient 00 (when "cb + In_offset" is positive). (Signed) 0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256) … 0h0FF = 0.99609375 (255/256) 0h100 = 1.0 (256/256) 0h300 = – 1.0 (– 256/256) 0h301 = – 0.99609375 (– 255/256) … 0h3FF = – 0.00390625 (– 1/256) 0h101 to 2FF = Reserved (do not use)	0x100
RSVD	[15:10]	–	Reserved	0
CBG1_x	[9:0]	RW	Specifies Hue matrix coefficient 00 (when "cb + In_offset" is negative). (Signed) 0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256) … 0h0FF = 0.99609375 (255/256) 0h100 = 1.0 (256/256) 0h300 = – 1.0 (– 256/256) 0h301 = – 0.99609375 (– 255/256) … 0h3FF = – 0.00390625 (– 1/256) 0h101 to 2FF = Reserved (do not use)	0x100

41.5.3.69 HUEOFFSET

- Base Address = 0x11C0_0000
- Address = Base Address + 0x020C, Reset Value = 0x0108_0080

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0
OFFSET_IN	[24:16]	RW	Specifies Hue matrix input offset (signed). 0h000 = + 0 0h001 = + 1 0h002 = + 2 … 0h0FF = + 255 0h100 = – 256 … 0x1FF = –1	0x180 (-128)
RSVD	[15:9]	–	Reserved	0
OFFSET_OUT	[8:0]	RW	Specifies Hue matrix output offset (signed). 0h000 = + 0 0h001 = + 1 0h002 = + 2 … 0h0FF = + 255 0h100 = –256 … 0x1FF = – 1	0x080 (+128)

NOTE: Generally, HUE_OFFSET_IN = – 128 and HUE_OFFSET_OUT = + 128

Example 41-7 Hue Equation

```
Cb<hue> = CBG0•(Cb + OFFSET_IN) + CBG1•(Cr + OFFSET_IN) + OFFSET_OUT
Cr<hue> = CRG0•(Cb + OFFSET_IN) + CRG1•(Cr + OFFSET_IN) + OFFSET_OUT
```

Example 41-8 Coefficient Decision

```
CBG0 = (Cb - 128) ≥ 0 ? CBG0_P: CBG0_N
CBG1 = (Cr - 128) ≥ 0 ? CBG1_P: CBG1_N
CRG0 = (Cb - 128) ≥ 0 ? CRG0_P: CRG0_N
CRG0 = (Cr- 1 28) ≥ 0 ? CRG1_P: CRG1_N
```

41.5.3.70 VIDW0ALPHA0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x021C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
ALPHA0_R_F	[23:16]	RW	Specifies Red Alpha value (case AEN == 0).	0
ALPHA0_G_F	[15:8]	RW	Specifies Green Alpha value (case AEN == 0).	0
ALPHA0_B_F	[7:0]	RW	Specifies Blue Alpha value (case AEN == 0).	0

41.5.3.71 VIDW0ALPHA1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0220, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
ALPHA1_R_F	[23:16]	RW	Specifies Red Alpha value (case AEN == 1).	0
ALPHA1_G_F	[15:8]	RW	Specifies Green Alpha value (case AEN == 1).	0
ALPHA1_B_F	[7:0]	RW	Specifies Blue Alpha value (case AEN == 1).	0

41.5.3.72 VIDW1ALPHA0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0224, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
RSVD	[23:20]	–	Reserved	0
ALPHA0_R_L_F	[19:16]	RW	Specifies Red Alpha lower value (case AEN == 0).	0
RSVD	[15:12]	–	Reserved	0
ALPHA0_G_L_F	[11: 8]	RW	Specifies Green Alpha lower value (case AEN == 0).	0
RSVD	[7: 4]	–	Reserved	0
ALPHA0_B_L_F	[3: 0]	RW	Specifies Blue Alpha lower value (case AEN == 0).	0

NOTE: ALPHA0_R (G, B) [7:4] = ALPHA0_R (G, B)_H[3:0] at VIDOSD1C
ALPHA0_R (G, B) [3:0] = ALPHA0_R (G, B)_L[3:0] at VIDW1ALPHA0

41.5.3.73 VIDW1ALPHA1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0228, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
RSVD	[23:20]	–	Reserved	0
ALPHA1_R_L_F	[19:16]	RW	Specifies Red Alpha lower value (case AEN == 1).	0
RSVD	[15:12]	–	Reserved	0
ALPHA1_G_L_F	[11: 8]	RW	Specifies Green Alpha lower value (case AEN == 1).	0
RSVD	[7: 4]	–	Reserved	0
ALPHA1_B_L_F	[3: 0]	RW	Specifies Blue Alpha lower value (case AEN == 1).	0

NOTE: ALPHA1_R (G, B) [7:4] = ALPHA1_R (G, B)_H[3:0] at VIDOSD1C
ALPHA1_R (G, B) [3:0] = ALPHA1_R (G, B)_L[3:0] at VIDW1ALPHA1

41.5.3.74 VIDW2ALPHA0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x022C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
RSVD	[23:20]	–	Reserved	0
ALPHA0_R_L_F	[19:16]	RW	Specifies Red Alpha lower value (case AEN == 0).	0
RSVD	[15:12]	–	Reserved	0
ALPHA0_G_L_F	[11: 8]	RW	Specifies Green Alpha lower value (case AEN == 0).	0
RSVD	[7: 4]	–	Reserved	0
ALPHA0_B_L_F	[3: 0]	RW	Specifies Blue Alpha lower value (case AEN == 0).	0

NOTE: ALPHA0_R (G, B) [7:4] = ALPHA0_R (G, B)_H[3:0] atVIDOSD2C
ALPHA0_R (G, B) [3:0] = ALPHA0_R (G, B)_L[3:0] at VIDW2ALPHA0

41.5.3.75 VIDW2ALPHA1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0230, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
RSVD	[23:20]	–	Reserved	0
ALPHA1_R_L_F	[19:16]	RW	Specifies Red Alpha lower value (case AEN == 1).	0
RSVD	[15:12]	–	Reserved	0
ALPHA1_G_L_F	[11: 8]	RW	Specifies Green Alpha lower value (case AEN == 1).	0
RSVD	[7: 4]	–	Reserved	0
ALPHA1_B_L_F	[3: 0]	RW	Specifies Blue Alpha lower value (case AEN == 1).	0

NOTE: ALPHA1_R (G, B) [7:4] = ALPHA1_R (G, B)_H[3:0] at VIDOSD2C
ALPHA1_R (G, B) [3:0] = ALPHA1_R (G, B)_L[3:0] at VIDW2ALPHA1

41.5.3.76 VIDW3ALPHA0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0234, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
RSVD	[23:20]	–	Reserved	0
ALPHA0_R_L_F	[19:16]	RW	Specifies Red Alpha lower value (case AEN == 0).	0
RSVD	[15:12]	–	Reserved	0
ALPHA0_G_L_F	[11: 8]	RW	Specifies Green Alpha lower value (case AEN == 0).	0
RSVD	[7: 4]	–	Reserved	0
ALPHA0_B_L_F	[3: 0]	RW	Specifies Blue Alpha lower value (case AEN == 0).	0

NOTE: ALPHA0_R (G, B) [7:4] = ALPHA0_R (G, B)_H[3:0] at VIDOSD3C
ALPHA0_R (G, B) [3:0] = ALPHA0_R (G, B)_L[3:0] at VIDW3ALPHA0

41.5.3.77 VIDW3ALPHA1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0238, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
RSVD	[23:16]	–	Reserved	0
ALPHA1_R_L_F	[19:16]	RW	Specifies Red Alpha lower value (case AEN == 1).	0
RSVD	[15:12]	–	Reserved	0
ALPHA1_G_L_F	[11: 8]	RW	Specifies Green Alpha lower value (case AEN == 1).	0
RSVD	[7: 4]	–	Reserved	0
ALPHA1_B_L_F	[3: 0]	RW	Specifies Blue Alpha lower value (case AEN == 1).	0

NOTE: ALPHA1_R (G, B) [7:4] = ALPHA1_R (G, B)_H[3:0]@VIDOSD3C
ALPHA1_R (G, B) [3:0] = ALPHA1_R (G,B)_L[3:0]@VIDW3ALPHA1

41.5.3.78 VIDW4ALPHA0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x023C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
RSVD	[23:20]	–	Reserved	0
ALPHA0_R_L_F	[19:16]	RW	Specifies Red Alpha lower value (case AEN == 0).	0
RSVD	[15:12]	–	Reserved	0
ALPHA0_G_L_F	[11: 8]	RW	Specifies Green Alpha lower value (case AEN == 0).	0
RSVD	[7: 4]	–	Reserved	0
ALPHA0_B_L_F	[3: 0]	RW	Specifies Blue Alpha lower value (case AEN == 0).	0

NOTE: ALPHA0_R (G, B) [7:4] = ALPHA0_R (G, B)_H[3:0] at VIDOSD4C
ALPHA0_R (G, B) [3:0] = ALPHA0_R (G, B)_L[3:0] at VIDW4ALPHA0

41.5.3.79 VIDW4ALPHA1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0240, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[24]	–	Reserved	0
RSVD	[23:20]	–	Reserved	0
ALPHA1_R_L_F	[19:16]	RW	Specifies Red Alpha lower value (case AEN == 1).	0
RSVD	[15:12]	–	Reserved	0
ALPHA1_G_L_F	[11: 8]	RW	Specifies Green Alpha lower value (case AEN == 1).	0
RSVD	[7: 4]	–	Reserved	0
ALPHA1_B_L_F	[3: 0]	RW	Specifies Blue Alpha lower value (case AEN == 1).	0

NOTE: ALPHA1_R (G, B) [7:4] = ALPHA1_R (G, B)_H[3:0] at VIDOSD4C
ALPHA1_R (G, B) [3:0] = ALPHA1_R (G, B)_L[3:0] at VIDW4ALPHA1

41.5.3.80 BLENDEQ1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0244, Reset Value = 0x0000_00C2

Name	Bit	Type	Description	Reset Value
RSVD	[31:22]	–	Reserved	0x000
Q_FUNC_F	[21:18]	RW	Specifies constant that it uses in alphaB (alpha value of background (1)) 0000 = 0 (zero) 0001 = 1 (maximum) 0010 = alphaA (2) (alpha value of foreground (1)) 0011 = 1 – alphaA 0100 = alphaB 0101 = 1 – alphaB 0110 = ALPHA0 0111 = Reserved 100x = Reserved 1010 = A (foreground color data) 1011 = 1 – A 1100 = B (background color data) 1101 = 1 – B 111x = Reserved	0x0
RSVD	[17:16]	–	Reserved	00
P_FUNC_F	[15:12]	RW	Specifies the constant that it uses in alpha. Same as above (see COEF_Q).	0x0
RSVD	[11:10]	–	Reserved	00
B_FUNC_F	[9:6]	RW	Specifies the constant that it uses in B. Same as above (see COEF_Q).	0x3
RSVD	[5:4]	–	Reserved	00
A_FUNC_F	[3:0]	RW	Specifies the constant that it uses in A. Same as above (see COEF_Q).	0x2

NOTE: For more information, refer to *Figure 41-23*, "Blending equation".

1. Background = Window 0, foreground = Window 1 (in Blend Equation 1)
2. BPPMODE_F, BLD_PIX, ALPHA_SEL at WINCONx, and WxPAL at WPALCON decides the alphaA and alphaB.

41.5.3.81 BLENDEQ2

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0248, Reset Value = 0x0000_00C2

Name	Bit	Type	Description	Reset Value
RSVD	[31:22]	–	Reserved	0x000
Q_FUNC_F	[21:18]	RW	Specifies constant that it uses in alphaB (alpha value of background (1)). 0000 = 0 (zero) 0001 = 1 (maximum) 0010 = alphaA (2) (alpha value of foreground (1)) 0011 = 1 – alphaA 0100 = alphaB 0101 = 1 – alphaB 0110 = ALPHA0 0111 = Reserved 100x = Reserved 1010 = A (foreground color data) 1011 = 1 – A 1100 = B (background color data) 1101 = 1 – B 111x = Reserved	0x0
RSVD	[17:16]	–	Reserved	00
P_FUNC_F	[15:12]	RW	Specifies constant that it uses in alpha. Same as above (see COEF_Q)	0x0
RSVD	[11:10]	–	Reserved	00
B_FUNC_F	[9:6]	RW	Specifies constant that it uses in B. Same as above (see COEF_Q)	0x3
RSVD	[5:4]	–	Reserved	00
A_FUNC_F	[3:0]	RW	Specifies constant that it uses in A. Same as above (see COEF_Q)	0x2

NOTE: For more information, Refer to *Figure 41-23*, "Blending equation".

1. Background = Window 01, foreground = Window 2 (in Blend Equation 2)
2. BPPMODE_F, BLD_PIX, ALPHA_SEL at WINCONx, and WxPAL at WPALCON decides the alphaA and alphaB.

41.5.3.82 BLENDEQ3

- Base Address = 0x11C0_0000
- Address = Base Address + 0x024C, Reset Value = 0x0000_00C2

Name	Bit	Type	Description	Reset Value
RSVD	[31:22]	–	Reserved	0x000
Q_FUNC_F	[21:18]	RW	Specifies constant that it uses in alphaB (alpha value of background (1)) 0000 = 0 (zero) 0001 = 1 (maximum) 0010 = alphaA (2) (alpha value of foreground (1)) 0011 = 1 – alphaA 0100 = alphaB 0101 = 1 – alphaB 0110 = ALPHA0 0111 = Reserved 100x = Reserved 1010 = A (foreground color data) 1011 = 1 – A 1100 = B (background color data) 1101 = 1 – B 111x = Reserved	0x0
RSVD	[17:16]	–	Reserved	00
P_FUNC_F	[15:12]	RW	Specifies constant that it uses in alpha. Same as above (see COEF_Q).	0x0
RSVD	[11:10]	–	Reserved	00
B_FUNC_F	[9:6]	RW	Specifies constant that it uses in B. Same as above (see COEF_Q).	0x3
RSVD	[5:4]	–	Reserved	00
A_FUNC_F	[3:0]	RW	Specifies constant that it uses in A. Same as above (see COEF_Q).	0x2

NOTE: For more information, Refer to *Figure 41-23*, "Blending equation".

1. Background = Window 012, foreground = Window 3 (in Blend Equation 3)
2. BPPMODE_F, BLD_PIX, ALPHA_SEL @WINCONx, and WxPAL @WPALCON decides the alphaA and alphaB.

41.5.3.83 BLENDEQ4

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0250, Reset Value = 0x0000_00C2

Name	Bit	Type	Description	Reset Value
RSVD	[31:22]	–	Reserved	0x000
Q_FUNC_F	[21:18]	RW	Specifies constant that it uses in alphaB (alpha value of background (1)) 0000 = 0 (zero) 0001 = 1 (maximum) 0010 = alphaA (2) (alpha value of foreground (1)) 0011 = 1 – alphaA 0100 = alphaB 0101 = 1 – alphaB 0110 = ALPHA0 0111 = Reserved 100x = Reserved 1010 = A (foreground color data) 1011 = 1 – A 1100 = B (background color data) 1101 = 1 – B 111x = Reserved	0x0
RSVD	[17:16]	–	Reserved	00
P_FUNC_F	[15:12]	RW	Specifies constant that it uses in alpha. Same as above (see COEF_Q).	0x0
RSVD	[11:10]	–	Reserved	00
B_FUNC_F	[9:6]	RW	Specifies constant that it uses in B. Same as above (see COEF_Q).	0x3
RSVD	[5:4]	–	Reserved	00
A_FUNC_F	[3:0]	RW	Specifies constant that it uses in A. Same as above (see COEF_Q).	0x2

NOTE: For more information, Refer to *Figure 41-23*, "Blending equation".

1. Background = Window 0123, foreground = Window 4 (in Blend Equation 4)
2. BPPMODE_F, BLD_PIX, ALPHA_SEL @WINCONx, and WxPAL @WPALCON decides the alphaA and alphaB. .

41.5.3.84 BLENDCON

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0260, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x000
BLEND_NEW	[0]	RW	Specifies alpha value width. 0 = 4-bit alpha value 1 = 8-bit alpha value	0x0

41.5.3.85 WnRTQOSCON (n = 0 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0264, + 0x0268, + 0x026C, + 0x0270, + 0x0274, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved NOTE: This bit should be set to 0.	0
FIFOLEVEL	[11:4]	RW	Specifies real-time QoS FIFO level. If FIFO depth is less than FIFOLEVEL[7:0], then RTQoS output is 1.	0
RSVD	[3:2]	–	Reserved NOTE: This bit should be set to 0.	0
QOS_GATE_DIS	[1]	RW	Disables RTQoS output signal gate. 0 = Gates 1 = Does not gate	0
RSVD	[0]	–	Reserved NOTE: This bit should be set to 0.	0

41.5.3.86 LDI_CMDn (n = 0 to 11)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0280, + 0x0284, + 0x0288, + 0x028C, + 0x0290, + 0x0294, + 0x0298, + 0x029C, + 0x02A0, + 0x02A4, + 0x02A8, + 0x02AC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
LDI_CMD	[23:0]	RW	Specifies LDI command.	0

41.5.4 Gamma Lookup Table

41.5.4.1 Gamma LUT Data for 64 Step Mode

- Base Address = 0x11C0_0000

Name	Bit	Type	Description	Reset Value
GM_LUT_x	[26:18]	RW	Specifies Gamma LUT value register of index x.	Undefined
GM_LUT_y	[10: 2]	RW	Specifies Gamma LUT value register of index y.	Undefined

41.5.4.2 Gamma LUT Data for 16 Step Mode

- Base Address = 0x11C0_0000

Name	Bit	Type	Description	Reset Value
GM_LUT_x	[26:18]	RW	Specifies Gamma LUT value register of index x.	Undefined
GM_LUT_y	[10: 2]	RW	Specifies Gamma LUT value register of index y.	Undefined

41.5.5 Shadow Windows Control

41.5.5.1 SHD_VIDW0nADD0 (n = 0 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x40A0, + 0x40A8, + 0x40B0, + 0x40B8, + 0x40C0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
VBASEU_F	[31:0]	R	Specifies A[31:0] of the start address for video frame buffer (shadow).	0

41.5.5.2 SHD_VIDW0nADD1 (n = 0 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x40D0, + 0x40D8, + 0x40E0, + 0x40E8, + 0x40F0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
VBASEL_F	[31:0]	R	Specifies A[31:0] of the end address for video buffer (shadow).	0x0

41.5.5.3 SHD_VIDW0nADD2 (n = 0 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x4100, + 0x4104, + 0x4108, + 0x410C, + 0x4110, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
OFFSIZE_F	[25:13]	R	Specifies virtual screen offset size that is the number of byte (shadow).	0
PAGEWIDTH_F	[12:0]	R	Specifies virtual screen page width (number of byte). This value defines the width of view port in the frame (shadow).	0

41.5.6 Palette Ram

41.5.6.1 Win0 Palette Ram Access Address (not SFR)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x2400, 0x0400, Reset Value = 0x0000_0000
- Address = Base Address + 0x2404, 0x0404, Reset Value = 0x0000_0000
- Address = Base Address + 0x27FC, 0x07FC, Reset Value = 0x0000_0000

Register	Address	Type	Description	Reset Value
00	0x0_2400 (0x0_0400)	RW	Specifies Window 0 Palette entry 0 address.	Undefined
01	0x0_2404 (0x0_0404)	RW	Specifies Window 0 Palette entry 1 address.	Undefined
–	–	–	–	–
FF	0x0_27FC (0x0_07FC)	RW	Specifies Window 0 Palette entry 255 address.	Undefined

41.5.6.2 Win1 Palette Ram Access Address (not SFR)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x2800,0x0800, Reset Value = 0x0000_0000
- Address = Base Address + 0x2804,0x0804, Reset Value = 0x0000_0000
- Address = Base Address + 0x2BFC,0x0BFC, Reset Value = 0x0000_0000

Register	Address	Type	Description	Reset Value
00	0x0_2800 (0x0_0800)	RW	Specifies Window 1 Palette entry 0 address.	Undefined
01	0x0_2804 (0x0_0804)	RW	Specifies Window 1 Palette entry 1 address.	Undefined
–	–	–	–	–
FF	0x0_2BFC (0x0_0BFC)	RW	Specifies Window 1 Palette entry 255 address.	Undefined

41.5.6.3 Win2 Palette Ram Access Address (not SFR)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x2C00,0x0C00, Reset Value = 0x0000_0000
- Address = Base Address + 0x2C04,0x0C04, Reset Value = 0x0000_0000
- Address = Base Address + 0x2FFC,0x0FFC, Reset Value = 0x0000_0000

Register	Address	Type	Description	Reset Value
00	0x0_2C00	RW	Specifies Window 2 Palette entry 0 address.	Undefined
01	0x0_2C04	RW	Specifies Window 2 Palette entry 1 address.	Undefined
–	–	–	–	–
FF	0x0_2FFC	RW	Specifies Window 2 Palette entry 255 address.	Undefined

41.5.6.4 Win3 Palette Ram Access Address (not SFR)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x3000, Reset Value = 0x0000_0000
- Address = Base Address + 0x3004, Reset Value = 0x0000_0000
- Address = Base Address + 0x33FC, Reset Value = 0x0000_0000

Register	Address	Type	Description	Reset Value
00	0x0_3000	RW	Specifies Window 3 Palette entry 0 address.	Undefined
01	0x0_3004	RW	Specifies Window 3 Palette entry 1 address.	Undefined
–	–	–	–	–
FF	0x0_33FC	RW	Specifies the Window 3 Palette entry 255 address.	Undefined

41.5.6.5 Win4 Palette Ram Access Address (not SFR)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x3400, Reset Value = 0x0000_0000
- Address = Base Address + 0x3404, Reset Value = 0x0000_0000
- Address = Base Address + 0x37FC, Reset Value = 0x0000_0000

Register	Address	Type	Description	Reset Value
00	0x0_3400	RW	Specifies Window 4 Palette entry 0 address.	Undefined
01	0x0_3404	RW	Specifies Window 4 Palette entry 1 address.	Undefined
–	–	–	–	–
FF	0x0_37FC	R/W	Specifies Window 4 Palette entry 255 address.	Undefined

42 Camera Interface and Scaler

42.1 Overview

The Camera Interface (CAMIF) in Exynos 4412 SCP is a fully interactive mobile camera interface. CAMIF supports ITU -R BT-601/656 standard, AXI-bus interface, and MIPI (CSI).

The maximum input image size of CAMIF is 8192 × 8192 pixels.

The four CAMIF units of Exynos 4412 SCP are:

- CAMIF0
- CAMIF1
- CAMIF2
- CAMIF3

These units are designed to perform different functions. *Figure 42-2* illustrates the functions of these units.

The functions of CAMIF units are:

- T_PatternMux generates test pattern to calibrate input sync signals as HREF and VSYNC.
- Capture specifies the capturing signal and window cut. Use the register settings to invert video sync signals and pixel clock polarity in CAMIF.
- Scaler generates various sizes of image.
- Input Direct Memory Access (DMA) (read only) reads image data from the memory.
- Output DMA (write only) writes image data to memory.
- CAMIF supports image rotation (90 degrees clockwise) and image effect functions.

The key application of CAMIR units is in a folder-type cellular phone.

Figure 42-1 illustrates the subset of visual system in Exynos 4412 SCP.

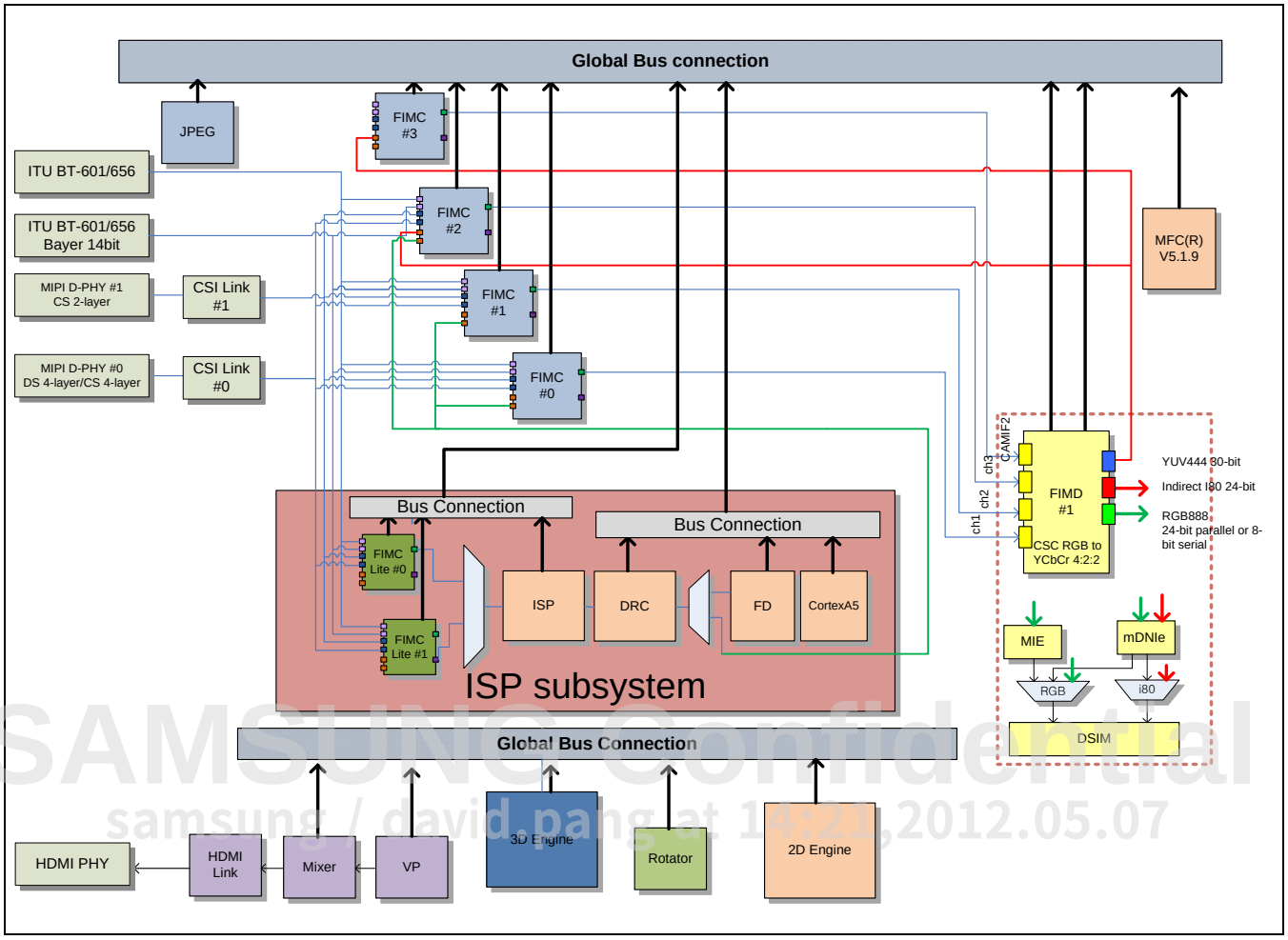


Figure 42-1 Subset of Visual System in Exynos 4412 SCP

42.2 Features of CAMIF

The features of CAMIF are:

- Supports multiple inputs. The inputs are:
  - ITU-R BT 601/656/709 mode
  - DMA (AXI 64-bit interface) mode
  - MIPI (CSI) mode
  - Direct FIFO (PlayBack) mode
- Supports multiple outputs. The outputs are:
  - DMA (AXI 64-bit interface) mode
  - Direct FIFO mode
- Digital Zoom In (DZI) capability
- Multiple camera input
- Video sync signals have programmable polarity
- Supports maximum 8192 × 8192 pixels input (Refer to *Table 42-1*)
- Image mirror and rotation (X-axis mirror, Y-axis mirror, 90°, 180°, and 270° rotation)
- Generates various image formats
- Supports capture frame control
- Supports image effect

Table 42-1 describes the maximum size.

Table 42-1 Maximum Size

–	Item	Maximum Size	
		CAMIF 0, 1, 2	CAMIF3
Scaler	Scaler input Horizontal size (= PreDestinationWidth)	4224 pixels	1920 pixels
	Scaler bypass mode	8192 pixels	8192 pixels
Output Rotator	TargetHsize (without output rotation)	4224 pixels	1920 pixels
	TargetHsize (with output rotation)	1920 pixels	1366 pixels
Input Rotator	REAL_WIDTH (without input rotation)	8192 pixels	8192 pixels
	REAL_HEIGHT (with input rotation)	1920 pixels	1366 pixels

NOTE:

1. The maximum size of scaler and rotator depends on the buffer line size.
   The maximum size of the output rotator and input rotator is less than the maximum size of scaler.
2. Minimum input size = 32 × 32
3. Minimum output size = 32 × 32 (normal), 128 × 128 (output rotation and interlace out are enable)

Figure 42-2 illustrates the camera interface overview.

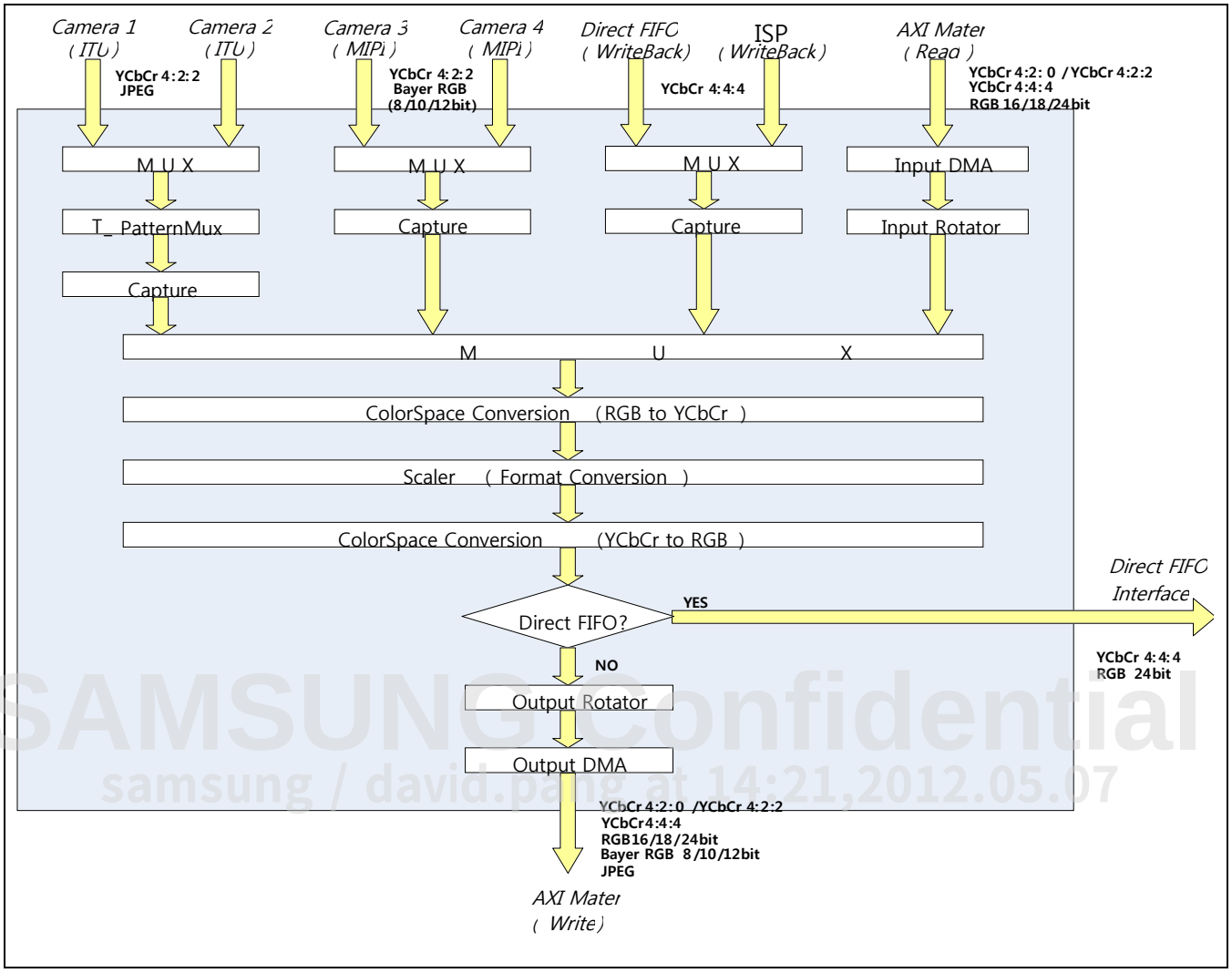


Figure 42-2 Camera Interface Overview

NOTE: In Direct FIFO (WriteBack) input mode, CAMIF does not support cropping, capture frame control, test pattern, and scaler bypass function.

42.3 External Interface

CAMIF supports three video standards. The three video standards are:

- ITU-R BT 601 YCbCr 8-bit mode
- ITU-R BT 656 YCbCr 8-bit mode
- MIPI mode

42.4 Timing Diagram and Data Alignment of Camera

Timing Diagram and Data Alignment of Camera section includes:

- Timing Diagram of ITU Camera
- MIPI CSI Data Alignment from MIPI Camera

42.4.1 Timing Diagram of ITU Camera

Figure 42-3 illustrates the ITU-R BT 601 input timing diagram.

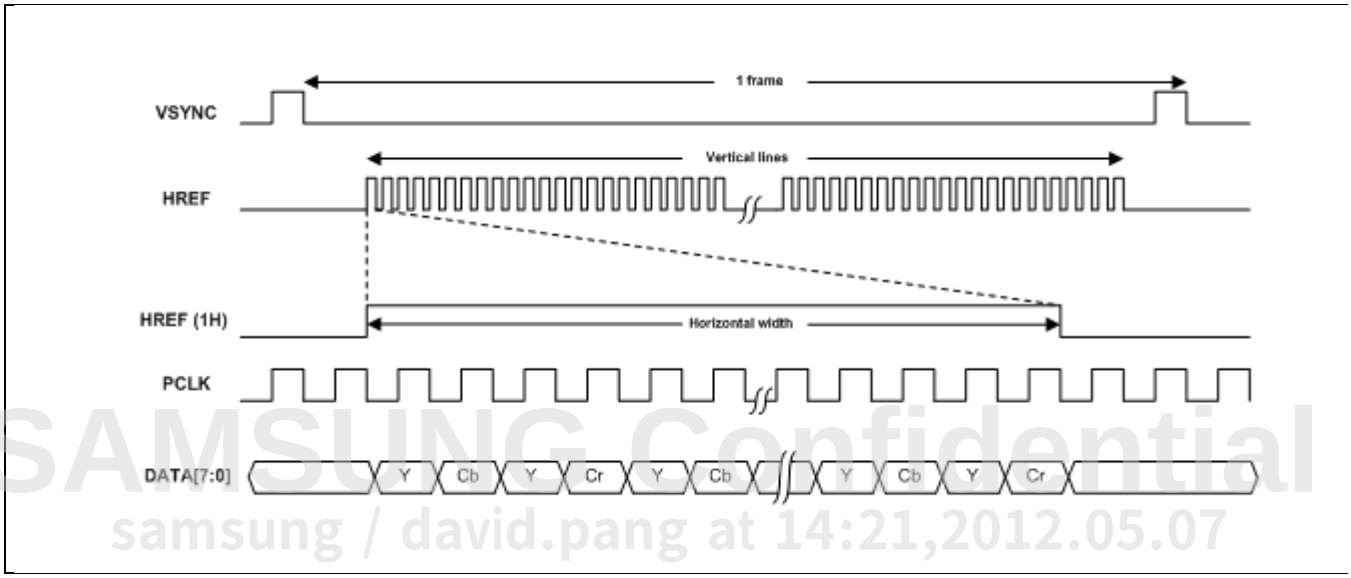


Figure 42-3 ITU-R BT 601 Input Timing Diagram

Figure 42-4 illustrates the ITU-R BT 601 interface handling diagram.

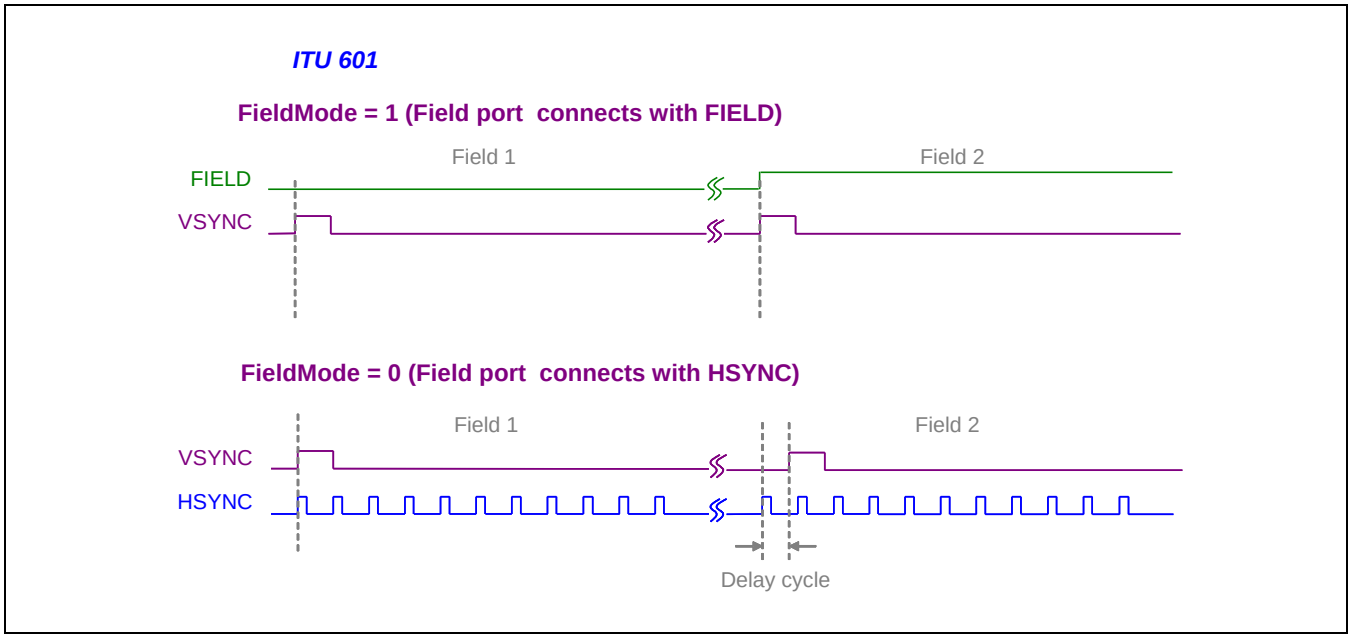


Figure 42-4 ITU-R BT 601 Interlace Handling Diagram

Figure 42-5 illustrates the -R BT 656 input timing diagram.

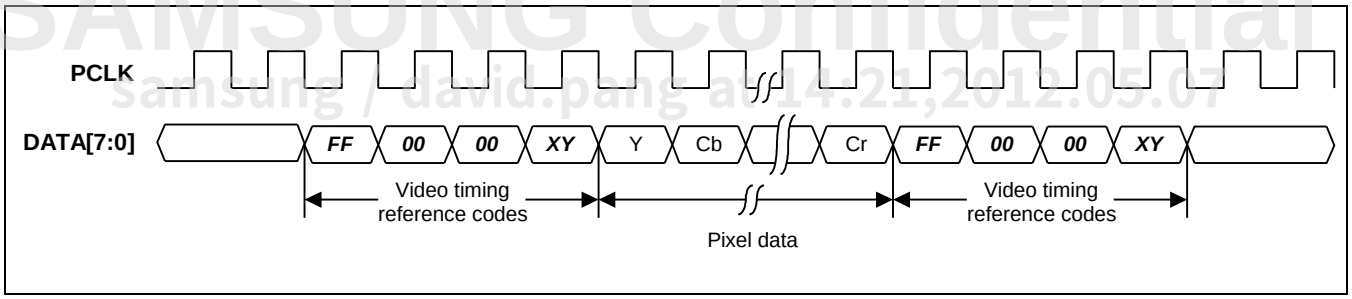


Figure 42-5 ITU-R BT 656 Input Timing Diagram

There are two timing reference signals in ITU-R BT 656 format:

- One at the beginning of each video data block (start of active video, SAV)
- Other at the end of each video data block (end of active video, EAV) as illustrated in *Figure 42-5* and described in *Table 42-2*.
- *Table 42-2* describes the video timing reference codes of ITU0656 8-bit format.

Table 42-2 Video Timing Reference Codes of ITU-656 8-bit Format

Data Bit Number	First Word	Second Word	Third Word	Fourth Word
7 (MSB)	1	0	0	1
6	1	0	0	F
5	1	0	0	V
4	1	0	0	H
3	1	0	0	P3
2	1	0	0	P2
1	1	0	0	P1
0	1	0	0	P0

NOTE: F = 0 (during field 1), 1 (during field 2)
V = 0 (elsewhere), 1 (during field blanking)
H = 0 (in SAV= Start of Active Video), 1 (in EAV= End of Active Video)
P0, P1, P2, P3 = Protection Bit
The camera interface logic catches video sync bits like H (SAV, EAV) and V (Frame Sync) after reserving data as "FF-00-00".

Figure 42-6 illustrates the sync signal timing diagram.

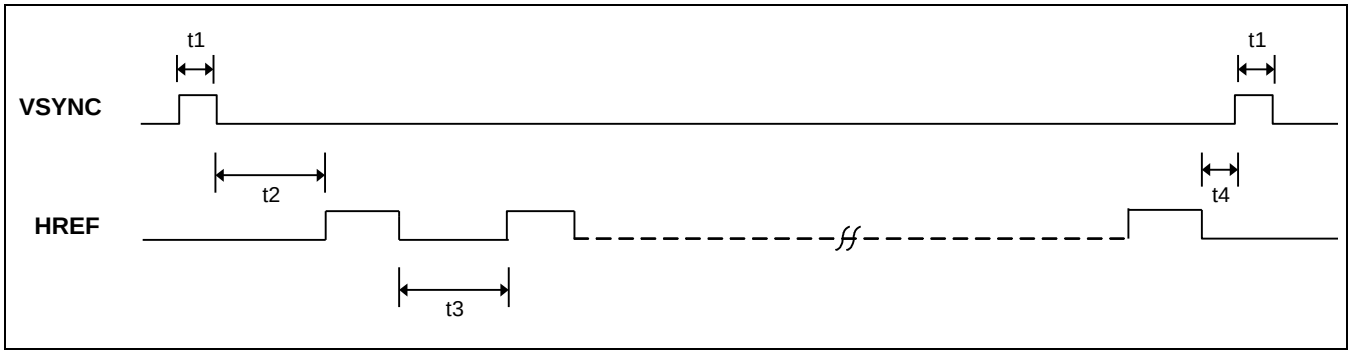


Figure 42-6 Sync Signal Timing Diagram

Table 42-3 describes the sync signal timing requirements.

Table 42-3 Sync Signal Timing Requirements

–	Minimum	Maximum
t1	2 Horizontal Line	–
t2	2 Cycles of Pixel Clock + 5 Cycles of System Bus Clock	–
t3	2 Cycles of Pixel Clock	–
t4	12 Cycles of Pixel Clock	–

NOTE: . If you enable the rotator, then (t4 + t1) should be sufficient to finish DMA transactions. Otherwise DMA transactions for rotator line buffer delays by four or eight horizontal lines.

Figure 42-7 illustrates the JPEG input timing diagram (ITU 601 and free run clock mode).

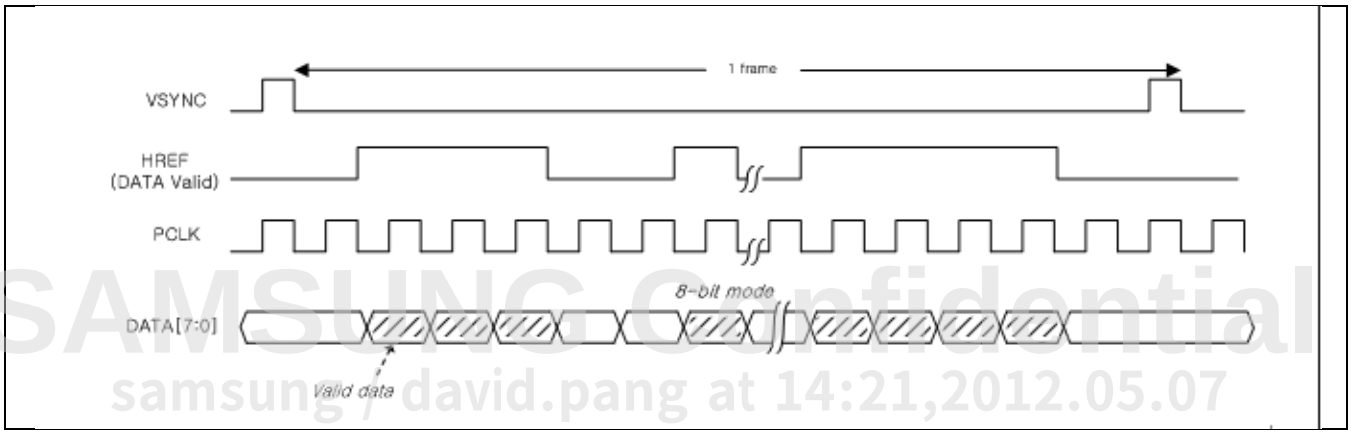


Figure 42-7 JPEG Input Timing Diagram (ITU 601 and Free run Clock Mode)

42.4.2 MIPI CSI Data Alignment from MIPI Camera

Figure 42-8 illustrates the MIPI CSI data alignment.

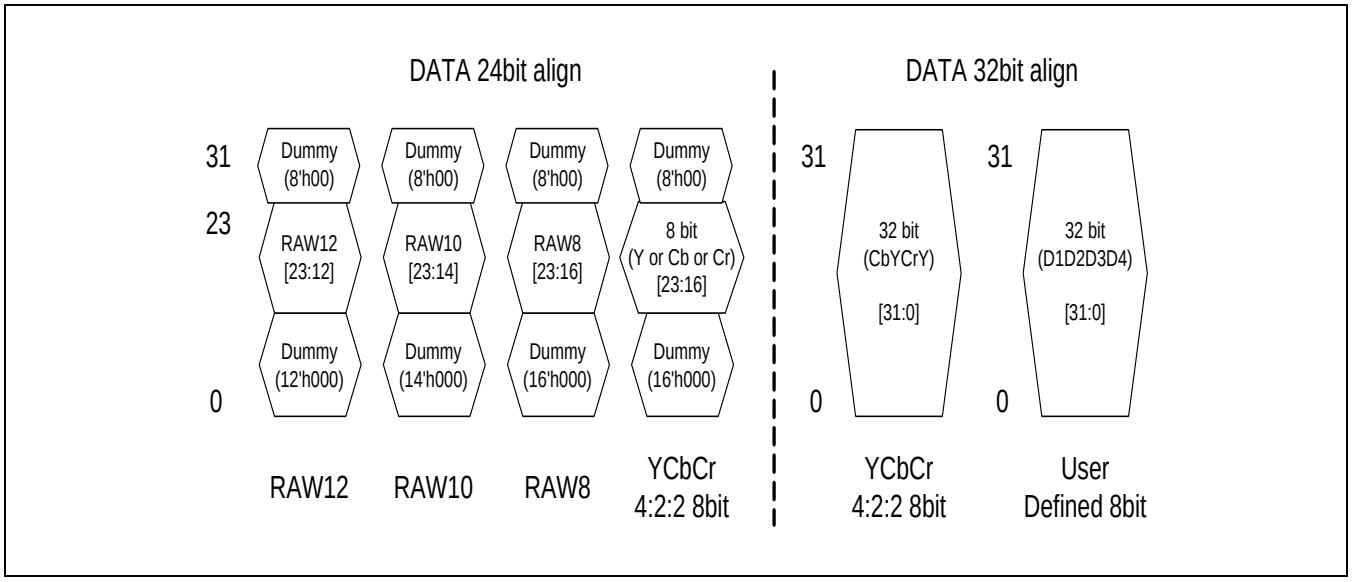


Figure 42-8 MIPI CSI DATA Alignment

Table 42-4 describes the data order of YCbCr 422 align.

Table 42-4 DATA Order of YCbCr422 Align

Format	Stream Order of Content	DATA: 24-bit Align	DATA: 32-bit Align
YCbCr422	Cb1 → Y1 → Cr1 → Y2 → ...	DATA1[23:16] = Cb1 DATA2[23:16] = Y1 DATA3[23:16] = Cr1 DATA4[23:16] = Y2	DATA1[31:24] = Cb1 DATA1[23:16] = Y1 DATA1[15:8] = Cr1 DATA1[7:0] = Y2
User defined	D1 → D2 → D3 → D4 → ...	N/A	DATA1[31:24] = D4[7:0] DATA1[23:16] = D3[7:0] DATA1[15:8] = D2[7:0] DATA1[7:0] = D1[7:0]

42.5 External Connection Guide

You should use next pin location and routing to prevent CAMIF input signals to result in inter-skewing of the pixel clock line.

Figure 42-9 illustrates the IO connection guide.

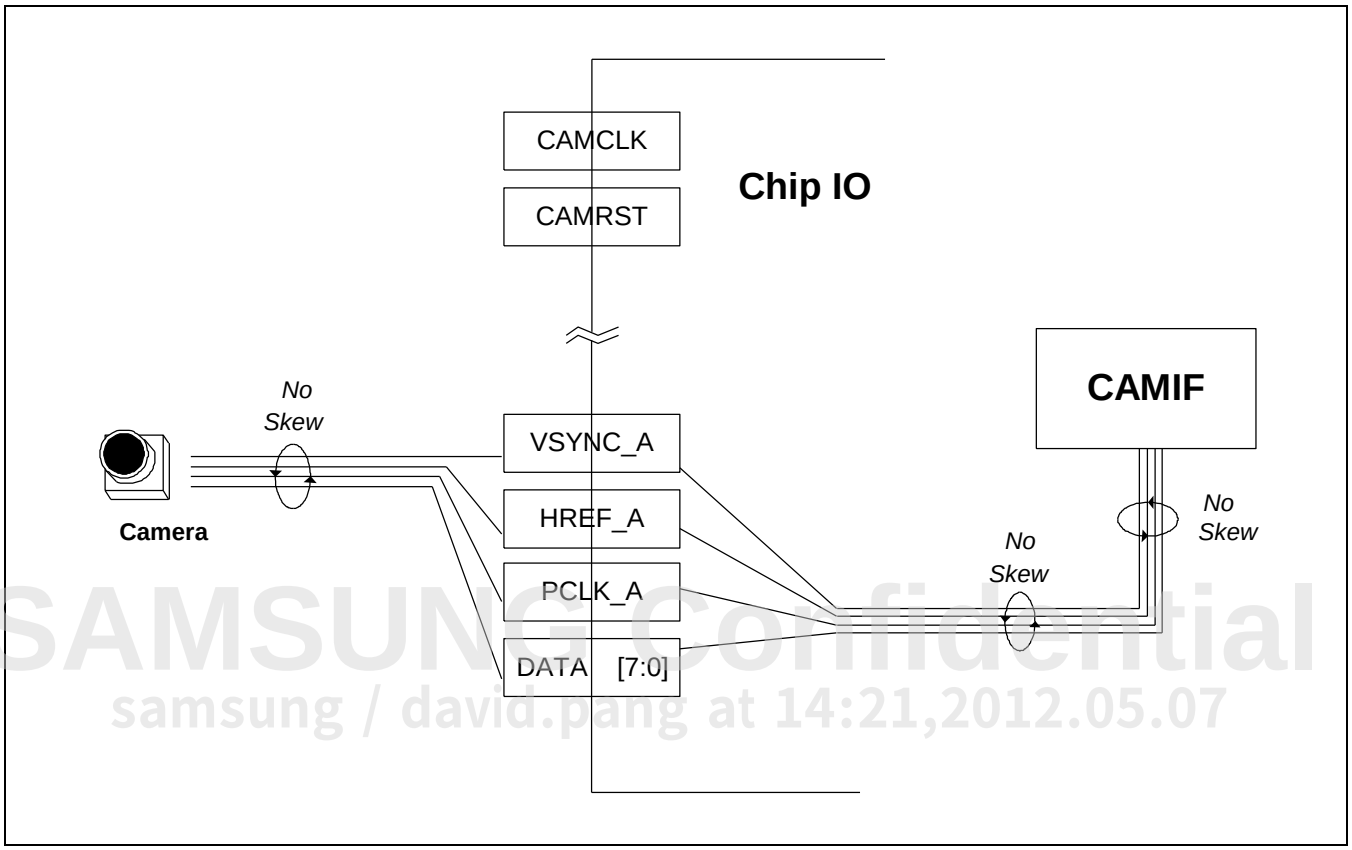


Figure 42-9 IO Connection Guide

42.6 Camera Interface Operation

The camera interface operation describes:

- Input/Output DMA Ports
- Clock Domain

42.6.1 Input/Output DMA Ports

The two DMA ports of CAMIF are:

- Input DMA Port
- Output DMA Port.

These two ports are independent from the system bus view point. The Input DMA port reads the image data from memory. On the other hand, the Output DMA port stores the image data into memory. These two master ports support various digital applications. These are:

- Digital Still Camera (DSC)
- MPEG-4 Video Conference
- Video Recording and so on.

Figure 42-10 illustrates the Input/Output DMA ports.

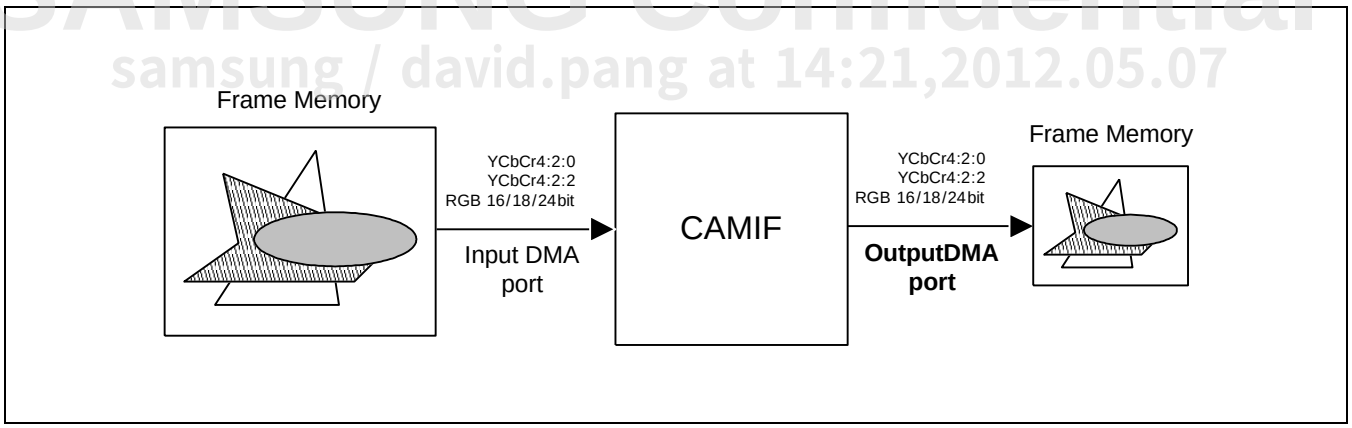


Figure 42-10 Input/Output DMA Ports

42.6.2 Clock Domain

The three clock domains of CAMIF are:

- System bus clock
- Camera pixel clock, PCLK
- Internal core clock

The system bus clock is faster than camera pixel clock.

Separates the CAM CLK from fixed frequency like PLL clock. If you use external clock oscillator, then it should float the CAM_MCLK. It is not necessary for the three clock domains to synchronize. You should connect other signals like PCLK similar to Schmitt triggered level shifter.

Figure 42-11 illustrates the CAMIF clock generation.

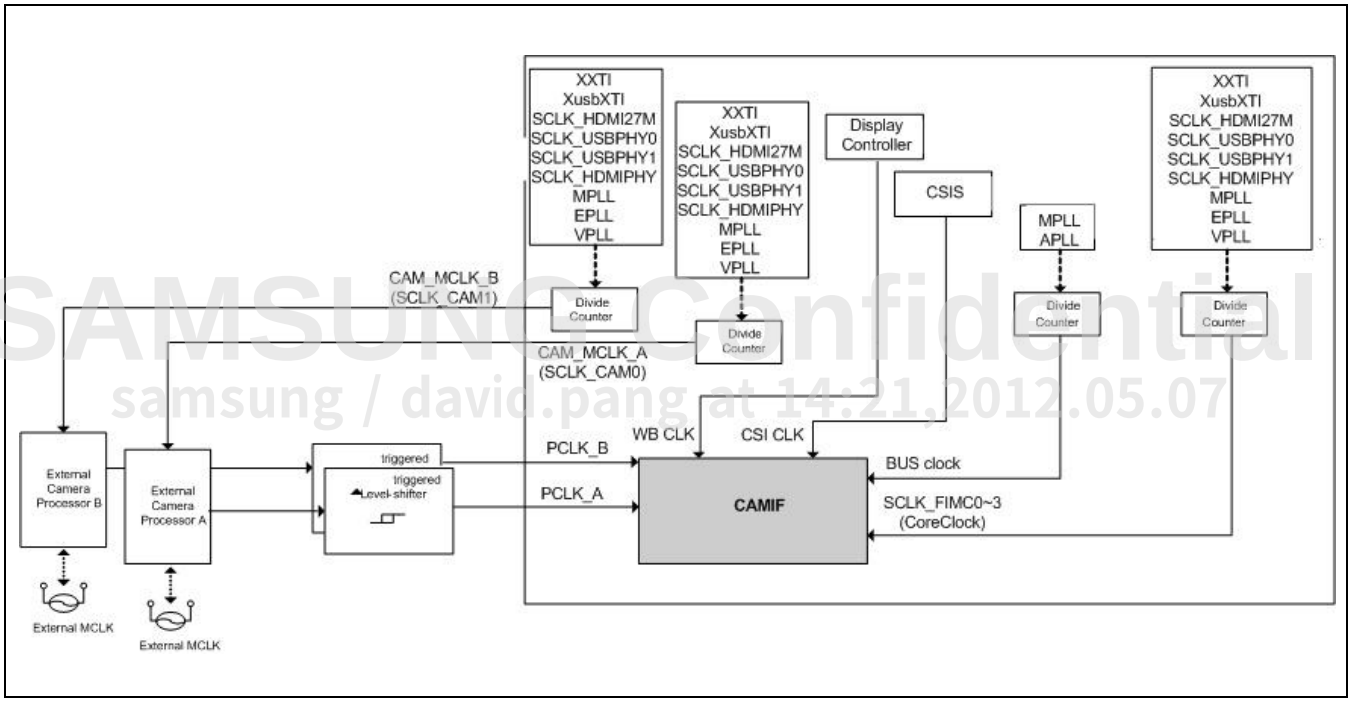


Figure 42-11 CAMIF Clock Generation

NOTE: The maximum frequency of CAM_MCLK_A, CAM_MCLK_B, PCLK_A and PCLK_B is 100 MHz.

42.6.3 Frame Memory Hierarchy

Frame memory consists of 32 ping-pong memories for output DMA ports. Ping-pong memory also consists of three element memories. The three memories are:

- Luminance Y
- Chrominance Cb
- Chrominance Cr

The arbitration priority of CAMIF should be higher than any other masters (except LCD controller). It is highly recommended to set the CAMIF priorities as fixed priorities, not rotation priorities.

The priority of system bus (including CAMIF) should be higher than others in case of multi AHB bus. When a DMA operation cannot complete for one horizontal period plus blank due to heavy bus traffic it results in mal-functioning. Therefore you should change the CAMIF priority to round robin or circular arbitration priorities. The bus that includes CAMIF is recommended to have higher priority than other buses in the memory matrix system. The CAMIF should not be the default master of AMBA system.

Figure 42-12 illustrates the ping-pong memory hierarchy.

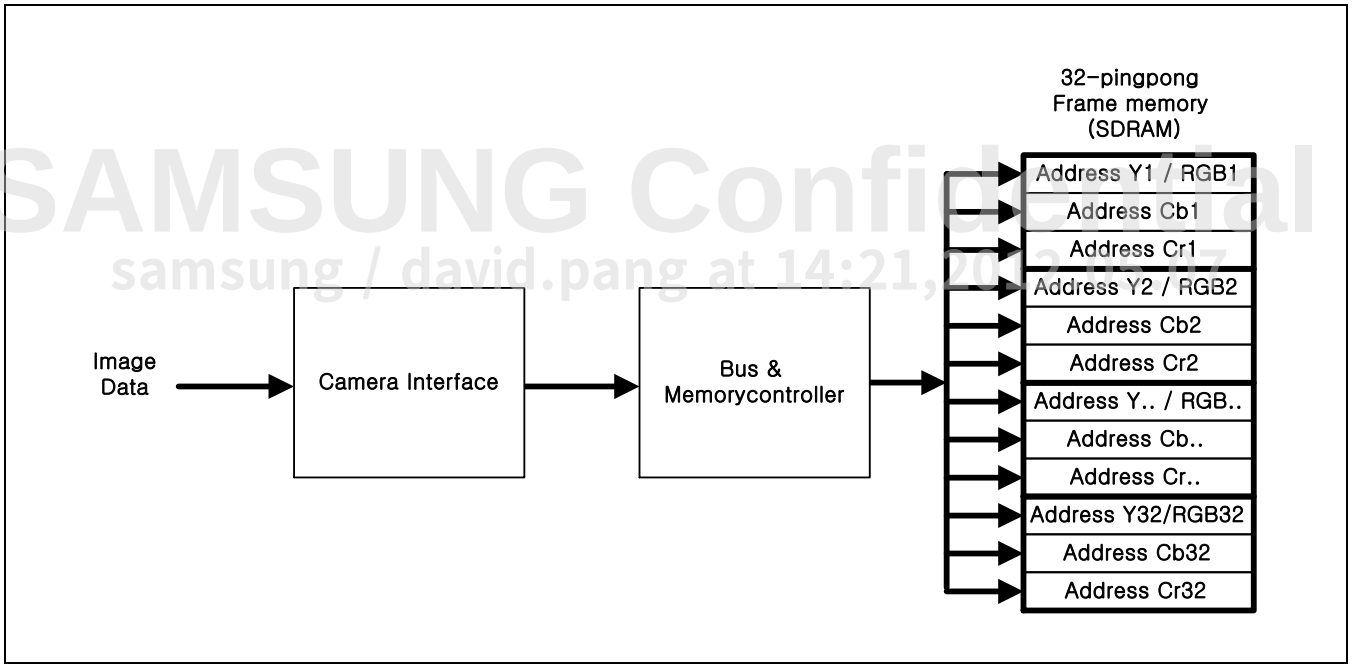


Figure 42-12 Ping-pong Memory Hierarchy

42.6.4 Memory Storing Method

Storing order of frame memory is little-endian. It stores the first entering pixels in LSB side and the last entering pixels in MSB side.

CAMIF frames Y-Cb-Cr words using little endian style. The AXI bus data width is 64-bit. CAMIF should pad with zero for RAW8, RAW10 and RAW12 format if end-of-horizontal line is not align with 64-bit. For more information, refer to *Figure 42-13*.

Figure 42-13 illustrates the memory storing style.

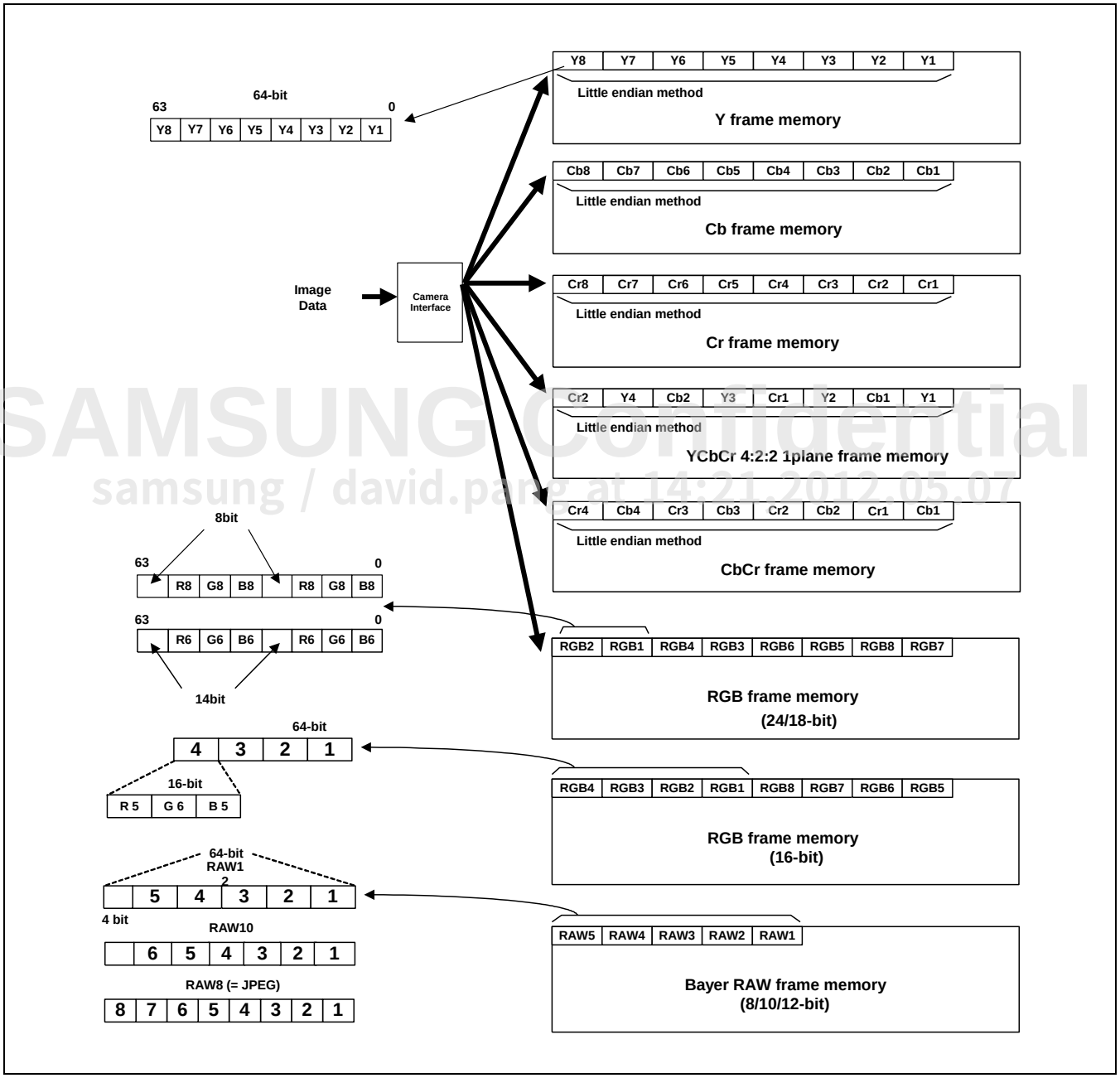


Figure 42-13 Memory Storing Style

42.6.5 Timing Diagram for Register Setting

For the first time, you can set the register for frame capture command at anytime in the frame period. It recommends to first set VSYNC "L" state, input DMA start "L" state and then VVALID 'H' state. You can read VSYNC and VVALID information form Special Function Register (SFR) status. For more information, refer to *Figure 42-14*.

All commands (including ImgCptEn) are valid at VSYNC falling edge or VVALID rising edge. You should program all commands in Interrupt Service Routine (ISR) except first SFR setting. During capture you are allowed to change size, image, mirror or rotation, windowing, and zoom in settings. In case of DMA input mode, all command are programmed after the Input DMA and Output DMA operation ends, as shown in *Figure 42-15*.

Figure 42-14 illustrates the timing diagram for camera input register setting.

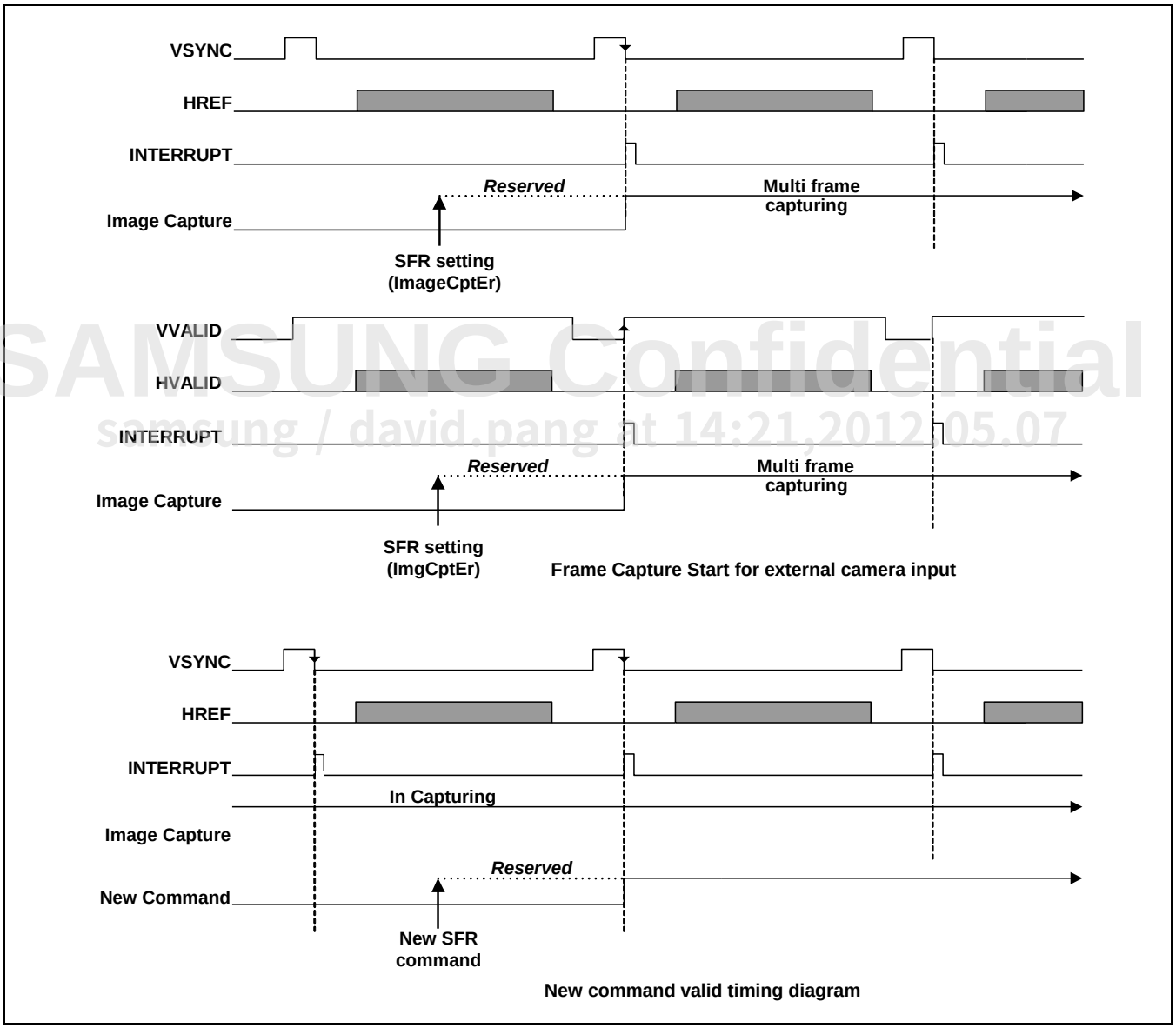


Figure 42-14 Timing Diagram for Camera Input Register Setting

Figure 42-15 illustrates the timing diagram for DMA input register setting.

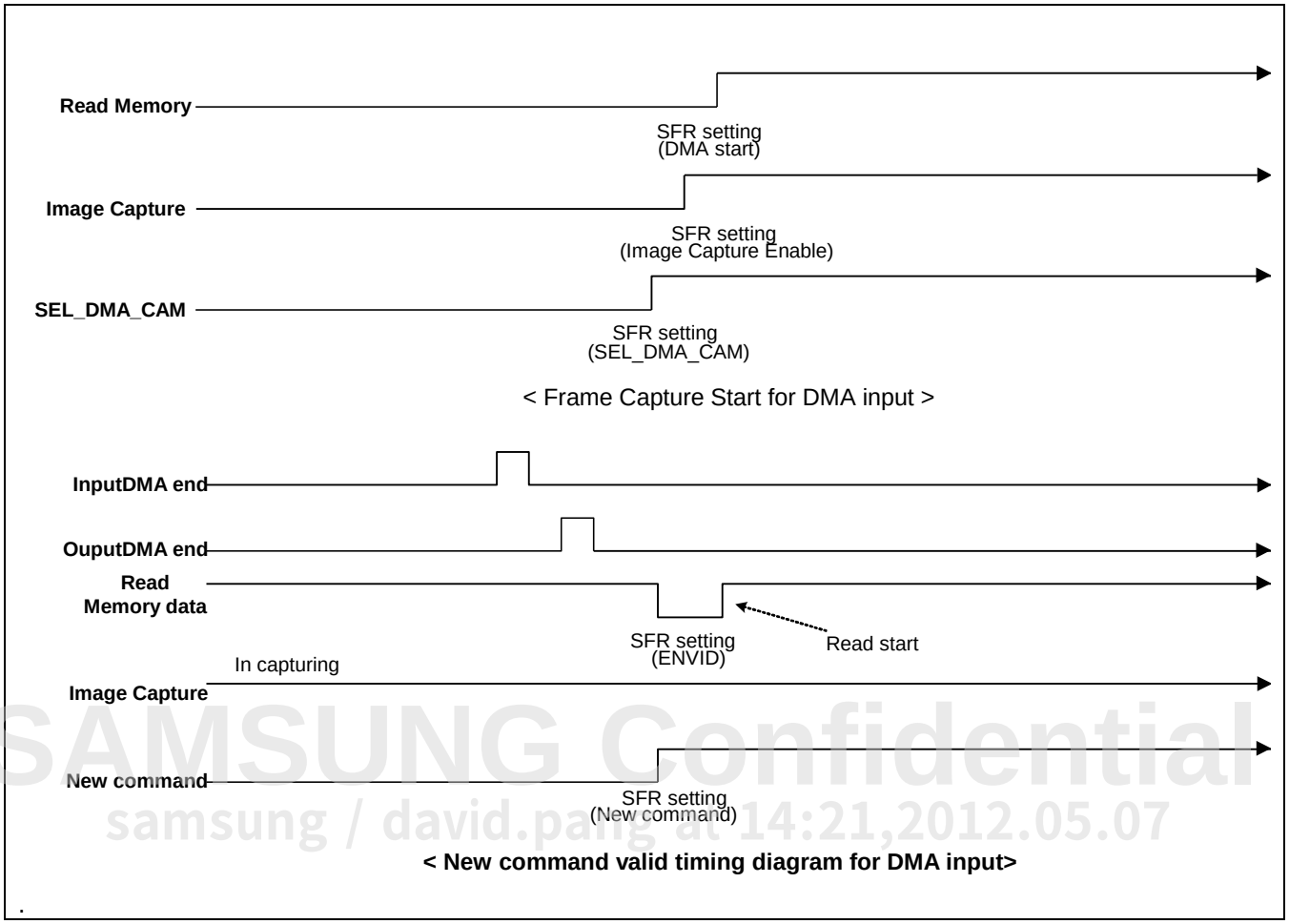


Figure 42-15 Timing Diagram for DMA input Register Setting

42.6.6 Timing Diagram for last IRQ

CAMIF generates IRQ (except Last IRQ) before Image capture Last IRQ specifies the end of camera signal capture. You can set IRQ according to the timing diagram shown in Figure 16.Last IRQ En specifies the ISR setting for next frame command. You should follow the next sequence between Last IRQEn and ImgCptEn/ ImgCptEn_Sc for proper last IRQ.

You are recommended to set ImgCptEn/ ImgCptEn_SC at the same time, at the end of SFR setting in ISR. You can read FrameCnt that specifies next frame count in ISR.

As shown in *Figure 42-17*, the last captured frame count is "1", that is, Frame 1 specifies the last captured frame is among frame 0 to 3. It increases FrameCnt by 1 at IRQ rising. You can select DMA input by setting SFR. In this case, CAMIF generates IRQ after completing the output DMA operation per frame The SFR setting (ENVID_M "0" → "1") makes this mode aware of the starting point. Therefore, this mode does not require IRQ of starting point and LastIRQ. "FrameCnt will increase by 1 at ENVID_M (InputDMA start) low to rising ("0" → "1") and ImgCptEn_SC "1".

Figure 42-16 illustrates the timing diagram for Last IRQ (LastIRQEn is enabled).

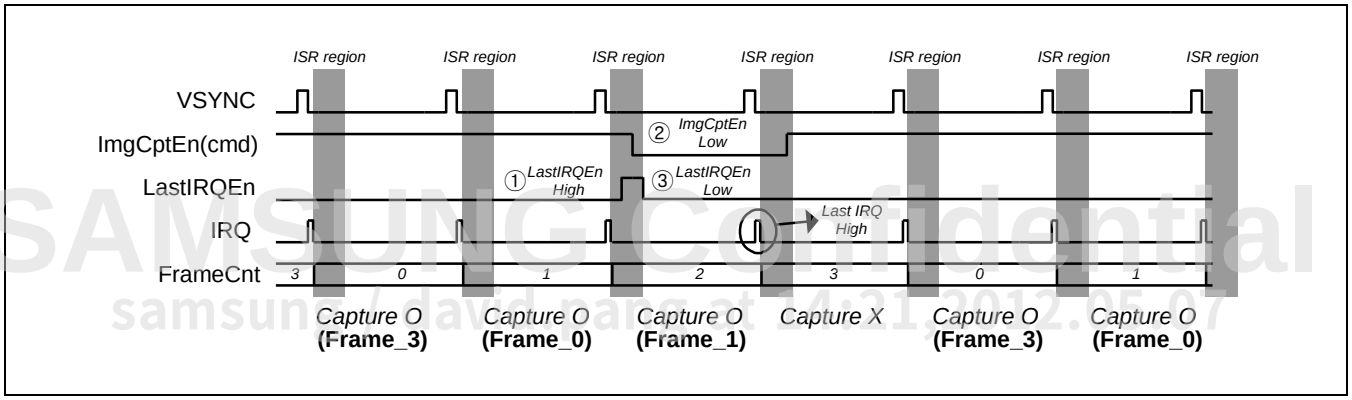


Figure 42-16 Timing Diagram for Last IRQ (LastIRQEn is Enabled)

Figure 42-17 illustrates the diagram for Last IRQ (LastIRQEn is disabled) and timing requirement.

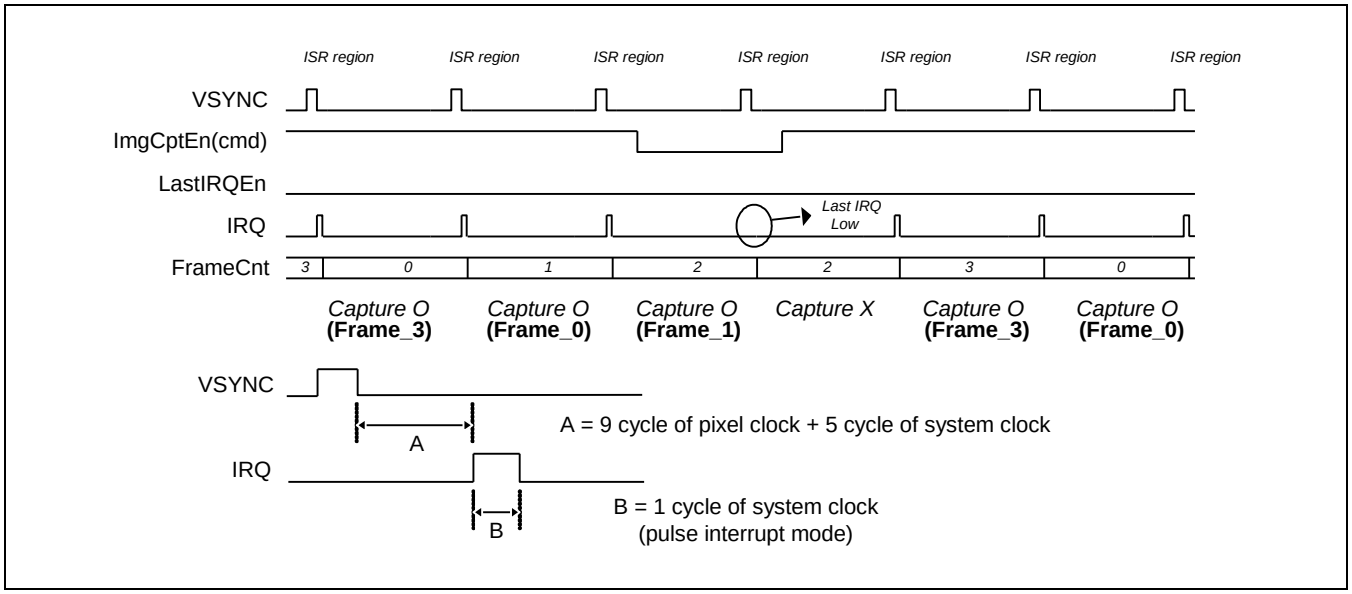


Figure 42-17 Diagram for Last IRQ (LastIRQEn is Disabled) and Timing Requirement

42.6.7 Timing Diagram for IRQ (Memory Data Scaling Mode)

You can select the Input DMA by setting SFR. Generates IRQ after completing the DMA operation for each frame. The SFR setting (ENVID_M "0" → "1") makes this mode aware of the starting point. Therefore, this mode does not require IRQ of starting point and LastIRQ. FrameCnt will increase by 1 at at ENVID_M low to rising ("0" → "1") and ImgCptEn_SC "1".

Figure 42-18 illustrates the timing diagram for IRQ (input DMA path).

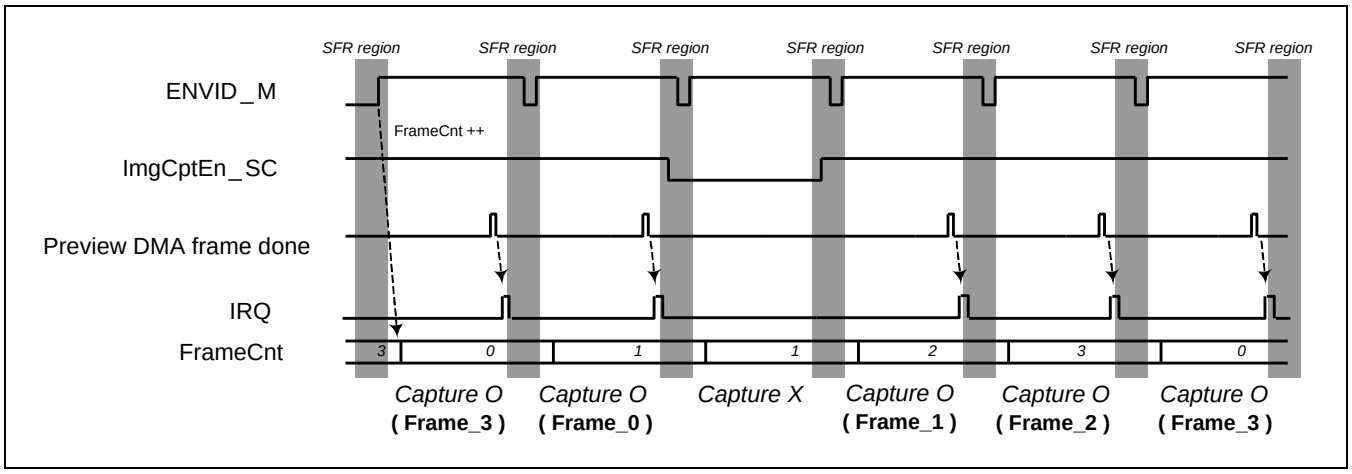


Figure 42-18 Timing Diagram for IRQ (Input DMA Path)

42.6.8 Input DMA Feature

Input DMA supports memory data scaling.

Picture-in-Picture (PIP) operation requires two different image data. Codecs like H.264, Camera, MPEG4, and so on saves the first image in memory while saves the second image through input DMA path.

The input DMA path comprises of YCbCr/RGB output format through scaler/DMA path. The LCD controller displays and controls the images.

If you use input DMA (reading the memory date) in the path, then you should set the SEL_DMA_CAM (MSCTRL bit[3]) signal to "1". This input path is called Memory Scaling DMA path. It disables window zoom function in Memory Scaling DMA path.

NOTE: The memory image format for input DMA input includes:

- YCbCr 4:2:0 (non-interleave)
- YCbCr 4:2:2 (non-interleave)
- YCbCr 4:2:2 (Interleave)
- RGB

Figure 42-19 illustrates the input DMA or external camera interface.

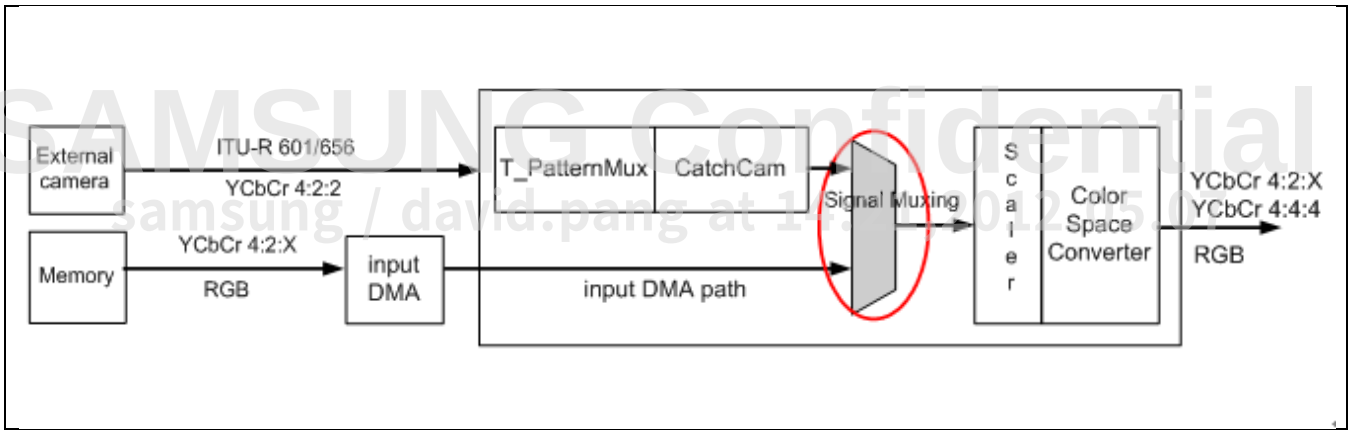


Figure 42-19 Input DMA or External Camera Interface

42.6.9 Camera Interlace Input Support

Exynos 4412 SCP provides two modes to get data from the external camera. The two modes are:

- ITU-R BT 601 YCbCr 8-bit mode
- ITU-R BT 656 YCbCr 8-bit mode

It supports progressive input and interlaced input in both the modes.

42.6.9.1 Progressive Input

Frame unit stores all input data sequentially in 32 buffers (that is in ping-pong memory designated by SFR) in progressive mode. For more information, refer to *Figure 42-20*.

42.6.9.2 Interlaced Input

In interlaced mode, it stores all input data in buffers (that is in ping-pong memory designated by SFR). In this mode store field frame data and odd field frame data successively. Therefore, it stores even field frame data in odd number ping-pong memories like first and third and stores odd field frame data in even number ping-pong memories like second and fourth. Start frame is always even filed frame in case of image capture.

Figure 42-20 illustrates the frame buffer control.

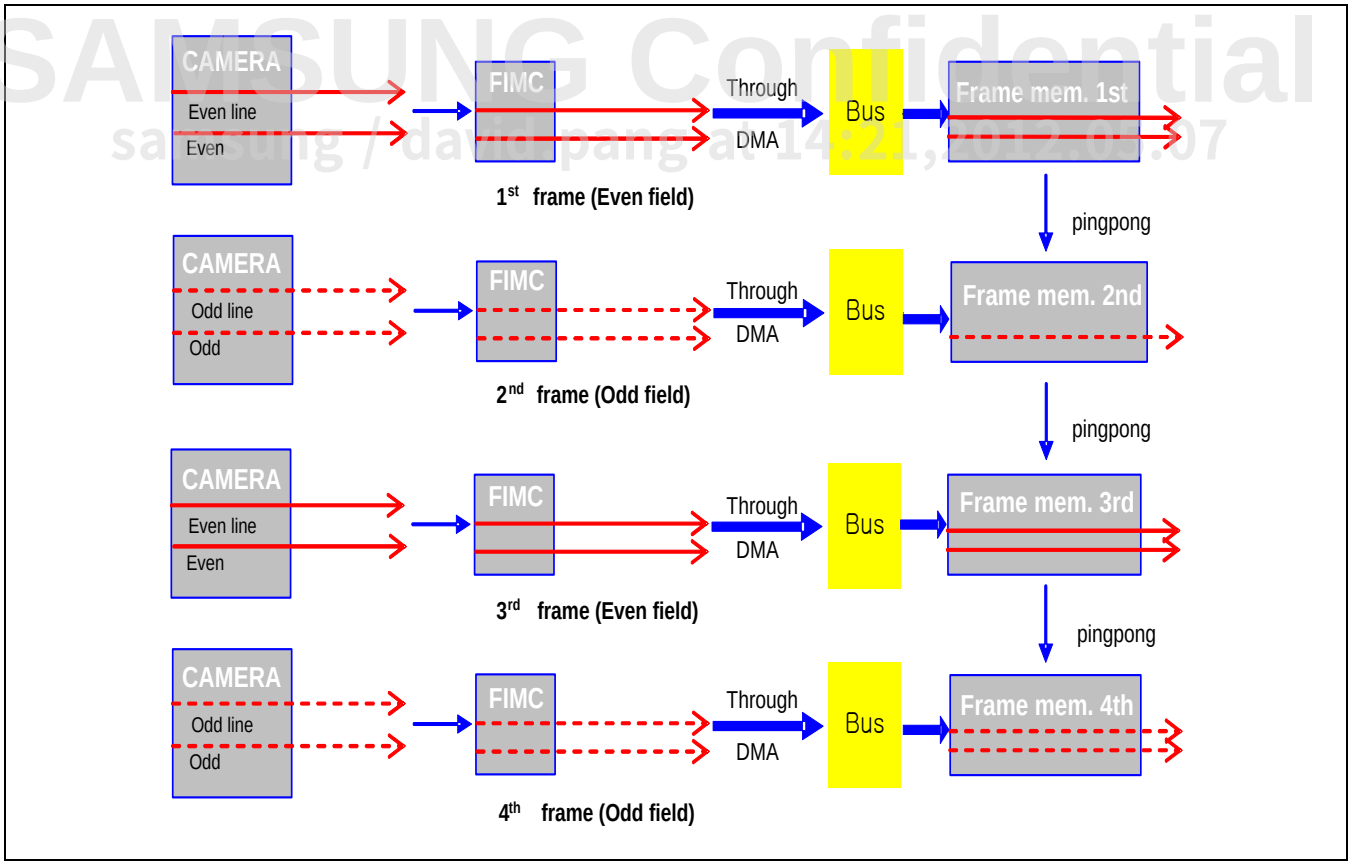


Figure 42-20 Frame Buffer Control

42.7 Input/Output Description

Table 42-5 describes the ITU camera interface signal description.

Table 42-5 ITU Camera Interface Signal Description

Signal	I/O	Description	PAD	Type
CAM_A_PCLK	I	Specifies the pixel clock driven by camera A.	XciPCLK	Muxed
CAM_A_VSYNC	I	Specifies the frame sync driven by camera A.	XciVSYNC	Muxed
CAM_A_HREF	I	Specifies the horizontal sync driven by camera A.	XciHREF	Muxed
CAM_A_DATA [7:0]	I	Specifies the pixel data driven by camera A.	XciDATA[7:0]	Muxed
CAM_A_FIELD	I	Specifies the field signal driven by camera A.	XciFIELD	Muxed
CAM_B_PCLK	I	Specifies the pixel clock driven by camera B.	XmdmWEn	Muxed
CAM_B_VSYNC	I	Specifies the frame sync driven by camera B.	XmdmCSn	Muxed
CAM_B_HREF	I	Specifies the horizontal sync driven by camera B.	XmdmRn	Muxed
CAM_B_DATA[7:0]	I	Specifies the pixel data driven by camera B.	XmdmADDR[7:0]	Muxed
CAM_B_FIELD	I	Specifies the field signal driven by camera B.	XmdmIRQn	Muxed

NOTE: I/O direction. I = input, O = output, and B = bi-direction.
Type field indicates, whether it dedicates pads to the signal or connects pads to the multiplexed signals.
CAM_MCLK_A and CAM_MCLK_B signals are output of CMU.
CAM_MCLK_A = Specifies the Clock for external Camera processor A. (Output, PAD: XciCLKenb, Type: Muxed)
CAM_MCLK_B = Specifies the Clock for external Camera processor B. (Output, PAD: XmdmADVN, Type: Muxed)

42.8 Register Description

42.8.1 Register Map Summary

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000

Register	Offset	Description	Reset Value
CISRCFMTn	0x0000	Specifies input source format.	0x0000_0000
CIWDOFSTn	0x0004	Specifies window offset register.	0x0000_0000
CIGCTRLn	0x0008	Specifies global control register.	0x2001_0480
CIWDOFST2n	0x0014	Specifies window offset register 2.	0x0000_0000
CIOYSA1n	0x0018	Specifies 1st Y frame start address for output DMA.	0x0000_0000
CIOYSA2n	0x001C	Specifies 2nd Y frame start address for output DMA.	0x0000_0000
CIOYSA3n	0x0020	Specifies 3rd Y frame start address for output DMA.	0x0000_0000
CIOYSA4n	0x0024	Specifies 4th Y frame start address for output DMA.	0x0000_0000
CIOCBSA1n	0x0028	Specifies 1st Cb frame start address for output DMA.	0x0000_0000
CIOCBSA2n	0x002C	Specifies 2nd Cb frame start address for output DMA.	0x0000_0000
CIOCBSA3n	0x0030	Specifies 3rd Cb frame start address for output DMA.	0x0000_0000
CIOCBSA4n	0x0034	Specifies 4th Cb frame start address for output DMA.	0x0000_0000
CIOCRSA1n	0x0038	Specifies 1st Cr frame start address for output DMA.	0x0000_0000
CIOCRSA2n	0x003C	Specifies 2nd Cr frame start address for output DMA.	0x0000_0000
CIOCRSA3n	0x0040	Specifies 3rd Cr frame start address for output DMA.	0x0000_0000
CIOCRSA4n	0x0044	Specifies 4th Cr frame start address for output DMA.	0x0000_0000
CITRGFMTn	0x0048	Specifies target image format.	0x0000_0000
CIOCTRLn	0x004C	Specifies control-related Output DMA.	0x0000_0000
CISCPRERATIOn	0x0050	Specifies pre-scalar control 1.	0x0000_0000
CISCPREDSTn	0x0054	Specifies pre-scalar control 2.	0x0000_0000
CISCCTRLn	0x0058	Specifies main-scalar control.	0x1800_0000
CITAREAn	0x005C	Specifies target area.	0x0000_0000
CIOLINESKIPn	0x0060	Specifies output DMA line skip.	0x0000_0000
CISTATUSn	0x0064	Specifies status register.	0x0000_0800
CISTATUS2n	0x0068	Specifies status2 register.	0x0000_0000
CIIMGCPTn	0x00C0	Specifies image capture enable command.	0x0000_0000
CICPTSEQn	0x00C4	Specifies sequence-related capture.	0xFFFF_FFFF
CITHOLDn	0x00C8	Specifies QoS threshold.	0x0000_0000
CIIMGEFFn	0x00D0	Specifies image related image effects.	0x0010_0080
CIIYSA0n	0x00D4	Specifies Y frame start address for Input DMA.	0x0000_0000
CIICBSA0n	0x00D8	Specifies Cb frame start address for Input DMA.	0x0000_0000
CIICRSA0n	0x00DC	Specifies Cr frame start address for Input DMA.	0x0000_0000
CIILINESKIP_Yn	0x00EC	Specifies input DMA Y line skip.	0x0000_0000

Register	Offset	Description	Reset Value
CIILINESKIP_Cbn	0x00F0	Specifies input DMA Cb line skip.	0x0000_0000
CIILINESKIP_Crn	0x00F4	Specifies input DMA Cr line skip.	0x0000_0000
CIREAL_ISIZEn	0x00F8	Specifies real input DMA image size.	0x0000_0000
MSCTRLn	0x00FC	Specifies input DMA control register.	0x0400_0000
CIIYSA1n	0x0144	Specifies Y frame start address 1 for Input DMA.	0x0000_0000
CIICBSA1n	0x0148	Specifies Cb frame start address 1 for Input DMA.	0x0000_0000
CIICRSA1n	0x014C	Specifies Cr frame start address 1 for Input DMA.	0x0000_0000
CIOYOFFn	0x0168	Specifies Y offset of output DMA.	0x0000_0000
CIOCBOFFn	0x016C	Cb offset of output DMA.	0x0000_0000
CIOCROFFn	0x0170	Specifies Cr offset of output DMA.	0x0000_0000
CIIYOFFn	0x0174	Specifies Y offset of input DMA.	0x0000_0000
CIICBOFFn	0x0178	Specifies Cb offset of input DMA.	0x0000_0000
CIICROFFn	0x017C	Specifies Cr offset of input DMA.	0x0000_0000
ORGISIZEn	0x0180	Specifies original image size of input DMA.	0x0000_0000
ORGOSIZEn	0x0184	Specifies original image size of output DMA.	0x0000_0000
CIEXTENn	0x0188	Specifies image size of real output DMA.	0x0000_0000
CIDMAPARAMn	0x018C	Specifies DMA parameter register.	0x0000_0000
CSIIMGFMTn	0x0194	Specifies MIPI CSI image format register.	0x0000_001E
CIKEYn	0x019C	Specifies key detect register.	0x0000_0000
CIINTER420n	0x01A0	Specifies output interlace YCbCr420 sampling register.	0x0000_0000
CIFCNTSEQn	0x01FC	Specifies output frame buffer sequence register.	0xFFFF_FFFF
CIOYSA5n	0x0200	Specifies 5th Y frame start address for output DMA.	0x0000_0000
CIOYSA6n	0x0204	Specifies 6th Y frame start address for output DMA.	0x0000_0000
CIOYSA7n	0x0208	Specifies 7th Y frame start address for output DMA.	0x0000_0000
CIOYSA8n	0x020C	Specifies 8th Y frame start address for output DMA.	0x0000_0000
CIOYSA9n	0x0210	Specifies 9th Y frame start address for output DMA.	0x0000_0000
CIOYSA10n	0x0214	Specifies 10th Y frame start address for output DMA.	0x0000_0000
CIOYSA11n	0x0218	Specifies 11th Y frame start address for output DMA.	0x0000_0000
CIOYSA12n	0x021C	Specifies 12th Y frame start address for output DMA.	0x0000_0000
CIOYSA13n	0x0220	Specifies 13th Y frame start address for output DMA.	0x0000_0000
CIOYSA14n	0x0224	Specifies 14th Y frame start address for output DMA.	0x0000_0000
CIOYSA15n	0x0228	Specifies 15th Y frame start address for output DMA.	0x0000_0000
CIOYSA16n	0x022C	Specifies 16th Y frame start address for output DMA.	0x0000_0000
CIOYSA17n	0x0230	Specifies 17th Y frame start address for output DMA.	0x0000_0000
CIOYSA18n	0x0234	Specifies 18th Y frame start address for output DMA.	0x0000_0000
CIOYSA19n	0x0238	Specifies 19th Y frame start address for output DMA.	0x0000_0000

Register	Offset	Description	Reset Value
CIOYSA20n	0x023C	Specifies 20th Y frame start address for output DMA.	0x0000_0000
CIOYSA21n	0x0240	Specifies 21th Y frame start address for output DMA.	0x0000_0000
CIOYSA22n	0x0244	Specifies 22th Y frame start address for output DMA.	0x0000_0000
CIOYSA23n	0x0248	Specifies 23th Y frame start address for output DMA.	0x0000_0000
CIOYSA24n	0x024C	Specifies 24th Y frame start address for output DMA.	0x0000_0000
CIOYSA25n	0x0250	Specifies 25th Y frame start address for output DMA.	0x0000_0000
CIOYSA26n	0x0254	Specifies 26th Y frame start address for output DMA.	0x0000_0000
CIOYSA27n	0x0258	Specifies 27th Y frame start address for output DMA.	0x0000_0000
CIOYSA28n	0x025C	Specifies 28th Y frame start address for output DMA.	0x0000_0000
CIOYSA29n	0x0260	Specifies 29th Y frame start address for output DMA.	0x0000_0000
CIOYSA30n	0x0264	Specifies 30th Y frame start address for output DMA.	0x0000_0000
CIOYSA31n	0x0268	Specifies 31th Y frame start address for output DMA.	0x0000_0000
CIOYSA32n	0x026C	Specifies 32th Y frame start address for output DMA.	0x0000_0000
CIOCBSA5n	0x0270	Specifies 5th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA6n	0x0274	Specifies 6th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA7n	0x0278	Specifies 7th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA8n	0x027C	Specifies 8th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA9n	0x0280	Specifies 9th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA10n	0x0284	Specifies 10th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA11n	0x0288	Specifies 11th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA12n	0x028C	Specifies 12th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA13n	0x0290	Specifies 13th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA14n	0x0294	Specifies 14th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA15n	0x0298	Specifies 15th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA16n	0x029C	Specifies 16th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA17n	0x02A0	Specifies 17th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA18n	0x02A4	Specifies 18th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA19n	0x02A8	Specifies 19th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA20n	0x02AC	Specifies 20th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA21n	0x02B0	Specifies 21st Cb frame start address for output DMA.	0x0000_0000
CIOCBSA22n	0x02B4	Specifies 22nd Cb frame start address for output DMA.	0x0000_0000
CIOCBSA23n	0x02B8	Specifies 23rd Cb frame start address for output DMA.	0x0000_0000
CIOCBSA24n	0x02BC	Specifies 24th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA25n	0x02C0	Specifies 25th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA26n	0x02C4	Specifies 26th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA27n	0x02C8	Specifies 27th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA28n	0x02CC	Specifies 28th Cb frame start address for output DMA.	0x0000_0000

Register	Offset	Description	Reset Value
CIOCBSA29n	0x02D0	Specifies 29th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA30n	0x02D4	Specifies 30th Cb frame start address for output DMA.	0x0000_0000
CIOCBSA31n	0x02D8	Specifies 31st Cb frame start address for output DMA.	0x0000_0000
CIOCBSA32n	0x02DC	Specifies 32nd Cb frame start address for output DMA.	0x0000_0000
CIOCRSA5n	0x02E0	Specifies 5th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA6n	0x02E4	Specifies 6th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA7n	0x02E8	Specifies 7th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA8n	0x02EC	Specifies 8th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA9n	0x02F0	Specifies 9th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA10n	0x02F4	Specifies 10th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA11n	0x02F8	Specifies 11th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA12n	0x02FC	Specifies 12th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA13n	0x0300	Specifies 13th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA14n	0x0304	Specifies 14th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA15n	0x0308	Specifies 15th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA16n	0x030C	Specifies 16th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA17n	0x0310	Specifies 17th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA18n	0x0314	Specifies 18th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA19n	0x0318	Specifies 19th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA20n	0x031C	Specifies 20th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA21n	0x0320	Specifies 21st Cr frame start address for output DMA.	0x0000_0000
CIOCRSA22n	0x0324	Specifies 22ndCr frame start address for output DMA.	0x0000_0000
CIOCRSA23n	0x0328	Specifies 23rd Cr frame start address for output DMA.	0x0000_0000
CIOCRSA24n	0x032C	Specifies 24th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA25n	0x0330	Specifies 25th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA26n	0x0334	Specifies 26th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA27n	0x0338	Specifies 27th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA28n	0x033C	Specifies 28th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA29n	0x0340	Specifies 29th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA30n	0x0344	Specifies 30th Cr frame start address for output DMA.	0x0000_0000
CIOCRSA31n	0x0348	Specifies 31st Cr frame start address for output DMA.	0x0000_0000
CIOCRSA32n	0x034C	Specifies 32nd Cr frame start address for output DMA.	0x0000_0000

NOTE: The last "L" word means that SFR can change at vsync edge during camera capture.
(O = Possible change, X = Impossible change).
Also, "M" word means that SFRs have relationship capturing result while using input DMA path.
(O = Relationship, X = No relationship).

42.8.1.1 CISRCFMTn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ITU601_656n	[31]	RW	0 = ITU-R BT.656 YCbCr 8-bit mode enable 1 = ITU-R BT.601 YCbCr 8-bit mode enable (ML = XX)	0
UVOffset	[30]	RW	Controls Cb,Cr value offset. 0 = 0 (normally used) 1 = Cb = Cb + 128, Cr = Cr + 128 (ML = XX)	0
RSVD	[29]	–	Reserved (Should be "0")	0
SrcHsize_CAM	[28:16]	RW	Specifies the horizontal source pixel number (camera or FIFO input). For more information, refer to *42.8.1.49 CIEXTENn (n = 0 to 3)* gathering extension register (SrcHsize_CAM_ext). NOTE: If WinOfsEn = 0, SrcHsize_CAM is 4's multiple of PreHorRatio. Otherwise, it is 16's multiple (ML = XO)	0
Order422_CAM	[15:14]	RW	Specifies YCbCr camera input order for 8-bit mode. 8-bit mode Data Flow → 00 = Y0Cb0Y1Cr0… 01 = Y0Cr0Y1Cb0… 10 = Cb0Y0Cr0Y1… 11 = Cr0Y0Cb0Y1… (ML = XX)	0
SrcVsize_CAM	[13:0]	RW	Specifies the vertical source pixel number (Camera or FIFO input). NOTE: If V scale down or WinOfsEn = 0, then it is multiple of PreVerRatio. If YCbCr 422 input and cam interlace mode, then it is 2'multiple. (ML = XO)	0

42.8.1.2 CIWDOFSTn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
WinOfsEn	[31]	RW	0 = No offset 1 = Enables window offset NOTE: If input format = RAW or WB (Write Back), then the function is invalid. (ML = XO)	0
ClrOvFiY	[30]	RW	0 = Normal 1 = Clears Y input FIFO overflow indication flag (ML = XX)	0
ClrOvRLB	[29]	RW	Clears line buffer overflow indication flag for rotation. (ML = XX)	0
RSVD	[28:27]	RW	Reserved	0
WinHorOfst	[26:16]	RW	Specifies horizontal window offset in pixel unit. It is multiple of 2. Refer to gathering extension register WinHorOfst_ext for more information. (ML = XO)	0
ClrOvFiCb	[15]	RW	0 = Normal 1 = Clears Cb input FIFO overflow indication flag (ML = XX)	0
ClrOvFiCr	[14]	RW	0 = Normal 1 = Clears Cr input FIFO overflow indication flag (ML = XX)	0
RSVD	[13:12]	–	Reserved	0
WinVerOfst	[11:0]	RW	Specifies vertical window offset in pixel unit. In case of interlaced input, this value should be 2's multiple. (ML = XO)	0

NOTE: Clear bits are set to zero after clearing the flags.

Figure 42-21 illustrates the camera window offset scheme.

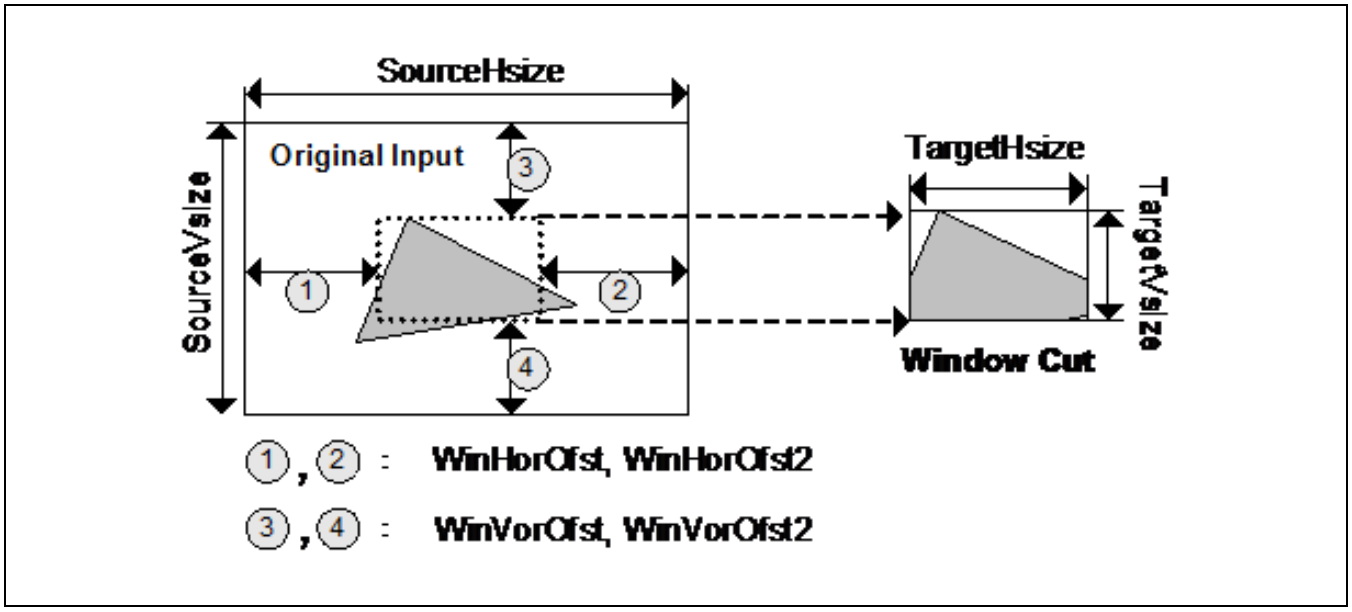


Figure 42-21 Camera Window Offset Scheme

WinHorOfst2 and WinVerOfst2 are assigned in the CIWDOFST2n registers.

Below constraints of Crop HSIZE and Crop Vsize are only for CAMIF0, CAMIF1, CAMIF2, and CAMIF3.

Crop Hsize (= SourceHsize – WinHorOfst – WinHorOfst2) should be 16's multiple. Also, it should be 4's multiple of PreHorRatio.

Crop Vsize (= SourceVsize – WinVerOfst – WinVerOfst2should be multiple of PreVerRatio when V scale is down. It should be an even number and minimum eight if the output image format is YCbCr 420.

Example:

Crop Hsize	Permitted Prescale_ratio	PreDstWidth_xx
8n	2	4n
16n	2 or 4	4n
32n	2, 4 or 8	4n

42.8.1.3 CIGCTRLn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0008, Reset Value = 0x2001_0480

Name	Bit	Type	Description	Reset Value
SwRst	[31]	RW	Specifies the camera interface software reset. Before you set this bit, set the ITU601_656n bit of CISRCFMT as 1 temporarily at first SFR setting. Next sequence is recommended. [ITU601 case = ITU601_656n "1" → SwRst "1" → SwRst "0" for first SFR setting], [ITU656 case = ITU601_656n "1" → SwRst "1" → SwRst "0" → ITU601_656n "0" for first SFR setting] NOTE: 1. You should not use SwRst function in the middle of transferring data out from DMA. 2. You should disable "ImgCptEn" and "IRQ_Enable" bit before using this function. (ML = XX)	0
RSVD	[30]	–	Reserved (Should be set to 0)	0
SelCam_ITU	[29]	RW	Selects external multiple ITU camera 0 = Selects ITU Camera B 1 = Selects ITU Camera A (ML = XX)	1
TestPattern	[28:27]	RW	You should set this register as ITU-T 601 8-bit mode and not as ITU-T 656 mode. Source CAM size should be 640 × 480 size and Order422_CAM register should be set to 00. 00 = External camera processor input (normal) 01 = Color bar test pattern  10 = Reserved 11 = Reserved (ML = XX)	0

Name	Bit	Type	Description	Reset Value
InvPolPCLK	[26]	RW	0 = Sets polarity of PCLK as normal 1 = Sets polarity of PCLK as inverse (ML = XX)	0
InvPolVSYNC	[25]	RW	0 = Sets polarity of VSYNC as normal 1 = Sets polarity of VSYNC as inverse (ML = XX)	0
InvPolHREF	[24]	RW	0 = Sets polarity of HREF as normal 1 = Sets polarity of HREF as inverse (ML = XX)	0
RSVD	[23]	–	Reserved	0
IRQ_Ovfen	[22]	RW	0 = Disables Overflow interrupt (normal) 1 = Enables Overflow interrupt (generates interrupt during overflow occurrence) (ML = XX)	0
Href_mask	[21]	RW	0 = No mask 1 = Mask out Href during Vsync blank (ML = XX)	0
RSVD	[20]	–	Reserved (Should be set to 1)	0
IRQ_CLR	[19]	RW	Writes IRQ_CLR to 1 to clear Interrupt. This bit Autoclears. (ML = XX)	0
IRQ_EndDisable	[18]	RW	This bit is related to Camera or Local FIFO (WB) input only. 0 = Disables interrupt at frame end point (default) 1 = Enables interrupt at frame end point (ML = XX)	0
IRQ_StartEnable	[17]	RW	This bit is related to Camera or Local FIFO (WB) input only. 0 = Enables interrupt at frame start point (default) 1 = Disables interrupt at frame start point (ML = XX)	0
IRQ_Enable	[16]	RW	0 = Disables Interrupt 1 = Enables Interrupt (default) NOTE: If the interrupt enables, then the bit[20] at CIGCTRLn is set to "1". (ML = XX)	1
RSVD	[15:14]	–	Reserved	0
SwUpdate	[13]	RW	Updates shadow register by software setting (You can apply both DMA input and camera input) 0 = Updates shadow register at hardware frame start pulse 1 = Updates shadow register immediately (Auto clears 1 to 0) (ML = XX)	0

Name	Bit	Type	Description	Reset Value
ShadowDisable	[12]	RW	Hardware frame start (input path should be local path) cannot update the shadow register.). At the start of first frame, it is necessary to set Shadow Disable as "0". 0 = Enables (update is possible) 1 = Disables (update is impossible) (ML = XX)	0
RSVD	[11]	–	Reserved	0
Sel_WB	[10]	RW	Multiple WriteBack I/F select 0 = Selects WriteBack B 1 = Selects WriteBack A	1
RSVD	[9]	–	Reserved (Should be set to "0")	0
CAM_JPEG	[8]	RW	Specifies the camera input in 8-bit JPEG format Image format conversion is not possible if you select JPEG image format. You should set the camera in scalar bypass mode and ITU601 8-bit mode. 0 = Selects non-JPEG format 1 = Selects JPEG format (ML = XX)	0
SelCam_MIPI	[7]	RW	Selects multiple external MIPI cameras. 0 = Selects MIPI Camera B 1 = Selects MIPI Camera A	1
SelWB_CAMIF	[6]	RW	Specifies the WriteBack input select signal. 0 = Selects camera input select signal 1 = Selects WriteBack input select signal (YCbCr4:4:4 only) (ML = XX)	0
CSC_601_709	[5]	RW	Selects Color Space Conversion equation. 0 = Selects ITU601 equation (SD size target method) 1 = Selects ITU709 equation (HD size target method) (ML = XO)	0
InvPolHSYNC	[4]	RW	0 = Normal 1 = Inverses the polarity of HSYNC (Use this bit only when you connect delay count interlace mode and FIELD port to HSYNC) (ML = XX)	0
SelCam_CAMIF	[3]	RW	Selects the External camera. 0 = Selects ITU Camera 1 = Selects MIPI Camera (ML = XX)	0
FIELDMODE	[2]	RW	Specifies the ITU601 interlace field mode (Do not use this bit in ITU656 mode). 0 = Uses the Edge delay count mode (FIELD port = HSYNC signal) 1 = Uses the FIELD port mode (FIELD port = FIELD signal)	0

Name	Bit	Type	Description	Reset Value
			NOTE: Check the FIELD port connection. (ML = XX)	
InvPolFIELD	[1]	RW	0 = Normal 1 = Inverses the polarity of FIELD (ML = XX)	0
Cam_Interlace	[0]	RW	Specifies the External Camera scan method. 0 = Selects Progressive method 1 = Selects Interlace method If you enable this mode, you cannot change control signals under operation except ImgCptEn,ImgCptEnSC and ScalerStart (ML = XX)	0

Figure 42-22 illustrates the interrupt generation scheme.

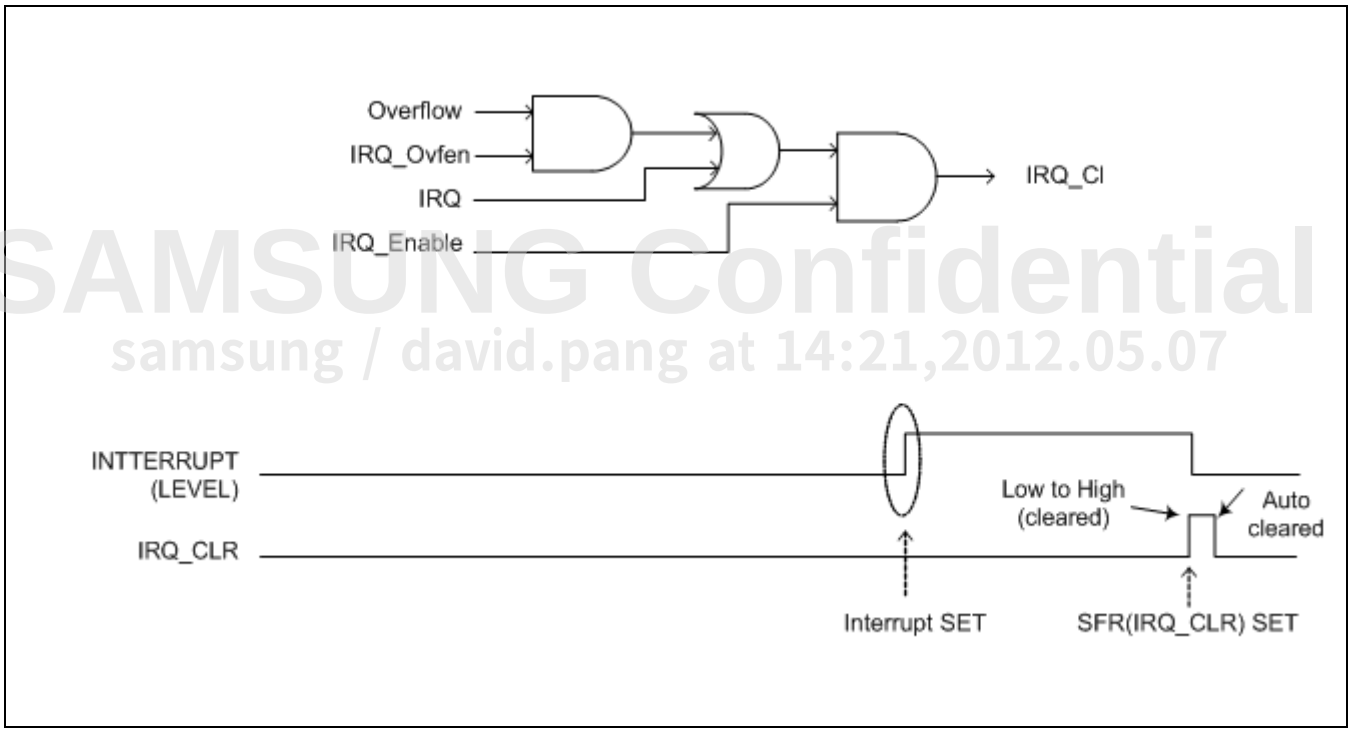


Figure 42-22 Interrupt Generation Scheme

42.8.1.4 CIWDOFST2n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0
WinHorOfst2	[27:16]	RW	Specifies the horizontal window offset2 by pixel unit. It should be multiple of 2. (ML = XO)	0
RSVD	[15:12]	–	Reserved	0
WinVerOfst2	[11:0]	RW	Specifies the vertical window offset2 by pixel unit. In case of interlaced input, this value should be 2's multiple. (ML = XO)	0

42.8.1.5 CIOYSA1n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA1	[31:0]	RW	Output format = YCbCr 2/3 plane → 1st Y frame start address. Output format = YCbCr 1 plane → 1st YCbCr frame start address. Output format = RGB → 1st RGB frame start address. NOTE: In tile mode, this value is aligned 4 KB that is CIOYSA1[11:0] is 0x000. (ML = OX)	0

42.8.1.6 CIOYSA2n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA2	[31:0]	RW	Output format = YCbCr 2/3 plane → 2nd Y frame start address. Output format = YCbCr 1 plane → 2nd YCbCr frame start address. Output format = RGB → 2nd RGB frame start address. NOTE: In tile mode, this value is aligned 4 KB that is CIOYSA2[11:0] is 0x000. (ML = OX)	0

42.8.1.7 CIOYSA3n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA3	[31:0]	RW	Output format = YCbCr 2/3 plane → 3rd Y frame start address. Output format = YCbCr 1 plane → 3rd YCbCr frame start address. Output format = RGB → 3rd RGB frame start address. NOTE: In tile mode, this value is aligned 4 KB that is CIOYSA3[11:0] is 0x000. (ML = OX)	0

42.8.1.8 CIOYSA4n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA4	[31:0]	RW	Output format = YCbCr 2/3 plane → 4th Y frame start address. Output format = YCbCr 1 plane → 4th YCbCr frame start address. Output format = RGB → 4th RGB frame start address. NOTE: In tile mode, this value is aligned 4 KB that is CIOYSA4[11:0] is 0x000. (ML = OX)	0

42.8.1.9 CIOCBSA1n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA1	[31:0]	RW	Output format = YCbCr 3 plane → 1st Cb frame start address. Output format = YCbCr 2 plane → 1st CbCr frame start address. NOTE: In tile mode, this value is aligned 4 KB that is CIOCBSA1[11:0] is 0x000. (ML = OX)	0

42.8.1.10 CIOCBSA2n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA2	[31:0]	RW	Output format = YCbCr 3 plane → 2nd Cb frame start address. Output format = YCbCr 2 plane → 2nd CbCr frame start address. NOTE: In tile mode, this value is aligned 4 KB that is CIOCBSA2[11:0] is 0x000. (ML = OX)	0

42.8.1.11 CIOCBSA3n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA3	[31:0]	RW	Output format = YCbCr 3 plane → 3rd Cb frame start address. Output format = YCbCr 2 plane → 3rd CbCr frame start address. NOTE: In tile mode, this value is aligned 4 KB that is CIOCBSA3[11:0] is 0x000. (ML = OX)	0

42.8.1.12 CIOCBSA4n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA4	[31:0]	RW	Output format = YCbCr 3 plane → 4th Cb frame start address. Output format =YCbCr 2 plane → 4th CbCr frame start address. NOTE: In tile mode, this value is aligned 4 KB that is CIOCBSA4[11:0] is 0x000. (ML = OX)	0

42.8.1.13 CIOCRSA1n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0038, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA1	[31:0]	RW	Output format = YCbCr 3 plane → 1st Cr frame start address. NOTE: In tile mode, this value is aligned 4 KB that is CIOCRSA1[11:0] is 0x000. (ML = OX)	0

42.8.1.14 CIOCRSA2n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x003C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA2	[31:0]	RW	Output format = YCbCr 3 plane → 2nd Cr frame start address NOTE: In tile mode, this value is aligned 4 KB that is CIOCRSA2 [11:0] is 0x000. (ML = OX)	0

42.8.1.15 CIOCRSA3n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA3	[31:0]	RW	Output format = YCbCr 3 plane → 3rd Cr frame start address. NOTE: In tile mode, this value is aligned 4 KB that is CIOCRSA3[11:0] is 0x000. (ML = OX)	0

42.8.1.16 CIOCRSA4n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA4	[31:0]	RW	Output format = YCbCr 3 plane → 4th Cr frame start address. NOTE: In tile mode, this value is aligned 4 KB that is CIOCRSA4[11:0] is 0x000. (ML = OX)	0

42.8.1.17 CITRGFMTn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0048, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
InRot90	[31]	RW	0 = Input Rotator bypass 1 = Rotate clockwise 90° (using the Input Rotator only). Input Rotator and Output Rotator do not work at the same time. If you enable Rot90 mode, then the output data path should be LCD FIFO path) (ML = OO)	0
OutFormat	[30:29]	RW	00 = Selects YCbCr 4:2:0 output image format (2 or 3 plane) 01 = Selects YCbCr 4:2:2 output image format (2 or 3 plane) (refer 2 or 3 plane format register → C_INT_OUT) 10 = Selects YCbCr 4:2:2 output image format (1 plane) 11 = Selects RGB output image format (refer RGB format register → OutRGB_FMT) NOTE: 1. For more information, refer to *42.8.1.49 CIEXTENn (n = 0 to 3)*gathering extension register for YCbCr444 format. 2. If you select RAW, camera JPEG or user-defined input format, then the format cannot change and OutFormat register value is invalid. (ML = OO)	0
TargetHsize	[28:16]	RW	Specifies the horizontal pixel number of target image. For more information, refer to *42.8.1.49 CIEXTENn (n = 0 to 3)* gathering extension register (TargetHsize_ext). NOTE: 1. In case of interlaced output DMA and 90-degree-rotation, TargetVsize should be more than 16. 2. In case of YCbCr 420 or 422 output format, TargetHsize should be an even number. (ML = OO)	0
OutFlipMd	[15:14]	RW	Specifies image mirror and rotation for output DMA. 00 = Selects normal mirror. 01 = Selects X-axis mirror. 10 = Selects Y-axis mirror. 11 = Selects 180° rotation. NOTE: You cannot use this function, if input format is CAM_JPEG or MIPI RAW or MIPI User-defined format. (ML = OO)	0
OutRot90	[13]	RW	0 = Output Rotator bypass	0

Name	Bit	Type	Description	Reset Value
			1 = Rotate clockwise 90° (using the Output Rotator) NOTE: You cannot use this function, if input format is CAM_JPEG or MIPI RAW or MIPI user-defined format. (ML = OO)	
TargetVsize	[12:0]	RW	Specifies vertical pixel number of target image. Minimum number is four. For more information, refer to *42.8.1.49 CIEXTENn (n = 0 to 3)* gathering extension register (TargetVsize_ext). NOTE: 1. In case of interlaced output DMA and 90-degree-rotation, TargetVsize should be more than 32. 2. In case of interlaced YCbCr 422 output or 90-degree-rotation, TargetVsize should be multiple of 2. 3. In case of interlaced YCbCr 422 output and 90-degree-rotation, TargetVsize should be multiple of four. (ML = OO)	0

TargetHsize and TargetVsize should not be larger than Camera SourceHsize and Camera SourceVsize. If input mode is DMA, Target image size and InputDMA source size do not have a relationship.

Caution: Input rotator supports only InputDMA image data. The output rotator supports camera or Input DMA image data. Input and output rotators does not work at the same time because it shares input and output rotator memories to save memory size.

NOTE:

1. If the TargetVsize value is set to an odd number (N) when output format is YCbCr 4:2:0, the it generates odd numbers (N) of Y lines and (N – 1)/2 of Cb, Cr lines. Also, it does not allow X-flip or XY-flip. Thus, YCbCr 4:2:0 output format uses an even TargetVsize number.
2. If you cannot divide the TargetVsize value by 4 (4n + 1, 4n + 2, 4n + 3) when output format is YCbCr 4:2:0 and Interlaced out, then it generates the odd number (N) of Y lines and the (N – 1)/2 of Cb, Cr lines. Also, it does not allow X-flip or XY-flip. Thus YCbCr 4:2:0 ouput format and Interlaced out should use 4's multiple TargetVsize number.

Figure 42-23 illustrates the image mirror and rotation.

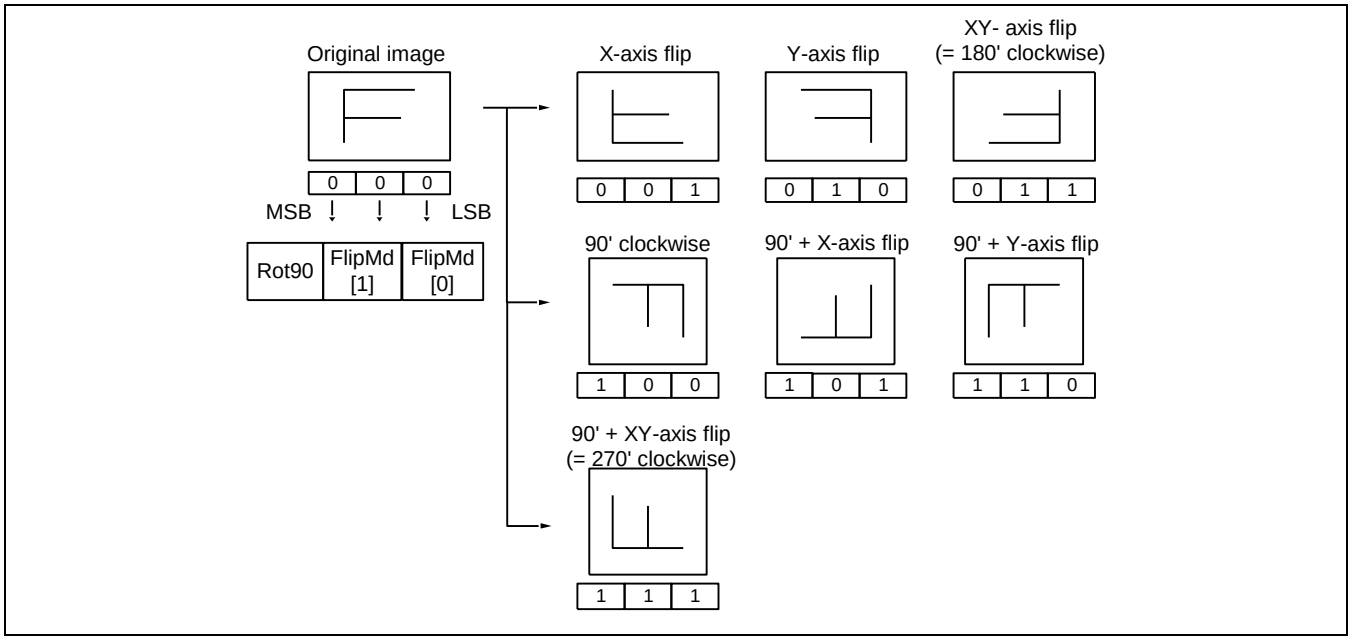


Figure 42-23 Image Mirror and Rotation

42.8.1.18 CIOCTRLn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x004C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Weave_out	[31]	RW	The output DMA weaves even and odd fields and combines to form a complete progressive frame with hardware. This field is useful for interlace DMA output mode (Interlace_out or CAM_INTERLACE). It uses even field address (1st frame start address) as weave address. Odd fields address (2nd frame start address) should be set same as even field address. 0 = Sets as Normal 1 = Sets as Weave (ML = 0X)	0
LastEndEn	[30]	RW	Specifies capture off timing control during the last camera capture. This bit is related to ScalerStart, ImgCptEn_SC and ImgCptEn registers 0 = Off at Next Frame Start trigger timing (Normal) 1 = Off at Last Capture End trigger timing (ML = XX)	0
RSVD	[29:26]	–	Reserved	0
Order2p_out	[25:24]	RW	Specifies YCbCr 4:2:0 or 4:2:2 2plane output chroma memory storing style order (should be C_INT_OUT = 1). bit: MSB … LSB 00: Cr3Cb3Cr2Cb2Cr1Cb1Cr0Cb0 01: Cb3Cr3Cb2Cr2Cb1Cr1Cb0Cr0 10: Reserved 11: Reserved (ML = OO)	0
RSVD	[23:18]	–	Reserved	0
RGB16bFMT	[17:16]	RW	Specifies RGB 16bit format. 00 = Selects RGB 565 format (Alpha_Out is invalid). 01 = Selects ARGB 1555 format (Only Alpha_Out[0] bit is valid). 10 = Selects ARGB 4444 format (Only Alpha_Out[3:0] bit is valid). 11 = Reserved (ML = OO)	0
RSVD	[15:12]	–	Reserved	0
Alpha_Out	[11:4]	RW	Specifies alpha value only for ARGB 8888 or ARGB1555 or ARGB4444 DMA out format If you do not want to use alpha RGB in ARGB888	0

Name	Bit	Type	Description	Reset Value
			format, then you should set RGB888 (24-bit) without alpha to 0x00. (ML = OO)	
C_INT_OUT	[3]	RW	0 = Selects YCbCr 4:2:0 or 4:2:2 or 4:4:4 3plane output format 1 = Selects YCbCr 4:2:0 or 4:2:2 or 4:4:4 2plane output format (ML = OO)	0
LastIRQEn	[2]	RW	0 = Sets to normal 1 = Enables last IRQ at the end of frame capture (You are recommended to check the done signal of capturing image for JPEG.) If LastEndEn is high, this bit should be set to 0 (ML = XX)	0
Order422_out	[1:0]	RW	Specifies YCbCr 4:2:2 1plane output memory storing style order. bit MSB LSB 00 Cr1Y3Cb1Y2Cr0Y1Cb0Y0 01 Cb1Y3Cr1Y2Cb0Y1Cr0Y0 10 Y3Cr1Y2Cb1Y1Cr0Y0Cb0 11 Y3Cb1Y2Cr1Y1Cb0Y0Cr0 (ML = OO)	0

Figure 42-24 illustrates the YCbCr plane memory storing style.

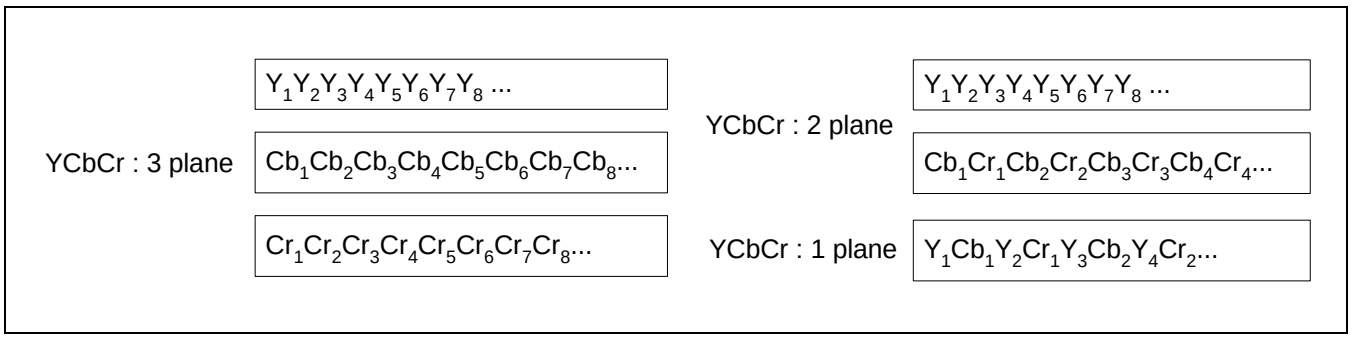


Figure 42-24 YCbCr Plane Memory Storing Style

42.8.1.18.1 Register Setting Guide for Scalar

SRC_Width and DST_Width satisfy the double word (8 bytes) boundary constraints. The number of horizontal pixel represents kn, where n = 1, 2, 3 … and k = 1/2/8 for 24 bpp RGB/16 bpp RGB/YCbCr 420 image.

Figure 42-25 illustrates the scaling scheme.

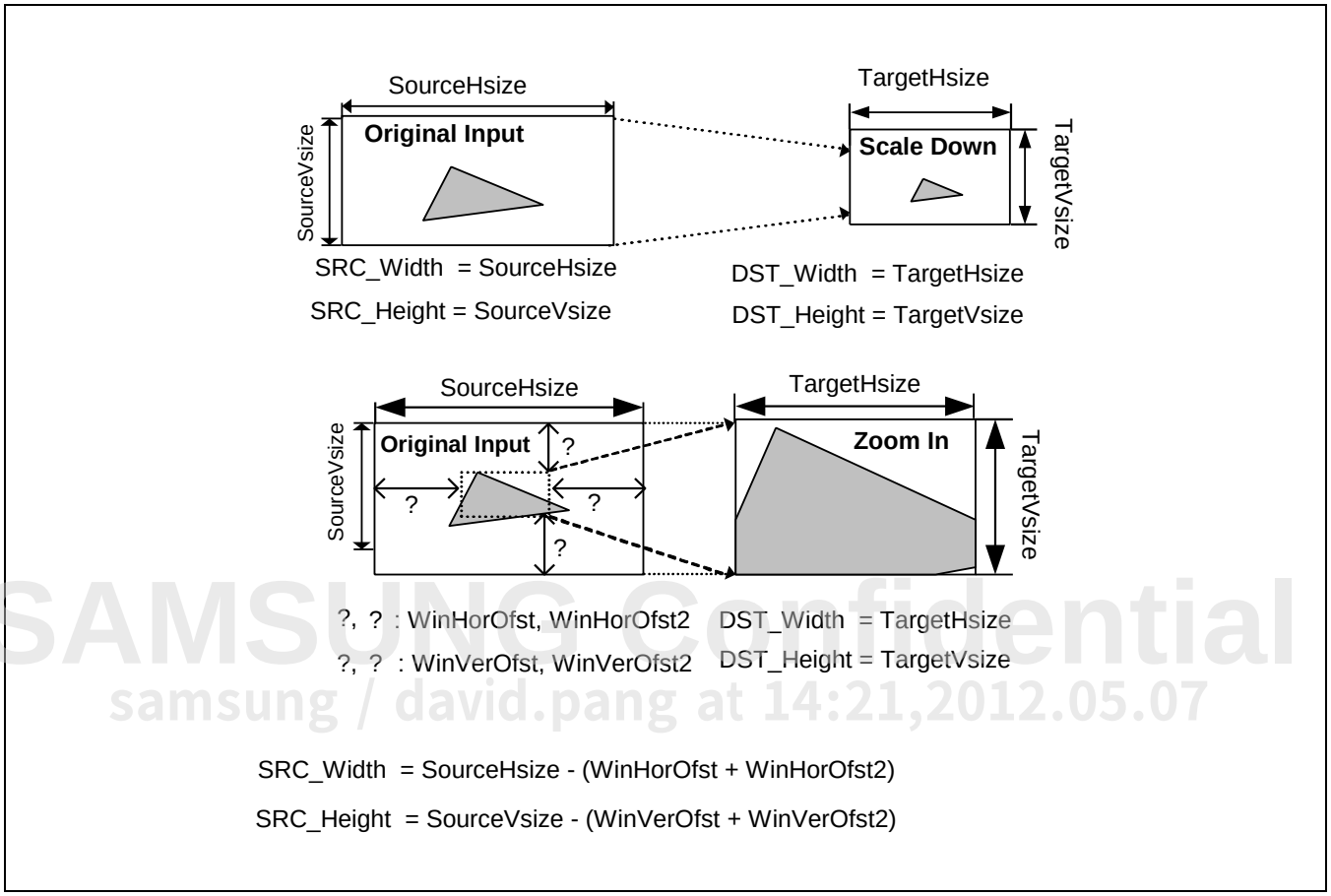


Figure 42-25 Scaling Scheme

Defines other control registers of pre-scale image size, pre-scale ratio, pre-scale shift ratio, and main scale ratio according to the equation here:

```
If (SRC_Width ≥ 64 × DST_Width) {Exit (−1); /* Out Of Horizontal Scale Range */}
      else if (SRC_Width ≥ 32 × DST_Width) {PreHorRatio = 32; H_Shift = 5}
      else if (SRC_Width ≥ 16 × DST_Width) {PreHorRatio = 16; H_Shift = 4}
      else if (SRC_Width ≥ 8 × DST_Width) {PreHorRatio = 8; H_Shift = 3}
      else if (SRC_Width ≥ 4 × DST_Width) {PreHorRatio = 4; H_Shift = 2}
      else if (SRC_Width ≥ 2 × DST_Width) {PreHorRatio = 2; H_Shift = 1}
      else { PreHorRatio = 1; H_Shift = 0}

      PreDstWidth = SRC_Width/PreHorRatio
      MainHorRatio = (SRC_Width << 14)/ (DST_Width << H_Shift)


If (SRC_Height ≥ 64 × DST_Height) {Exit (−1); /* Out Of Vertical Scale Range */}
      else if (SRC_Height ≥ 32 × DST_Height) {PreVerRatio = 32; V_Shift = 5}
      else if (SRC_Height ≥ 16 × DST_Height) {PreVerRatio = 16; V_Shift = 4}
      else if (SRC_Height ≥ 8 × DST_Height) {PreVerRatio = 8; V_Shift = 3}
      else if (SRC_Height ≥ 4 × DST_Height) {PreVerRatio = 4; V_Shift = 2}
      else if (SRC_Height ≥ 2 × DST_Height) {PreVerRatio = 2; V_Shift = 1}
      else {PreVerRatio = 1; V_Shift = 0}

      PreDstHeight = SRC_Height/PreVerRatio
      MainVerRatio = (SRC_Height <<14)/ (DST_Height << V_Shift)

      Specifies DMA output RGB format.

      00 = Selects RGB 16-bit (RGB565, ARGB1555, ARGB4444) format.
      01 = Selects RGB666 format.
      10 = Selects ARGB888 format.
      11 = Reserved
      (For more information, refer RGB 16-bit selection format register → RGB16bFMT)
      (ML = OO) = 10 − (H_Shift + V_Shift)
```


Caution: In case of Zoom-in, you should check the next equation (CAM-In case).
((SourceHsize – (WinHorOfst + WinHorOfst2))/PreHorRatio) ≤ Maximum scalar line buffer size width.

42.8.1.19 CISCPRERATIOn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SHfactor	[31:28]	RW	Specifies the shift factor for pre-scalar. (ML = OO)	0
RSVD	[27:23]	–	Reserved	0
PreHorRatio	[22:16]	RW	Specifies the horizontal ratio of pre-scaler. (ML = OO)	0
RSVD	[15:7]	–	Reserved	0
PreVerRatio	[6:0]	RW	Specifies the vertical ratio of pre-scaler. NOTE: 1. In case of DMA input YCbCr 4:2:2 1plane and input 90-degree-rotation, PreVerRatio should be less than or equal to 16. 2. In case of interlaced DMA output YCbCr 4:2:0 and O_INTER420 = 2, PreVerRatio should be less than or equal to 16. (ML = OO)	0

42.8.1.20 CISCPREDSTn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
PreDstWidth	[29:16]	RW	Specifies the destination width for pre-scalar. (ML = OO)	0
RSVD	[15:14]	–	Reserved	0
PreDstHeight	[13:0]	RW	Specifies the destination height for pre-scalar. (ML = OO)	0

42.8.1.21 CISCCTRLn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0058, Reset Value = 0x1800_0000

Name	Bit	Type	Description	Reset Value
ScalerBypass	[31]	RW	Specifies scalar bypass. In this case, ImgCptEn_SC should be set to "0", but ImgCptEn should be set to "1". Use this mode, to handle a large image whose size is greater than the maximum size of scalar. (This mode captures JPEG input image for DSC application). In this case, the input pixel buffering depends only on input FIFOs. Therefore, the system bus should not be busy in this mode. Scalar Bypass has certain restrictions. For example, it does not allow size scaling, color space conversion, input DMA mode, Write Back mode, and RGB format. If input format is YCbCr4:2:2, then the output format should also be YCbCr4:2:2 or YCbCr4:2:0. If input format is RAW or JPEG format, then the output format cannot change its own format. (ML = XX)	0
ScaleUp_H	[30]	RW	Specifies horizontal scale up/down flag for scalar (In 1:1 scale ratio, this bit should be "1".) 0 = Sets horizontal scale flag Down. 1 = Sets horizontal scale flag Up. (ML = OO)	0
ScaleUp_V	[29]	RW	Specifies vertical scale up/down flag for scalar (In 1:1 scale ratio, this bit should be "1".) 0 = Sets vertical scale flag Down. 1 = Sets vertical scale flag Up. (ML = OO)	0
CSCR2Y	[28]	RW	Selects YCbCr data dynamic range for color space conversion from RGB to YCbCr. 0 = Selects Narrow → Y (16 to 235), Cb/Cr (16 to 240). 1 = Selects Wide → Y/Cb/Cr (0 to 255): Wide default. ※Recommends CSC range setting: CSCR2Y = CSCY2R (Wide = Wide or Narrow = Narrow) (ML = OO)	1
CSCY2R	[27]	RW	Specifies YCbCr data dynamic range selection for the color space conversion from YCbCr to RGB. 0 = Selects Narrow → Y (16 to 235), Cb/Cr (16 to 240). 1 = Selects Wide → Y/Cb/Cr (0 to 255): Wide default. (ML = OO)	1

Name	Bit	Type	Description	Reset Value
LCDPathEn	[26]	RW	Enables FIFO mode. 0 = Enables DMA output. 1 = Enables FIFO output If you enable FIFO mode, then the input mode should be DMA input and WSWP bit at WINCON0 (0xF800_0020) in LCD controller should be set to 0. The FIFO output mode format is YCbCr4:4:4 3plane or RGB 24-bit. Its selection depends on OutFormat register. OutFormat = RGB → RGB 24-bit. Other setting means YCbCr4:4:4. If you want to set interlace out and DMA input together, then output mode should be FIFO output. NOTE: From a performance perspective, FIFO output mode has some limitations than DMA output. Only CSC (Color Space Conversion) without scaling is possible. (ML = OO)	0
Interlace	[25]	RW	Output scan method selection register. (RAW and JPEG input formats are not available) 0 = Selects progressive scan out. 1 = Selects interlace scan out (Input data should be in the form of progressive data). NOTE: If you configure this bit by 0 for interlaced input, then output is also interlaced format not converted into progressive. (ML = OX)	0
MainHorRatio	[24:16]	RW	Specifies horizontal scale ratio for main-scalar. For more information, refer to *42.8.1.49* gathering extension register (MainHorRatio_ext). (ML = OO)	0
ScalerStart	[15]	RW	Specifies the Scalar start. 0 = Enables scaler stop or scalar bypass. 1 = Enables scalar start. (ML = OO)	0
InRGB_FMT	[14:13]	RW	Specifies DMA input RGB format. 00 = Selects RGB565 format. 01 = Selects RGB666 format. 10 = Selects RGB888 format. 11 = Reserved (ML = OX)	0
OutRGB_FMT	[12:11]	RW	Specifies DMA output RGB format. 00 = Selects RGB 16-bit (RGB565, ARGB1555, ARGB4444) format 01 = Selects RGB666 format 10 = Selects ARGB888 format	0

Name	Bit	Type	Description	Reset Value
			11 = Reserved (For more information, refer RGB 16-bit selection format register → RGB16bFMT) (ML = OO)	
Ext_RGB	[10]	RW	Specifies RGB input data extension enable bit for conversion of RGB565/666 mode to RGB888 mode. 0 = Enables normal. 1 = Enables extension. • Input R = 5-bit in RGB565 mode 10100 → 10100101 (Extension): [7] = [2], [6] = [1], [5] = [0] 10100 → 10100000 (Normal) • Input R = 6-bit in RGB666 mode 101100 → 10110010 (Extension): [7] = [1], [6] = [0] 101100 → 10110000 (Normal) (ML = OO)	0
One2One	[9]	RW	Scaler does not run interpolation, but runs repetition for upsampling. Sometimes other IP needs this method if input format are YCbCr4:2:0 and YCbCr4:2:2. Scalar bypass should be set to "0" and YUV plane can be set to anything. Caution: If input/output format and size are same, then use One2One. One2One function has size constraints, as described in *Table 42-1* For example: Input YCbCr4:2:0 2plane → output YCbCr4:2:0 3plane (O.K) (ML = OO)	0
MainVerRatio	[8:0]	RW	Specifies vertical scale ratio for main-scalar. For more information, refer to the gathering extension register (MainVerRatio_ext) (ML = OO)	0

Table 42-6 describes the color space equations.

Table 42-6 Color Space Conversion Equations

–	Wide	Narrow
CSCY2R (601)	R = Y + 1.371 (Cr-128) G = Y – 0.698 (Cr-128) – 0.336 (Cb-128) B = Y + 1.732 (Cb-128)	R = 1.164 (Y-16) + 1.596 (Cr-128) G = 1.164 (Y-16) – 0.813 (Cr-128) – 0.391 (Cb-128) B = 1.164 (Y-16) + 2.018 (Cb-128)
CSCY2R (709)	R = Y + 1.540 (Cr-128) G = Y – 0.459 (Cr-128) – 0.183 (Cb-128) B = Y + 1.816 (Cb-128)	R = 1.164 (Y-16) + 1.793 (Cr-128) G = 1.164 (Y-16) – 0.534 (Cr-128) – 0.213(Cb-128) B = 1.164 (Y-16) + 2.115 (Cb-128)
CSCR2Y (601)	Y = 0.299R + 0.587G + 0.114B Cb = – 0.172R – 0.339G + 0.511B + 128 Cr = 0.511R – 0.428G – 0.083B + 128	Y = 0.257R + 0.504G + 0.098B + 16 Cb = – 0.148R – 0.291G + 0.439B + 128 Cr = 0.439R – 0.368G – 0.071B +128
CSCR2Y (709)	Y = 0.213R + 0.715G + 0.072B Cb = – 0.117R – 0.394G + 0.511B +128 Cr = 0.511R – 0.464G – 0.047B + 128	Y = 0.183R + 0.614G + 0.062B + 16 Cb = – 0.101R – 0.338G + 0.439B + 128 Cr = 0.439R – 0.399G – 0.040B +128

DMA Mode Operation (Normal Mode): DMA Input → DMA Output

The source image formats are:

- YCbCr420
- YCbCr422
- YCbCr444
- RGB16-/18-/24-bit.

The destination image formats are:

- YCbCr420
- YCbCr422
- YCbCr444
- RGB 16-/18-/24-bit.

(Input and output format are possible for Progressive and Interlace format).

You need store all source and destination image data in memory system aligned with double word boundary and should support DMA operation. Therefore, the width of source and destination image should satisfy the double word boundary condition.

FIFO Mode Operation: DMA Input → FIFO Output

FIFO Mode (LCDPathEn = 1) and DMA Mode has two types of color space conversion. They are:

- RGB2YCbCr
- YCbCr2RGB.

This is similar to DMA mode operation. You can transfer destination image to FIFO in display controller (or some other IP with FIFO interface) without additional memory bandwidth such as CAMIF-to-Memory and Memory-to-Display Controller. Output format register determines output data format. OutFormat = RGB format (24-bit RGB) or OutFormat = YCbCr format (YCbCr444). *Table 42-7* describes image format and destination image format restrictions.

Table 42-7 describes the FIFO mode image format.

Table 42-7 FIFO Mode Image Format

DMA input (Progressive/Interlace)		FIFO output (Progressive/Interlace)
YCbCr	YCbCr420: 3/2 plane	YCbCr 444 3 plane or RGB 24-bit
	YCbCr422: 3/2/1 plane	
	YCbCr444: 3/2 plane	
RGB	RGB 16-/18-/24-bit	

In FIFO mode (LCDPathEnable = 1), you can select either progressive or interlace scan mode based on the interlace control register and register files lists.

The interlace control bit is available if LCDPathEn = 1, otherwise DMA mode operation does not affect its value, which supports only progressive scan mode. Even if you enable an interlaced scan mode (LCDPathEn = 1 and Interlace = 1), per frame management which consists of even and odd field is automatic. This means that user interruption is unnecessary to interfield switching in the same frame. Therefore, the frame management scheme is identical for progressive and interlaced scan modes. Interlace does not support if camera processor selects the input data.

Figure 42-26 illustrates the I/O timing diagram for direct path.

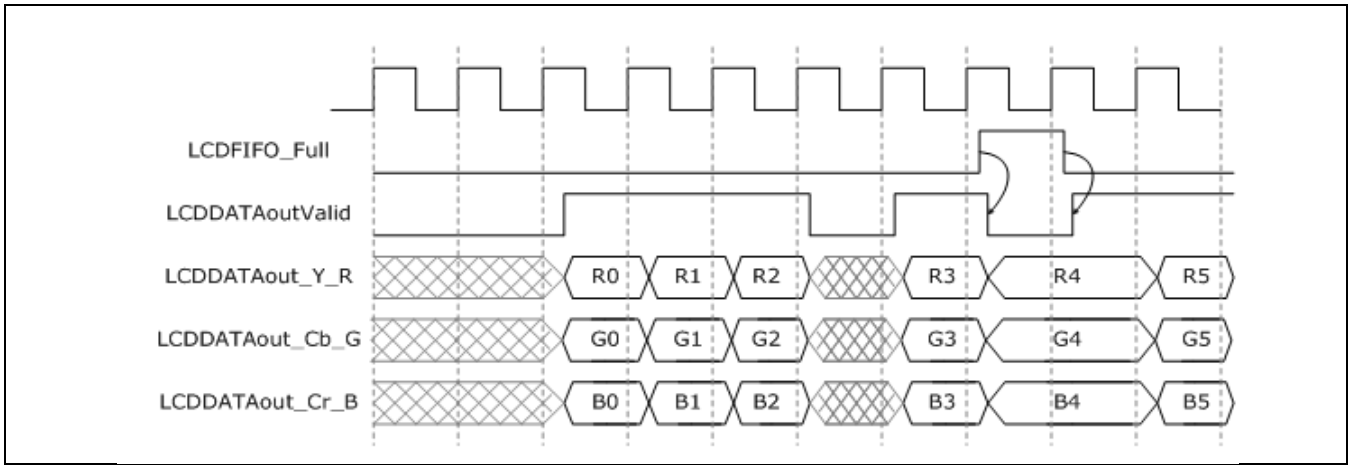


Figure 42-26 I/O Timing Diagram for Direct Path

Figure 42-27 illustrates the input output modes in CAMI.

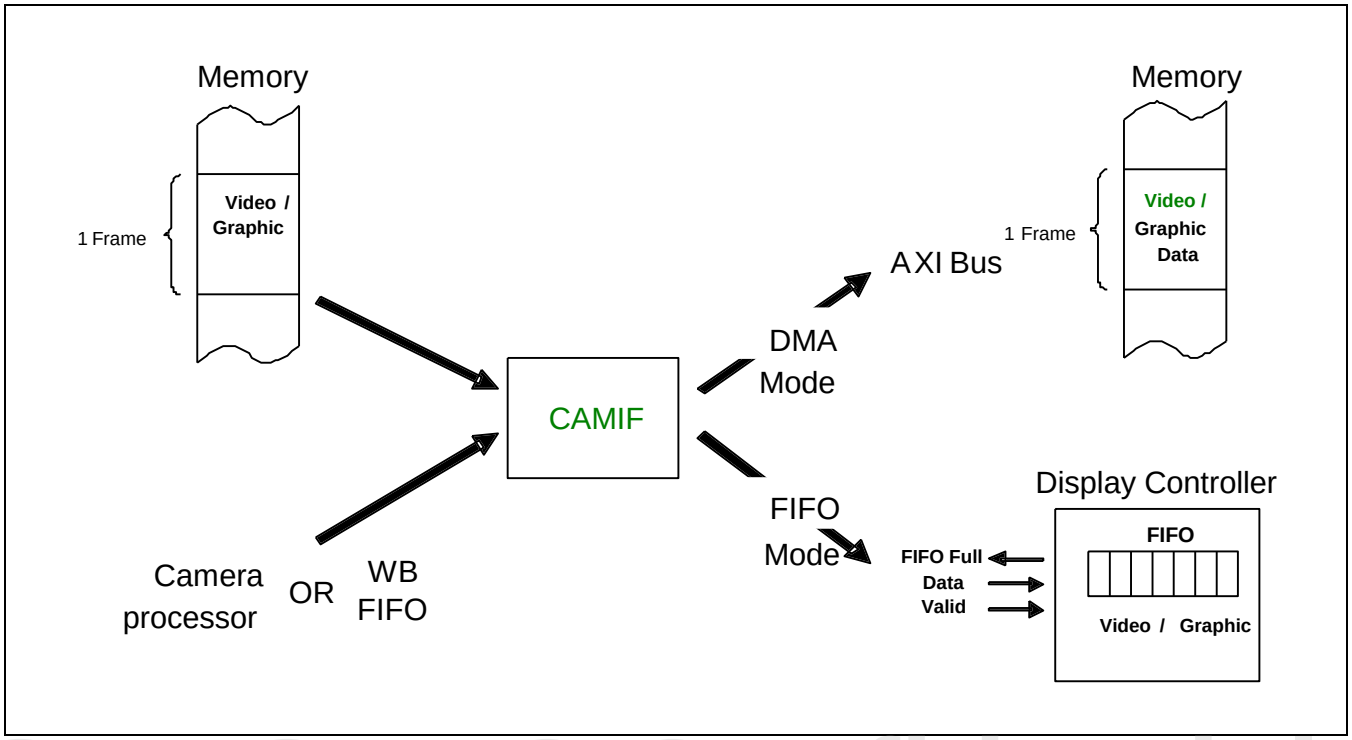


Figure 42-27 Input & Output Modes in CAMIF

42.8.1.22 CITAREAn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x005C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0
CITAREA	[27:0]	RW	Specifies target area for output DMA. Target area = Target H size × Target V size. (ML = OO)	0

42.8.1.23 CIOLINESKIPn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0
OLINESKIP_Cr	[23:20]	RW	Specifies Cr line skip for output DMA. If OLINESKIP_Cr is k, then It stores Cr line in every k + 1 line. NOTE: Maximum value is 8. (ML = OO)	0
RSVD	[19:14]	–	Reserved	0
OLINESKIP_Cb	[13:10]	RW	Specifies Cb line skip for output DMA. If OLINESKIP_Cb is k, then it stores Cb line in every k + 1 line. NOTE: Maximum value is 8. (ML = OO)	0
RSVD	[9:4]	–	Reserved	0
OLINESKIP_Y	[3:0]	RW	Specifies Y line skip for output DMA. If OLINESKIP_Y is k, then it stores Cb line in every k + 1 line. NOTE: Maximum value is 8. (ML = OO)	0

42.8.1.24 CISTATUSn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0064, Reset Value = 0x0000_0800

Name	Bit	Type	Description	Reset Value
OvFiY	[31]	R	Specifies overflow state of Y FIFO. (ML = XX)	0
OvFiCb	[30]	R	Specifies overflow state of Cb FIFO. (ML = XX)	0
OvFiCr	[29]	R	Specifies overflow state of Cr FIFO. (ML = XX)	0
VSYNC	[28]	R	Specifies camera VSYNC (CPU refers this bit for first SFR setting after external camera muxing. It is seen in the ITU-R BT 656 mode). (ML = XX)	0
RSVD	[27]	–	Reserved	0
ScalerStart	[26]	R	Specifies the Scalar start. (After shadowed) (ML = XX)	0
WinOfstEn	[25]	R	Specifies window offset enable status. (ML = XX)	0
FlipMd	[24:23]	R	Specifies flip mode of DMA output. (ML = XX)	0
ImgCptEn	[22]	R	Specifies image capture enable of global camera interface. (After shadowed) (ML = XX)	0
ImgCptEn_SC	[21]	R	Specifies image capture enable of scalar path. (After shadowed) (ML = XX)	0
VSYNC_A	[20]	R	Specifies external camera A VSYNC. Does not adopt polarity inversion (ML = XX)	X
VSYNC_B	[19]	R	Specifies external camera B VSYNC. Polarity inversion is not adopted. (ML = XX)	X
OvRLB	[18]	R	Specifies overflow status of line buffer for rotation. (ML = XX)	0
FrameEnd	[17]	R/W	If frame operation finishes, then it generates FrameEnd and clears the FrameEnd status by setting to 0. (ML = XX)	0
LastCaptureEnd	[16]	R/W	Specifies last frame capture status. Clears the LastCaptureEnd status by setting to 0. You can apply this signal only by camera input mode.	0

Name	Bit	Type	Description	Reset Value
			(ML = XX)	
VVALID_A	[15]	R	Specifies external camera A VVALID. (ML = XX)	X
VVALID_B	[14]	R	Specifies external camera B VVALID. (ML = XX)	X
IRQ_CAM	[13]	R	Specifies interrupt status for camera input mode. (ML = XX)	0
IRQ_DMAend	[12]	R	Specifies interrupt status for DMA frame end in DMA input mode. (ML = XX)	0
FrameCptStatus	[11]	R	Specifies capture frame control status. 0 = Disables present capture 1 = Enables present capture (ML = XX)	1
FrameFieldStatus	[10]	R	Specifies ITU camera field status and internal value after inverse polarity. 0 = Enables present frame Field0 1 = Enables present frame Field1 (ML = XX)	0
LCD_ENSTATUS	[9]	R	Specifies LCD controller enable status. 0 = Disables LCD controller 1 = Enables LCD controller (ML = XX)	0
ENVID_STATUS	[8]	R	Specifies DMA input enable internal status. Sometimes you can use this status to check whether the software completely clears ENVID bit or not. 0 = Disables DMA input operation 1 = Enables DMA input operation	
RSVD	[7:0]	–	Reserved	0

42.8.1.25 CISTATUS2n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0068, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	0
FrameCnt_before	[12:7]	R	Specifies frame count before DMA output. This counter value means the last frame (an accomplished) position number. i.e.) Value = 0= Initial value after reset. Value = 1= 1st frame buffer finish. Value = 2= 2nd frame buffer finish. (ML = XX)	0
RSVD	[6]	–	Reserved	0
FrameCnt_present	[5:0]	R	Specifies frame count present of DMA output. This counter value means the last frame (an accomplished) position number i.e.) Value = 0= Initial value after reset Value = 1= 1st frame buffer under operation Value = 2= 2nd frame buffer under operation (ML = XX)	0

42.8.1.26 CIIMGCPTn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ImgCptEn	[31]	RW	Enables camera interface global capture. (ML = XO)	0
ImgCptEn_Sc	[30]	RW	Enables capture for scalar. This bit should be zero in scalar-bypass mode. (ML = OO)	0
RSVD	[29:26]	–	Reserved	0
Cpt_FrEn	[25]	RW	Controls frame capture (applies only camera progressive input.). 0 = Disables (Free Run mode) 1 = Enables (Step-by-Step frame one shot mode) NOTE: You should configure 0 for user-defined packet or CAM_JPEG mode. You should not change this bit under capture enable status. (ML=XO)	0
RSVD	[24]	–	Reserved	0
Cpt_FrPtr	[23:19]	RW	Captures sequence turn around pointer. (ML = XX)	0
Cpt_ FrMod	[18]	RW	Captures frame control mode. 0 = Apply Cpt_FrEn mode (Captures frames with Cpt_FrSeq when Cpt_FrEn is high. This sequence repeats until you disable capture frame control.) 1 = Applies Cpt_FrCnt mode (Captures Cpt_FrCnt frames with Cpt_FrSeq, after enabling capture DMA frame control. If Cpt_FrCnt = 0, then capture ends.) (ML = XX)	0
Cpt_ FrCnt	[17:10]	RW	Specifies number of frames to be captured. If register reads, then you can see the value of a shadow register, which is downcounted if a frame is captured. In other words, Cpt_FrCnt loads an initial value after capturing the frame. NOTE: You have to disable and enable Cpt_FrEn register before CAMIF starts to use this (capture frame count) function. (ML = XX)	0
RSVD	[9:0]	–	Reserved	0

42.8.1.27 CICPTSEQn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x00C4, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
Cpt_FrSeq	[31:0]	RW	Specifies capture sequence pattern. This register is valid if Cpt_FrEn has a high value. (ML = XX)	FFFF_FFFF

Figure 42-28 illustrates the capture frame control.

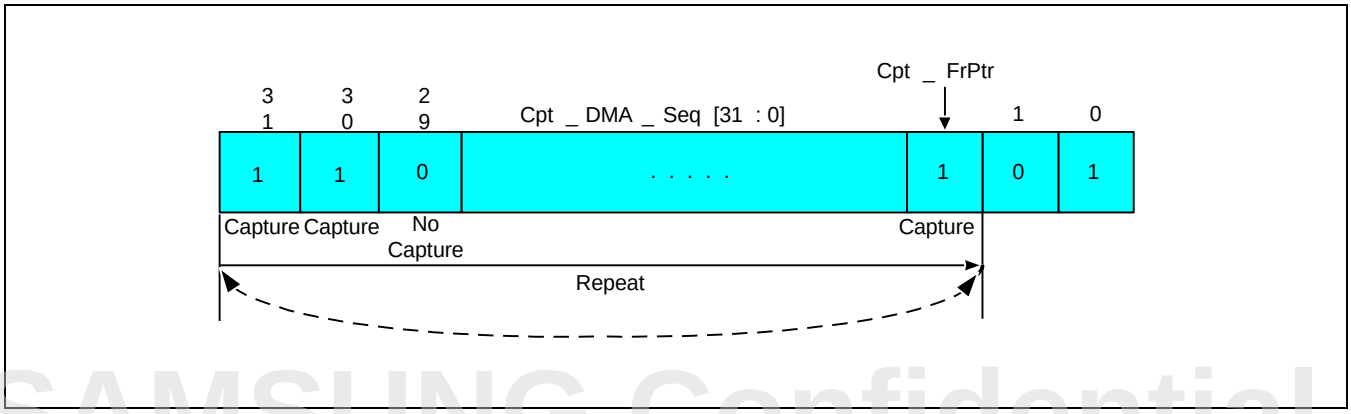


Figure 42-28 Capture Frame Control

NOTE: For skipped frames, it does not generate IRQ and does not increase FrameCnt.

42.8.1.28 CITHOLDn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x00C8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
R_QoS_EN	[31]	RW	0 = Disables Read DMA QoS channel buffer 1 = Enables Read DMA QoS channel buffer You cannot set Read DMA QoS and Write DMA QoS status as enable simultaneously. You can only enable one QoS at a time. (ML = XO)	0
W_QoS_EN	[30]	RW	0 = Disables Write DMA QoS channel buffer 1 = Enables Write DMA QoS channel buffer You cannot set Read DMA QoS and Write DMA QoS status as enable simultaneously. You can only enable one QoS at a time. (ML = XO)	0
RSVD	[29:23]	–	Reserved	0
RTh_QoS	[22:16]	RW	Reads buffer threshold register.(related Input DMA) If Rth_QoS ≥ buffer write count, then it generates read channel. No margin signal for bus high performance. (ML = XO)	0
RSVD	[15:7]	–	Reserved	0
WTh_QoS	[6:0]	RW	Writes buffer threshold register.(related Output DMA) If Wth_QoS < buffer write count, then it generates write channel. No margin signal for bus high performance. (ML = XO)	0

42.8.1.29 CIIMGEFFn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x00D0, Reset Value = 0x0010_0080

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
IE_ON	[30]	RW	0 = Disables image effect function 1 = Enables image effect function (ML = OO)	0
IE_AFTER_SC	[29]	RW	Specifies image effect location. 0 = Specifies image effect location before scaling (You can apply only ITU camera image). 1 = Specifies image effect location after scaling (You can apply camera, write back mode and input DMA image except scalar bypass mode). It applies image effect, even if it is in scalar bypass mode. (ML = OO)	0
FIN	[28:26]	RW	Specifies image effect selection. 3'd0 = Selects bypass effect. 3'd1 = Selects arbitrary Cb/Cr effect. 3'd2 = Selects negative effect. 3'd3 = Selects art Freeze effect. 3'd4 = Selects embossing effect. 3'd5 = Selects silhouette effect. (ML = OO)	0
RSVD	[25:21]	–	Reserved	0
PAT_Cb	[20:13]	RW	You can use this bit only for FIN is Arbitrary Cb/Cr (PAT_Cb/Cr = 8'd128 for Grayscale) Wide CSC Range = 0 < PAT_Cb < 255 Narrow CSC Range = 16 ≤ PAT_Cb ≤ 240 (ML = OO)	8'd128
RSVD	[12:8]	–	Reserved	0
PAT_Cr	[7:0]	RW	You can use this bit only for FIN is Arbitrary Cb/Cr (PAT_Cb/Cr = 8'd128 for Grayscale) Wide CSC Range =0 < PAT_Cr < 255 Narrow CSC Range =16 ≤ PAT_Cr ≤ 240 (ML = OO)	8'd128

NOTE: Cf) sepia: PAT_Cb = 8'd115, PAT_Cr = 8'd145.

Figure 42-29 illustrates the different types of image effects.

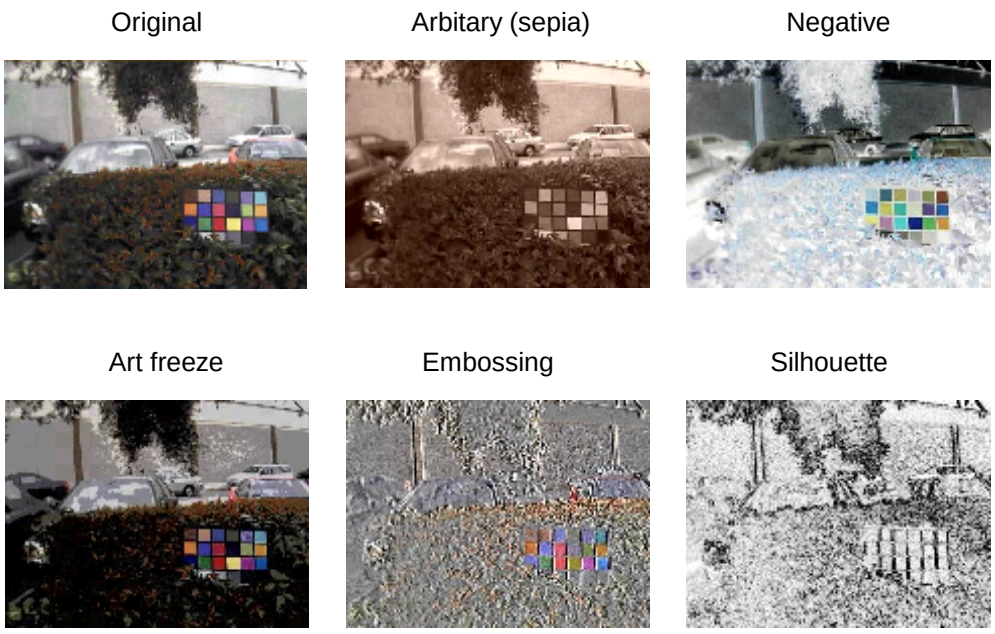


Figure 42-29 Image Effect

42.8.1.30 CIIYSA0n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x00D4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIIYSA0	[31:0]	RW	Input format = YCbCr 2/3 plane → Y frame start address Input format = YCbCr 1 plane → YCbCr frame start address Input format =RGB → RGB frame start address NOTE: In tile mode, align this value 4 KB, that is CIIYSA0[11:0] = 0x000. (ML = OX)	0

42.8.1.31 CIICBSA0n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x00D8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIICBSA0	[31:0]	RW	Input format = YCbCr 3 plane → Cb frame start address Input format = YCbCr 2 plane → CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIICBSA0[11:0] = 0x000. (ML = OX)	0

42.8.1.32 CIICRSA0n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x00DC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIICRSA0	[31:0]	RW	Input format = YCbCr 3 plane → Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIIYSA0[11:0] = 0x000. (ML = OX)	0

42.8.1.33 CIILINESKIP_Yn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x00EC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0
ILINESKIP_Y	[27:24]	RW	Specifies number of Y line skip for input DMA. NOTE: Maximum value is 8 (ML = OX)	0
RSVD	[23:0]	–	Reserved	0

42.8.1.34 CIILINESKIP_CBn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x00F0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0
ILINESKIP_Cb	[27:24]	RW	Specifies number of Cb line skip for input DMA. NOTE: Maximum value is 8. (ML = OX)	0
RSVD	[23:0]	–	Reserved	0

42.8.1.35 CIILINESKIP_CRn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x00F4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	–	Reserved	0
ILINESKIP_Cr	[27:24]	RW	Specifies number of Cr line skip for input DMA. NOTE: Maximum value is 8. (ML = OX)	0
RSVD	[23:0]	–	Reserved	0

42.8.1.36 CIREAL_ISIZEn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x00F8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
AutoLoadEnable	[31]	RW	Restarts input DMA automatically (only software trigger mode). At the start of first frame you should set ENVID_M to start. After the first frame, the next frame does not need ENVID_M setting. If autoload function is running, then you should fix size and format value. 0 = Disables Auto Load 1 = Enables Auto Load (ML = OX)	0
ADDR_CH_DIS	[30]	RW	Disables input DMA address change (only software trigger mode) At the start of first frame, ADDR_CH_DIS should be set equal to "0". 0 = Enables address change 1 = Disables address change (ML = OX)	0
REAL_HEIGHT	[29:16]	RW	Specifies vertical pixel size of input DMA real image. Minimum value is 8. NOTE: 1. 2's multiple =YCbCr 420 2. 4's multiple = Weave-in mode and YCbCr 420 input 3. 2's multiple = Weave-in mode except YCbCr 420 input 4. 2's multiple = Input rotator ON except RGB, YCbCr 444 input Should be multiple of PreVerRatio. If InRot90 = 1, then it should be 16's multiple. It should be 4's multiple of PreHorRatio. Minimum value is 16. (ML = OX)	0
RSVD	[15:14]	–	Reserved	0
REAL_WIDTH	[13:0]	RW	Specifies horizontal pixel size of input DMA real image. NOTE: It should be 16's multiple. It should be 4's multiple of PreHorRatio. Minimum value is 16. If InRot90 = 1, then it should be minimum value of 8. Should be multiple of PreVerRatio. (ML = OX)	0

42.8.1.37 MSCTRLn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x00FC, Reset Value = 0x0400_0000

<table>
<tr><th>Name</th><th>Bit</th><th>Type</th><th>Description</th><th>Reset Value</th></tr>
<tr><td>Weave_in</td><td>[31]</td><td>RW</td><td>The input DMA reads even and odd fields from complete progressive frame. The 1st frame reads even field data and the 2nd frame reads odd field data. Disable Input DMA operation after 1st and 2nd frame operation finishes. Set Weave_in and Interlace_out in simultaneous frames. Also it recommends not to change ping pong address at Interlace even/odd field (BC_SEL field should 0).
0 = Selects Weave mode
1 = Selects Normal mode
NOTE: When using input rotator in Weave_in mode, output horizontal size should be even value. Because vertical data will convert into horizontal one after rotating.
(ML = XX)</td><td>0</td></tr>
<tr><td>RSVD</td><td>[30:28]</td><td>–</td><td>Reserved</td><td>0</td></tr>
<tr><td>Successive_cnt</td><td>[27:24]</td><td>RW</td><td>Specifies input DMA burst successive count (Default is 4 but 3, 2 or 1 are also possible). This value should not be 0.
(ML = OX)</td><td>4’d4</td></tr>
<tr><td>RSVD</td><td>[23:20]</td><td>–</td><td>Reserved</td><td>0</td></tr>
<tr><td>InBuf_Mode</td><td>[19]</td><td>RW</td><td>Specifies input DMA buffer address mode.
0 = Specifies Single buffer mode (Only address 0 is valid)
1 = Specifies Ping-Pong buffer mode (Address 0 and 1 are valid)
(ML = OX)</td><td>0</td></tr>
<tr><td>RSVD</td><td>[18]</td><td>–</td><td>Reserved</td><td>0</td></tr>
<tr><td>Order2p_in</td><td>[17:16]</td><td>RW</td><td>Specifies YCbCr 4:2:0 or 4:2:2 2plane memory reading style order in source input DMA image.
<table>
<tr><th>Bit</th><th>MSB LSB</th></tr>
<tr><td>00</td><td>Cr3Cb3Cr2Cb2Cr1Cb1Cr0Cb0</td></tr>
<tr><td>01</td><td>Cb3Cr3Cb2Cr2Cb1Cr1Cb0Cr0</td></tr>
<tr><td>10</td><td>Reserved</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table>
(ML = OX)</td><td>0</td></tr>
<tr><td>C_INT_IN</td><td>[15]</td><td>RW</td><td>Specifies YCbCr 4:2:0 or 4:2:2 plane
0 = YCbCr 4:2:0 or 4:2:2 3plane input format
1 = YCbCr 4:2:0 or 4:2:2 2plane input format
(ML = OX)</td><td>0</td></tr>
<tr><td>InFlipMd</td><td>[14:13]</td><td>RW</td><td>Specifies image mirror and rotation for input DMA.
00 = Normal
01 = Specifies X-axis mirror</td><td>0</td></tr>
</table>

<table>
<tr><th>Name</th><th>Bit</th><th>Type</th><th>Description</th><th>Reset Value</th></tr>
<tr><td></td><td></td><td></td><td>10 = Specifies Y-axis mirror
11 = Specifies 180° rotation (XY-axis mirror)
(ML = OX)</td><td></td></tr>
<tr><td>FIFO_CTRL</td><td>[12]</td><td>RW</td><td>Specifies a basis FIFO control of input DMA or Input Rotator.
0 = Specifies FIFO Empty (Next burst transaction is possible when FIFO is empty)
1 = Specifies FIFO Full (Next burst transaction is possible except Full FIFO)
(ML = XX)</td><td>0</td></tr>
<tr><td>RSVD</td><td>[11]</td><td>–</td><td>Reserved</td><td>0</td></tr>
<tr><td>BC_SEL</td><td>[10]</td><td>RW</td><td>Selects the input DMA buffer change.
0 = Changes ping pong address at interlace even/ odd field end.
1 = Changes ping pong address at frame operation end.
(ML = OX)</td><td>0</td></tr>
<tr><td>RSVD</td><td>[9]</td><td>–</td><td>Reserved</td><td>0</td></tr>
<tr><td>Buffer_Ptr</td><td>[8]</td><td>RW</td><td>Specifies input DMA buffer address selection pointer. This register initializes the first frame address before starting input DMA.
Do not write this register under frame operation.
0 = Selects Buffer address 0
1 = Selects Buffer address 1
(ML = OX)</td><td>0</td></tr>
<tr><td>RSVD</td><td>[7]</td><td>–</td><td>Reserved</td><td>0</td></tr>
<tr><td>EOF_M</td><td>[6]</td><td>R</td><td>If Input DMA operation is complete, then it generates end of frame.
(ML = OX)</td><td>0</td></tr>
<tr><td>Order422_M</td><td>[5:4]</td><td>RW</td><td>If source input DMA image is 1plane YCbCr 4:2:2, then 1plane YCbCr 4:2:2 inputs memory reading order style.
<table>
<tr><th>Bit</th><th>MSB LSB</th></tr>
<tr><td>00</td><td>Y3Cb1Y2Cr1Y1Cb0Y0Cr0</td></tr>
<tr><td>01</td><td>Cb1Y3Cr1Y2Cb0Y1Cr0Y0</td></tr>
<tr><td>10</td><td>Y3Cr1Y2Cb1Y1Cr0Y0Cb0</td></tr>
<tr><td>11</td><td>Cr1Y3Cb1Y2Cr0Y1Cb0Y0</td></tr>
</table>
(ML = OX)</td><td>0</td></tr>
<tr><td>SEL_DMA_CAM</td><td>[3]</td><td>RW</td><td>Selects input data selection.
0 = Selects external camera input path
1 = Selects memory data input path (Input DMA)
(ML = OX)</td><td>0</td></tr>
<tr><td>InFormat_M</td><td>[2:1]</td><td>RW</td><td>Specifies the source image format for input DMA.
00 = Specifies YCbCr 4:2:0 input image format. (2 or 3 plane)</td><td>0</td></tr>
</table>

Name	Bit	Type	Description	Reset Value
			01 = Specifies YCbCr 4:2:2 input image format. (2 or 3 plane) (Refer to 2 or 3 plane format register → C_INT_IN) 10 = Specifies YCbCr 4:2:2 input image format. (1 plane) 11 = Specifies RGB input image format. (Refer to RGB format register → InRGB_FMT) NOTE: For more information, refer to gathering extension register. YCbCr444_IN (ML = OX)	
ENVID_M	[0]	RW	Starts input DMA operation (Software setting triggers low to high). The hardware clears automatically. If data flows from input DMA to local direct FIFO, then the software can clear this bit when LCD_ENSTATUS is "0". SEL_DMA_CAM = 0, ENVID_M don't care (using external camera signal) SEL_DMA_CAM = 1, ENVID_M is set (0 → 1), then Input DMA operation starts (ML = OX)	0

NOTE: ENVID_M SFR should be set at the end. Starting order for using DMA input path.
SEL_DMA_CAM (others SFR setting) → Image Capture Enable and Scaler start SFR setting → ENVID_M SFR setting:

- Cf.) Image Capture Enable SFR is set at the end. Starting order for using Direct FIFO input path.
SEL_DMA_CAM → SelWB_CAMIF (others SFR setting) → Image Capture Enable and Scaler start SFR setting
- Cf.) Image Capture Enable SFR should be set at the end. Starting order for using camera input path.
SEL_DMA_CAM → SelWB_CAMIF → SelCam_CAMIF (others SFR setting) → Image Capture Enable and Scaler start SFR setting

Figure 42-30 illustrates the ENVID_M SFR setting when input DMA start to read memory data.

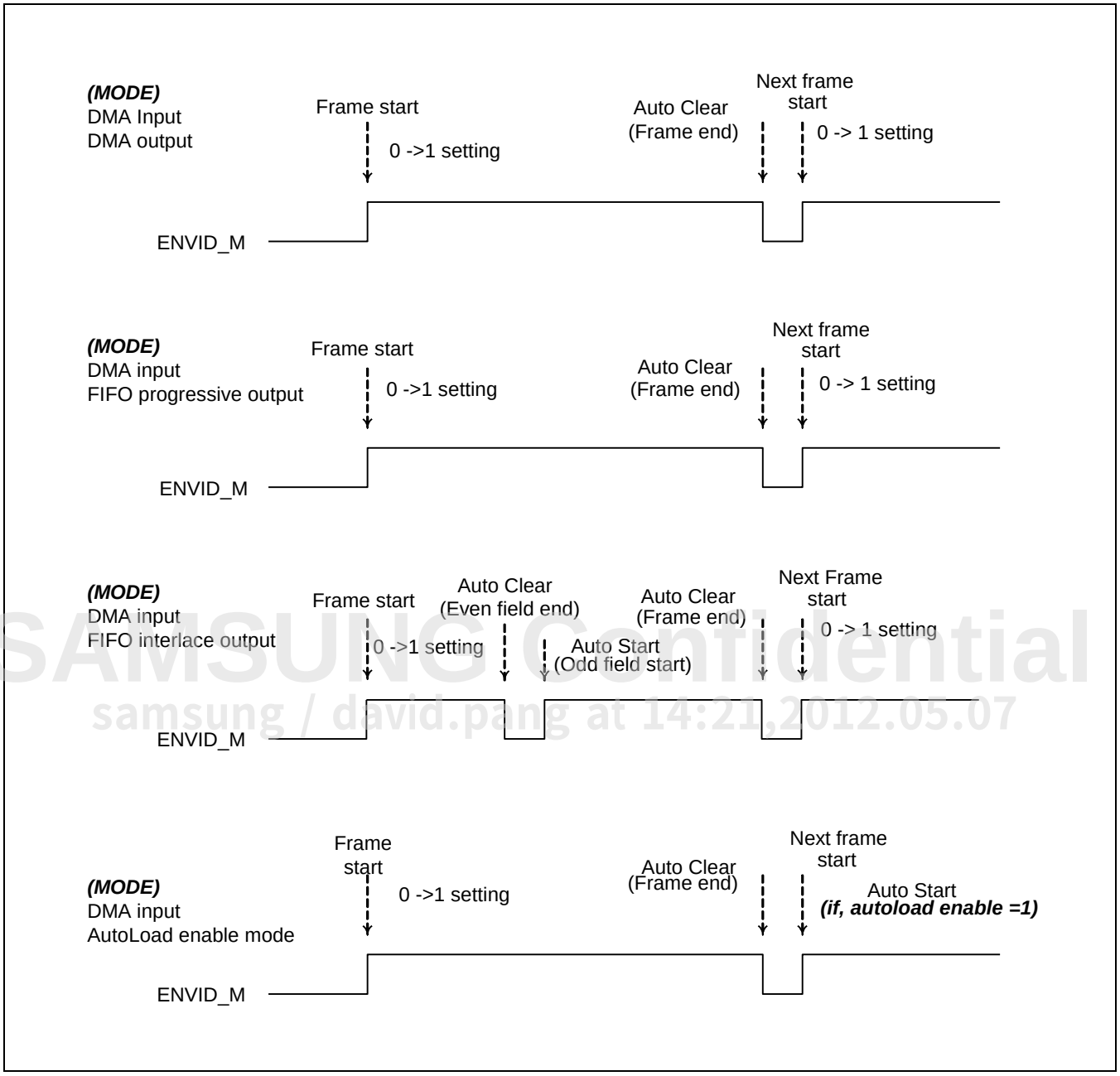


Figure 42-30 ENVID_M SFR Setting When Input DMA Start to Read Memory Data

Figure 42-31 illustrates the SFR and operation.

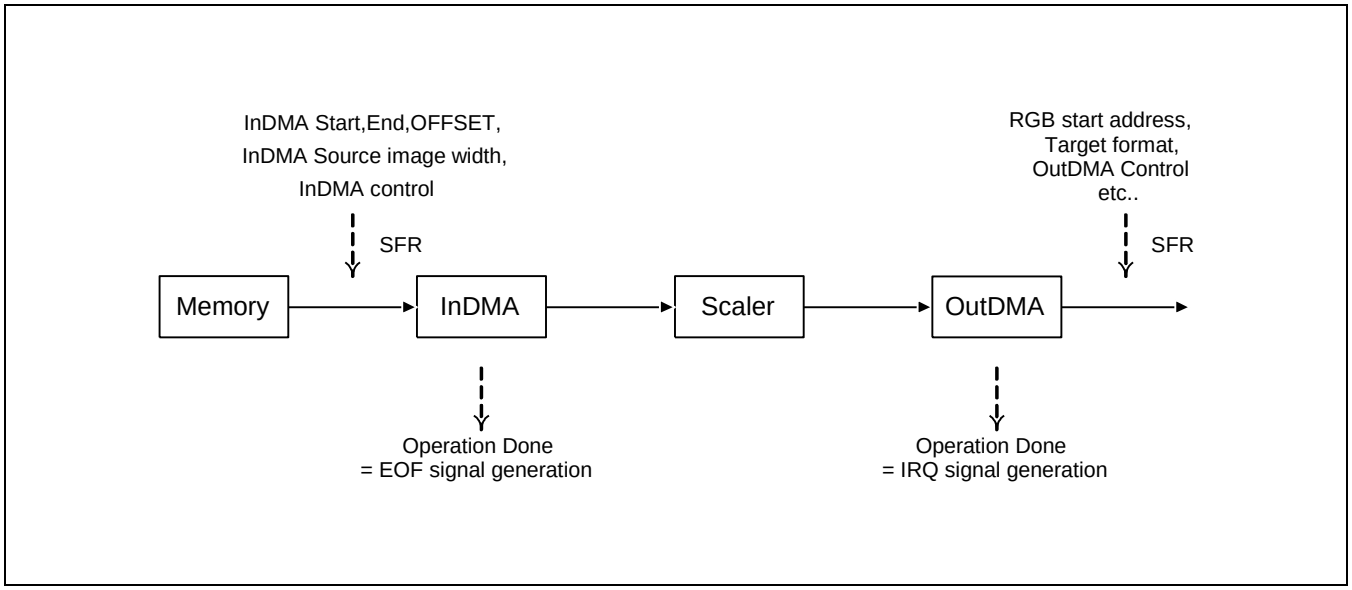


Figure 42-31 SFR and Operation (Related Each DMA When Selected Input DMA Path)

Figure 42-32 illustrates the input DMA address change timing (progressive to progressive).

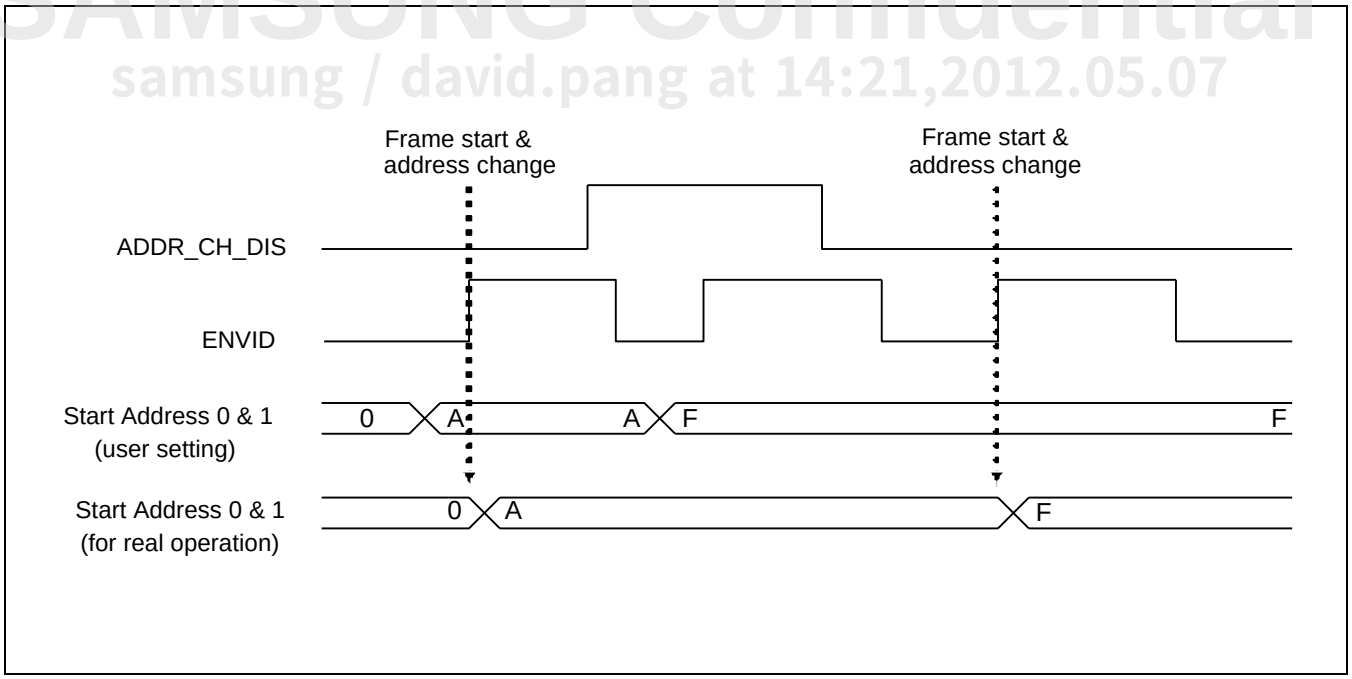


Figure 42-32 Input DMA Address Change Timing (Progressive to Progressive)

Figure 42-33 illustrates the input DMA address change timing (progressive to interlace).

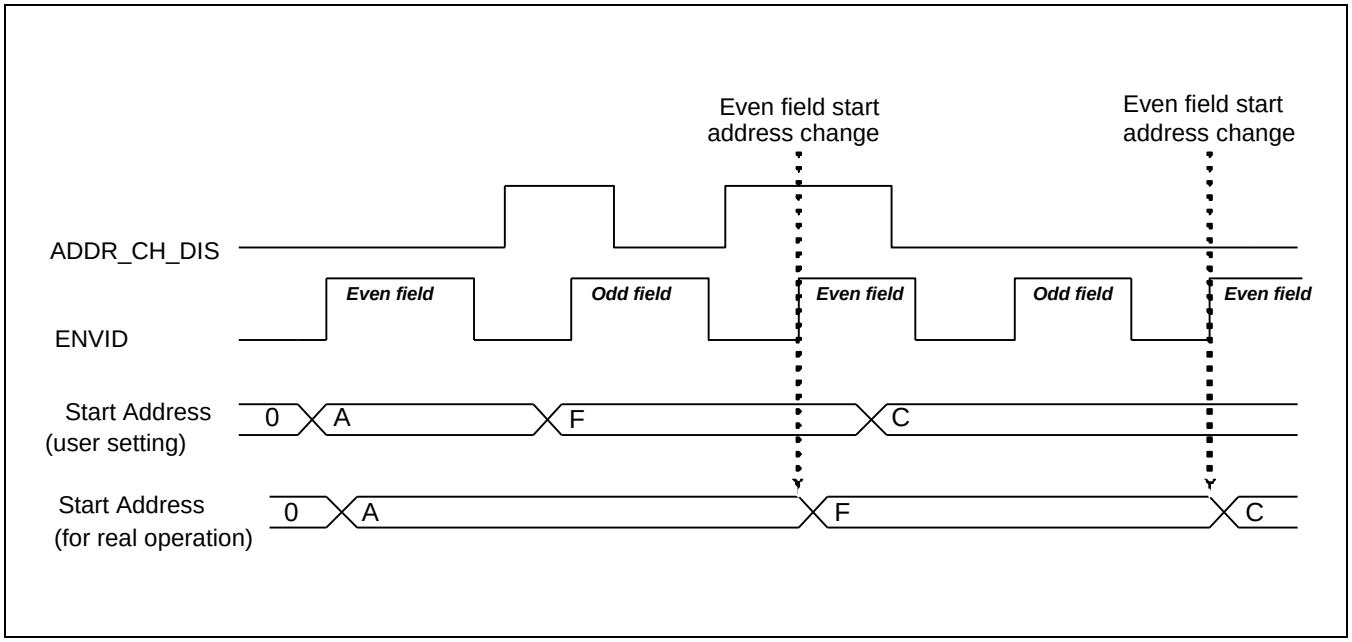


Figure 42-33 Input DMA Address Change Timing (Progressive to Interlace)

Figure 42-34 illustrates the input DMA address change timing (software update).

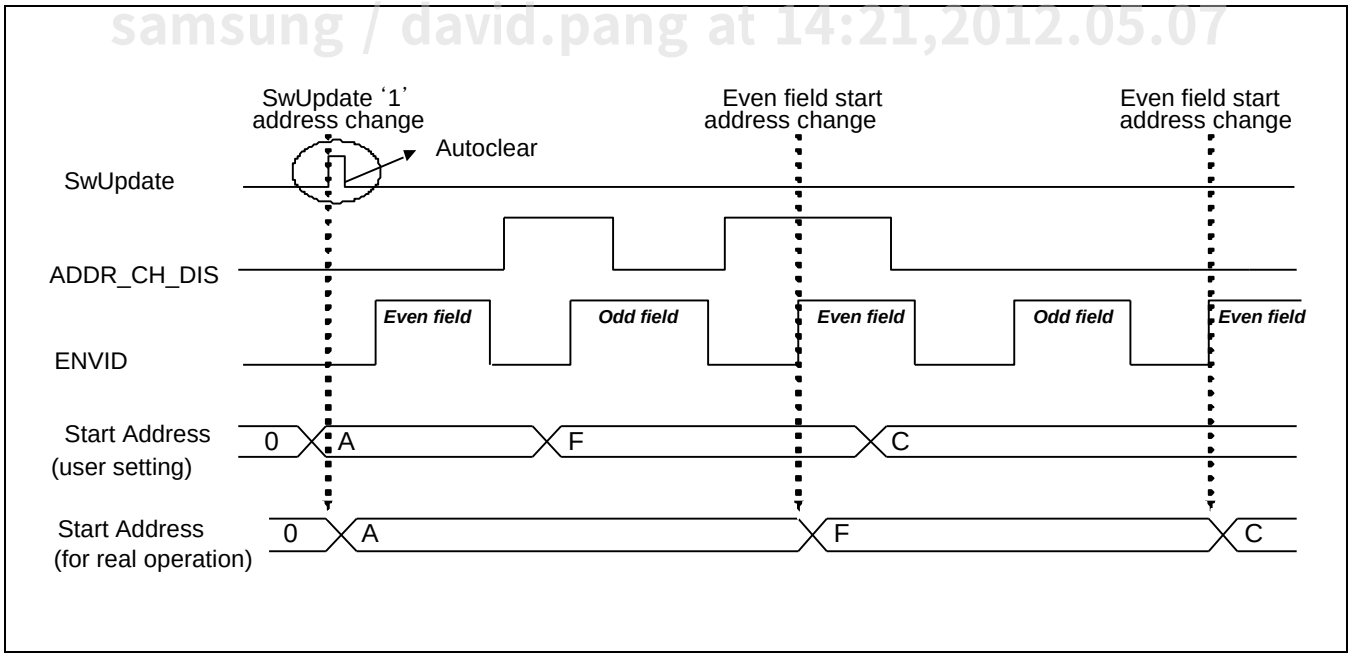


Figure 42-34 Input DMA Address Change Timing (Software Update)

Figure 42-35 illustrates the Input/Output DMA ping pong address change scheme.

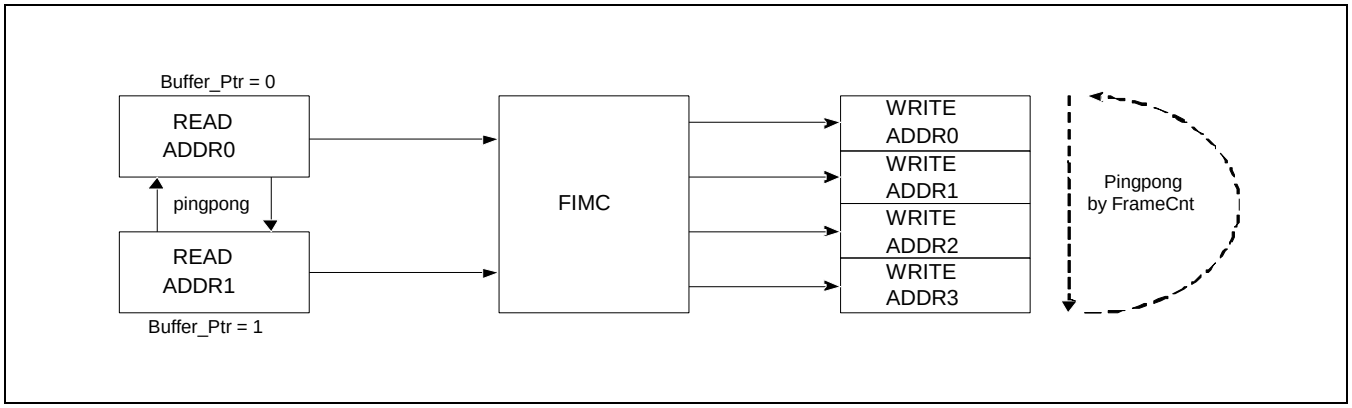


Figure 42-35 Input/Ouput DMA pingpong Address Change Scheme

Figure 42-36 illustrates the input DMA progressive-in to interlace-out (only interlace_out setting).

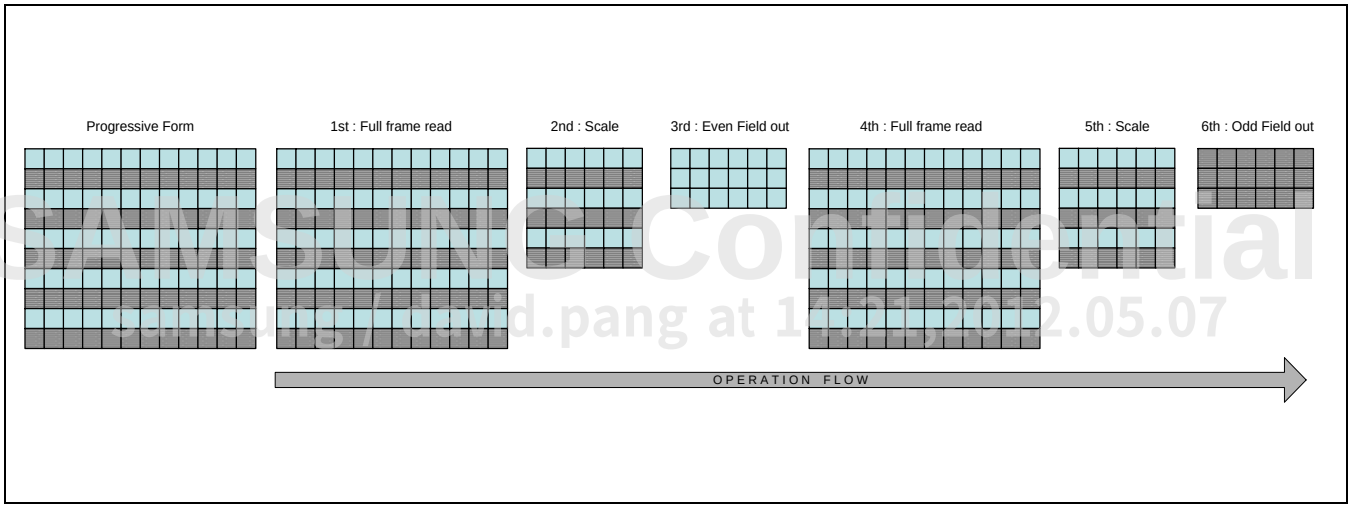


Figure 42-36 Input DMA Progressive-in to Interlace-out (Only Interlace_Out Setting)

Figure 42-37 illustrates the input DMA progressive-in to interlace-out (weave_in and interlace_out setting).

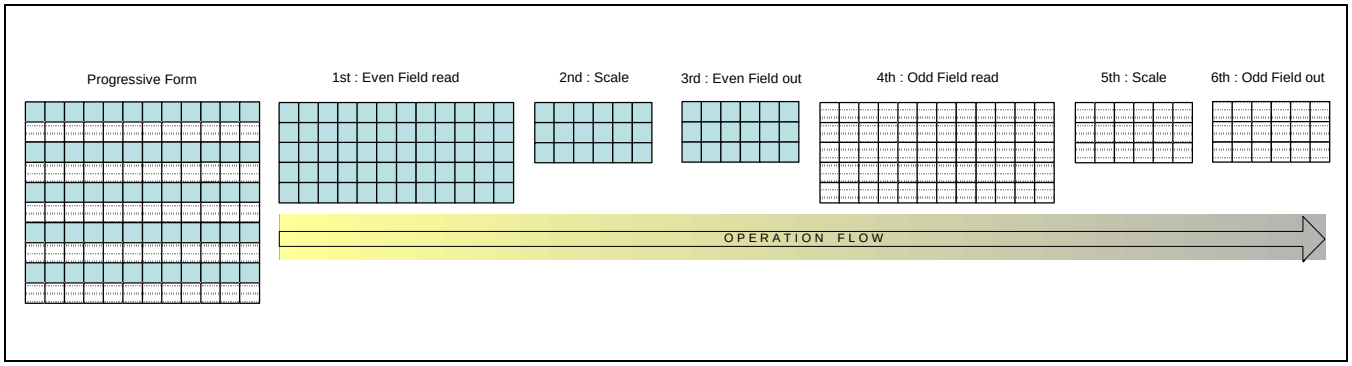


Figure 42-37 Input DMA Progressive-in to Interlace-out (Weave_In and Interlace_Out Setting)

42.8.1.38 CIIYSA1n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0144, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIIYSA1	[31:0]	RW	Input format = YCbCr 2/3 plane → Y frame start address Input format = YCbCr 1 plane → YCbCr frame start address Input format = RGB → RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIIYSA1[11:0] = 0x000. (ML = OX)	0

42.8.1.39 CIICBSA1n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0148, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIICBSA1	[31:0]	RW	Input format = YCbCr 3 plane → Cb frame start address Input format = YCbCr 2 plane → CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIICBSA1[11:0] = 0x000. (ML = OX)	0

42.8.1.40 CIICRSA1n (n = 0 to 3er)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x014C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIICRSA1	[31:0]	RW	Input format=YCbCr 3 plane → Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIICRSA1[11:0] = 0x000. (ML = OX)	0

42.8.1.41 CIOYOFFn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0168, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
OYOFF_V	[29:16]	RW	Output DMA vertical offset for Y component Output format = YCbCr 2/3 plane → Y height offset Output format = YCbCr 1 plane → YCbCr height offset Output format = RGB → RGB height offset NOTE: Offset value is based on line unit. (ML = OO)	0
RSVD	[15:14]	–	Reserved	0
OYOFF_H	[13:0]	RW	Output DMA horizontal offset for Y component Output format = YCbCr 2/3 plane → Y width offset Output format = YCbCr 1 plane → YCbCr width offset Output format = RGB → RGB width offset NOTE: Offset value is based on pixel unit. (ML = OO)	0

42.8.1.42 CIOCBOFFn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x016C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
OCBOFF_V	[29:16]	RW	Output DMA vertical offset for Cb component Output format = YCbCr 3 plane → Cb height offset Output format = YCbCr 2 plane → CbCr height offset NOTE: Offset value is based on line unit (ML = OO)	0
RSVD	[15:14]	–	Reserved	0
OCBOFF_H	[13:0]	RW	Output DMA horizontal offset for Cb component Output format =YCbCr 3 plane → Cb width offset Output format =YCbCr 2 plane → CbCr width offset NOTE: Offset value is based on pixel unit (ML = OO)	0

42.8.1.43 CIOCROFFn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0170, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
OCROFF_V	[29:16]	RW	Specifies output DMA vertical offset for Cr component. Output format = YCbCr 3 plane → Cr height offset NOTE: Offset value is based on line unit (ML = OO)	0
RSVD	[15:14]	–	Reserved	0
OCROFF_H	[13:0]	RW	Specifies output DMA horizontal offset for Cr component. Output format = YCbCr 3 plane → Cr width offset NOTE: Offset value is based on pixel unit (ML = OO)	0

42.8.1.44 CIIYOFFn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0174, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
IYOFF_V	[29:16]	RW	Specifies input DMA vertical offset for Y component. Input format = YCbCr 2/3 plane → Y height offset Input format = YCbCr 1 plane → YCbCr height offset Input format = RGB → RGB height offset NOTE: Offset value is based on line unit (ML = OX)	0
RSVD	[15:14]	–	Reserved	0
IYOFF_H	[13:0]	RW	Specifies input DMA horizontal offset for Y component. Input format = YCbCr 2/3 plane → Y width offset Input format = YCbCr 1 plane → YCbCr width offset Input format = RGB → RGB width offset NOTE: Offset value is based on pixel unit (ML = OX)	0

42.8.1.45 CIICBOFFn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0178, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
ICBOFF_V	[29:16]	RW	Specifies input DMA vertical offset for Cb component. Input format = YCbCr 3 plane → Cb height offset Input format = YCbCr 2 plane → CbCr height offset NOTE: Offset value is based on line unit (ML = OX)	0
RSVD	[15:14]	–	Reserved	0
ICBOFF_H	[13:0]	RW	Specifies input DMA horizontal offset for Cb component. Input format = CbCr 3 plane → Cb width offset Input format = YCbCr 2 plane → CbCr width offset NOTE: Offset value is based on pixel unit (ML = OX)	0

42.8.1.46 CIICROFFn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x017C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
ICROFF_V	[29:16]	RW	Specifies input DMA vertical offset for Cr component. Input format = YCbCr 3 plane → Cr height offset NOTE: Offset value is based on line unit (ML = OX)	0
RSVD	[15:14]	–	Reserved	0
ICROFF_H	[13:0]	RW	Specifies input DMA horizontal offset for Cr component. Input format = YCbCr 3 plane → Cr width offset NOTE: Offset value is based on pixel unit (ML = OX)	0

Figure 42-38 illustrates the input DMA offset and image size.

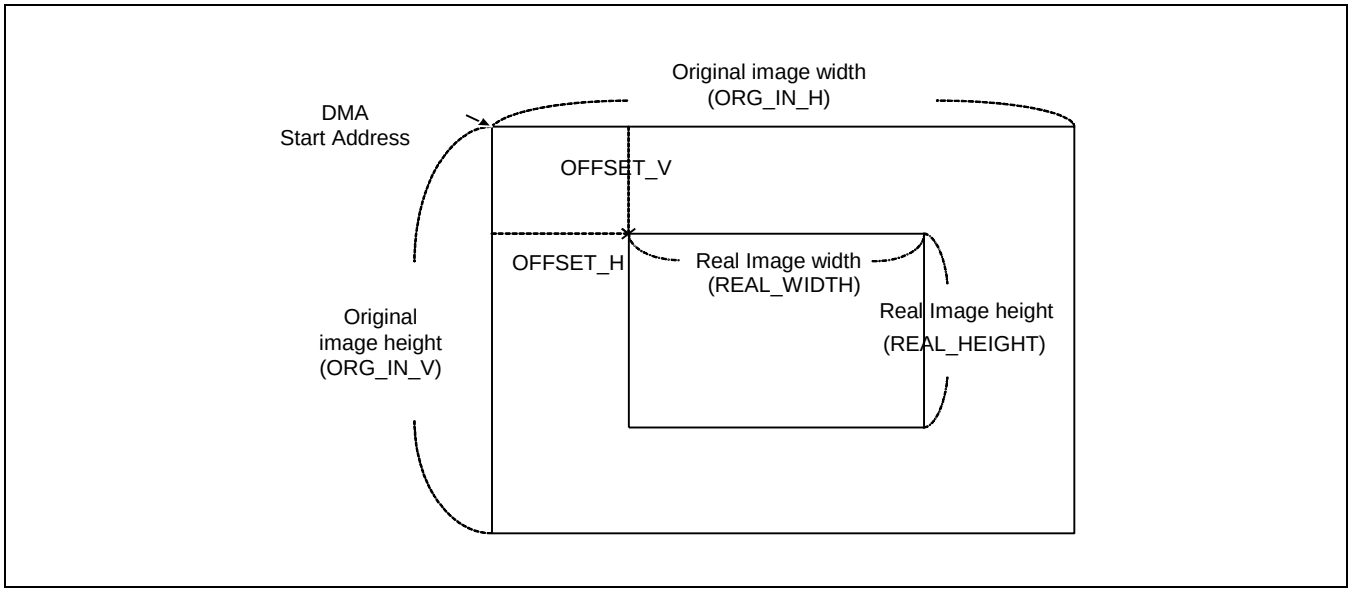


Figure 42-38 Input DMA Offset and Image Size

Figure 42-39 illustrates the output DMA offset and image size.

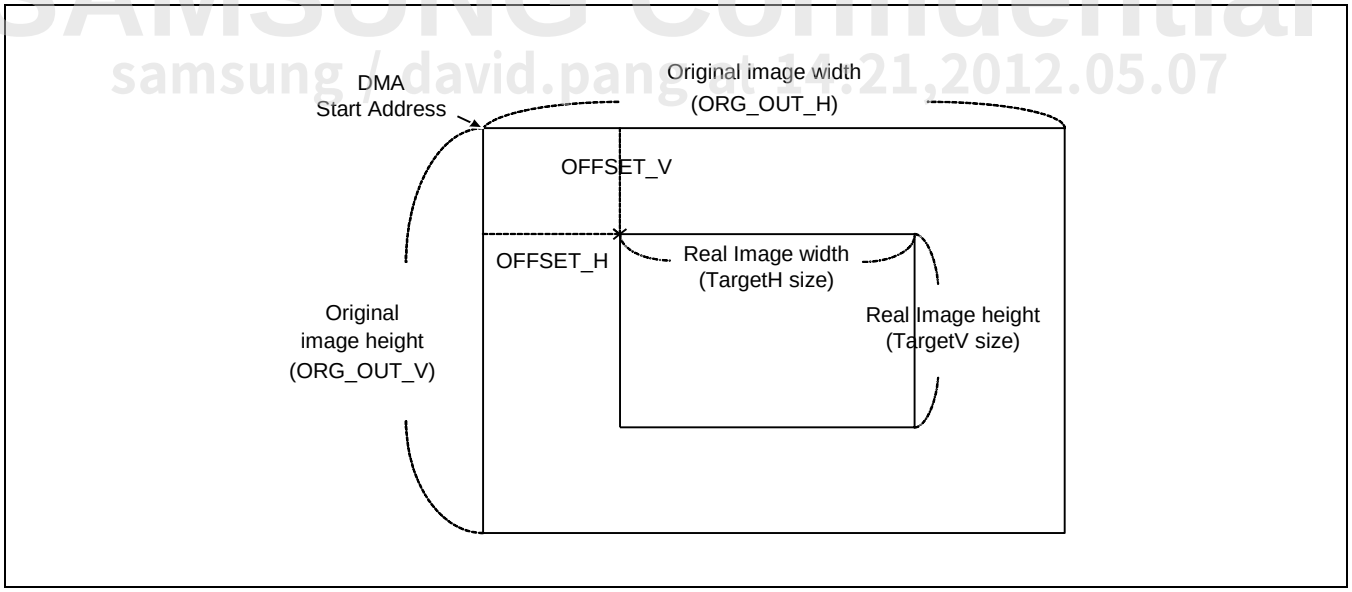


Figure 42-39 Output DMA Offset and Image Size

- **DMA Start Address**
  Start address of ADDRStart_Y/Cb/Cr/RGB points to the first address, where the corresponding component of Y/Cb/Cr/RGB is read or written. ADDRStart_Cb is valid only for two or three planes YCbCr420, 422, 444 source image formats. ADDRStart_Cr is valid only for three planes YCbCr420, 422, 444 source images.

  Each of these should be aligned with double word boundary (That is ADDRStart_X[2:0] = 3'b000).

- **DMA Offset**
  - Offset_H_Y = Y offset per horizontal line = Number of pixel (or sample) in horizontal offset
  - Offset_H_Cb = Cb offset per horizontal line = Number of pixel (or sample) in horizontal offset
  - Offset_H_ Cr = Cr offset per horizontal line = Number of pixel (or sample) in horizontal offset
  - Offset_V_Y = Number of vertical Y offset
  - Offset_V_Cb = Number of vertical Cb offset
  - Offset_V_Cr = Number of vertical Cr offset

42.8.1.47 ORGISIZEn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0180, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
ORG_IN_V	[29:16]	RW	Specifies vertical pixel size of input DMA original image. (Minimum size is 8). This size should not be less than REAL_HEIGHT register. NOTE: When you enable input rotator, this value should be multiple of eight (ML = OX)	0
RSVD	[15:14]	–	Reserved	0
ORG_IN_H	[13:0]	RW	Specifies horizontal pixel size of input DMA source image. Should be multiple of 16. This size should not be less than REAL_WIDTH register. (ML = OX)	0

NOTE: The equation to enable input rotator for memory region of input DMA is:
ORG_IN_V + [8-{(ORG_IN_V - OFFSET_V)%8}]

42.8.1.48 ORGOSIZEn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0184, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
ORG_OUT_V	[29:16]	RW	Specifies vertical pixel size of output DMA original image. (Minimum size is 8). This size should not be less than TargetVsize register. If output rotator is running, then this size should not be less than TargetHsize register. NOTE: If output format is YCbCr 420, then this value should be even number. (ML = OO)	0
RSVD	[15:14]	–	Reserved	0
ORG_OUT_H	[13:0]	RW	Specifies horizontal pixel size of output DMA source image.It is multiple of 16. This size should not be less than TargetHsize register. If output rotator is running, then this size should not be less than TargetVsize register. (ML = OO)	0

42.8.1.49 CIEXTENn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0188, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
SrcHsize_CAM_ext	[30]	RW	Specifies bit value [13] of camera source horizontal pixel number register. {SrcHsize_CAM_ext,SrcHsize_CAM} = {[13], [12:0]}. Thus, total camera source horizontal size = [13:0] (ML = OO)	0
RSVD	[29]	–	Reserved	0
WinHorOfst_ext	[28]	RW	Specifies bit value [11] of window horizontal offset register. {WinHorOfst_ext,WinHorOfst} = {[11], [10:0]}. Thus, total window horizontal offset size = [11:0] (ML = OO)	0
RSVD	[27]	–	Reserved	
TargetHsize_ext	[26]	RW	Specifies bit value [13] of target image horizontal pixel number register. {TargetHsize_ext,TargetHsize} = {[13], [12:0]} Thus, total target image horizontal size = [13:0] (ML = OO)	0
RSVD	[25]	–	Reserved	0
TargetVsize_ext	[24]	RW	Specifies bit value [13] of target image vertical number register. {TargetVsize_ext,TargetVsize} = {[13], [12:0]} Thus, total target image vertical size = [13:0] (ML = OO)	0
RSVD	[23]	–	Reserved	0
YCbCr444_OUT	[22]	RW	If this bit is set to 1, then the output format is YCbCr 444. This register priority is higher than OutFormat in CITRGFMTn (Described in Chapter 8.17). (ML = OO)	0
RSVD	[21]	–	Reserved	0
YCbCr444_IN	[20]	RW	Input DMA format YCbCr4:4:4 (this register priority is higher than InFormat_M register) (ML = OX)	0
RSVD	[19:16]	–	Reserved	0
MainHorRatio_ext	[15:10]	RW	Bit value [5:0] of the Mainscale horizontal ratio register. {MainHorRatio,MainHorRatio_ext} = {[14:6], [5:0]} Thus, total Mainscale horizontal ratio register range = [14:0] (ML = OO)	0

Name	Bit	Type	Description	Reset Value
RSVD	[9:6]	–	Reserved	0
MainVerRatio_ ext	[5:0]	RW	Bit value [5:0] of the Mainscale vertical ratio register. {MainVerRatio,MainVerRatio_ext} = {[14:6], [5:0]} Thus, total Mainscale vertical ratio register range = [14:0] (ML = OO)	0

42.8.1.50 CIDMAPARAMn

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x018C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
MODE_R	[30:29]	RW	Specifies the INPUT DMA address access style. 0 = Specifies Linear style. 1 = Reserved 2 = Reserved 3 = Specifies 64 × 32 tile style. (ML = OX)	0
RSVD	[28:15]	–	Reserved	0
MODE_W	[14:13]	RW	Specifies the OUTPUT DMA address access style. 0 = Specifies Linear style. 1 = Reserved 2 = Reserved 3 = Specifies 64 × 32 tile style. NOTE: If input format is either CAM_JPEG or MIPI RAW, then you cannot use 64 × 32 tile mode. (ML = XX)	0
RSVD	[12:4]	–	Reserved	0
RSVD	[3:0]	–	Reserved	0

NOTE:

1. For more information, refer to Chapter 53, "MFC" for Tile Mode description.
2. Frame End Address calculation method (useful only for TILE 64 × 32 access mode)
3. When tile mode is enable, lower 13 bits of Base_address[31:0] have Zero value. So, SFRs related to Base_adddress should be zero in their lower 13 bits

When condition is YCbCr4:2:0 3plane and in Rotator 90' & X,XY-flip and Horizontal size ≤ 32 and OFFSET_X_Cr ≠ 0, Minimum input size is 64 × 64.

Example 42-1 Image Pixel Size: 720p (1280 × 720), Format: YCbCr4:2:0 2plane (NV12)

```
      * hor_img_size (width) = 1280byte, ver_img_size (height) = 720
      if (hor_img_size % 16 == 0) hor_img_offset = 0
      else  hor_img_offset = 16 - (hor_img_size % 16)
      if (ver_img_size % 16 == 0) ver_img_offset = 0
      elsever_img_offset = 16 - (ver_img_size % 16)
      if (Luma) {//Y plane
      pixel_x = hor_img_size+hor_img_offset = 1280
      pixel_y = ver_img_size+ver_img_offset = 720
      }
      else if (Chroma) {//Cb/Cr plane
      pixel_x = hor_img_size+hor_img_offset = 1280
      pixel_y = (ver_img_size+ver_img_offset)/2 = 360
      }

      1) Luma case
      pixel_x_minus = pixel_x - 1 = 1279
      pixel_y_minus = pixel_y - 1 = 719 = 1011001111 (binary)
      roundup_x = INT (INT ((pixel_x - 1)/16)/8) + 1 = 10
      roundup_y = INT(INT((pixel_y - 1)/16)/4) + 1 = 12

      if (pixel_y_minus[5] == 0) // pixel_y_minus[5:0]= 'b 001111, pixel_y_minus[5]=0
      pic_range = pixel_y_minus[14:6] × roundup_x + pixel_x_minus[14:8] + 1 = 11 × 10 + 4 + 1 = 115
      else
      pic_range = roundup_x × roundup_y

      2) Chroma case
      pixel_x_minus = pixel_x - 1 = 1279
      pixel_y_minus = pixel_y - 1 = 359 = 101100111 (binary)
      roundup_x = INT (INT ((pixel_x - 1)/16)/8) + 1 = 10
      roundup_y = INT(INT((pixel_y - 1)/16)/4) + 1 = 6
      if (pixel_y_minus[5] == 0) // pixel_y_minus[5:0]='b 100111, pixel_y_minus[5]=1
      pic_range = pixel_y_minus[14:6] × roundup_x + pixel_x_minus[14:8] + 1
      else
      pic_range = roundup_x × roundup_y = 10 × 6 = 60
Thus, each plane frame end address = Base_address[31:0] + {pic_range, 2'b0, 11'b0}
```

42.8.1.51 CSIIMGFMTn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0194, Reset Value = 0x0000_001E

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	0
DATAb_port	[9:8]	RW	Specifies the MIPI CSIS data align. 0 = Specifies 24-bit align 1 = Specifies 32-bit align 2 = Reserved 3 = Reserved (ML = XX)	0
RSVD	[7:6]	–	Reserved	0
ImgFormOfCh0	[5:0]	RW	Specifies the image format of MIPI Channel 0. If the RAW format is image format, then the image format conversion is not possible. Set scaler bypass mode. 0x1E = YUV422 8-bit 0x2A = RAW8 0x2B = RAW10 0x2C = RAW12 0x30 to 3F = User defined (ML = XX)	0x1E

42.8.1.52 CIKEYn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x019C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
KEY_DETECT	[31]	RW	Specifies KEY detect for graphic layer scaling. 0 = KEY detect OFF. (Normal) 1 = KEY detect ON (ML = OO)	0
RSVD	[30:28]	–	Reserved	0
R_KEY	[27:20]	RW	Specifies "R" of RGB region key value. (ML = OO)	0
RSVD	[19:18]	–	Reserved	0
G_KEY	[17:10]	RW	Specifies "G" of RGB region key value. (ML = OO)	0
RSVD	[9:8]	–	Reserved	0
B_KEY	[7:0]	RW	Specifies "B" of RGB region key value. (ML = OO)	0

42.8.1.53 CIINTER420n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x01A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[30:2]	–	Reserved	0
O_INTER420	[1:0]	RW	Specifies any Progressive YCbCr or RGB input to Interlace YCbCr 4:2:0 output (included 2 plane and 3 plane) chroma sampling position register. If you do not select YCbCr 4:2:0 output and Interlace output, then this register value is invalid. Also, OutRot90 (Output rotator) should be not set "1". 2'b00 = Chroma 1, 2, 5, 6 … 2'b01 = Chroma 1, 3, 5, 7 … 2'b10 = Chroma interpolation (If you select this mode, PreScaler should limit to maximum vertical ratio and Scalerbypass should be set to "0".) A new PreScaler Vertical ratio is less than or equal to 16 (≤ 16). (ML = XX)	0

Figure 42-40 illustrates the interlace YCbCr 4:2:0 output chroma position

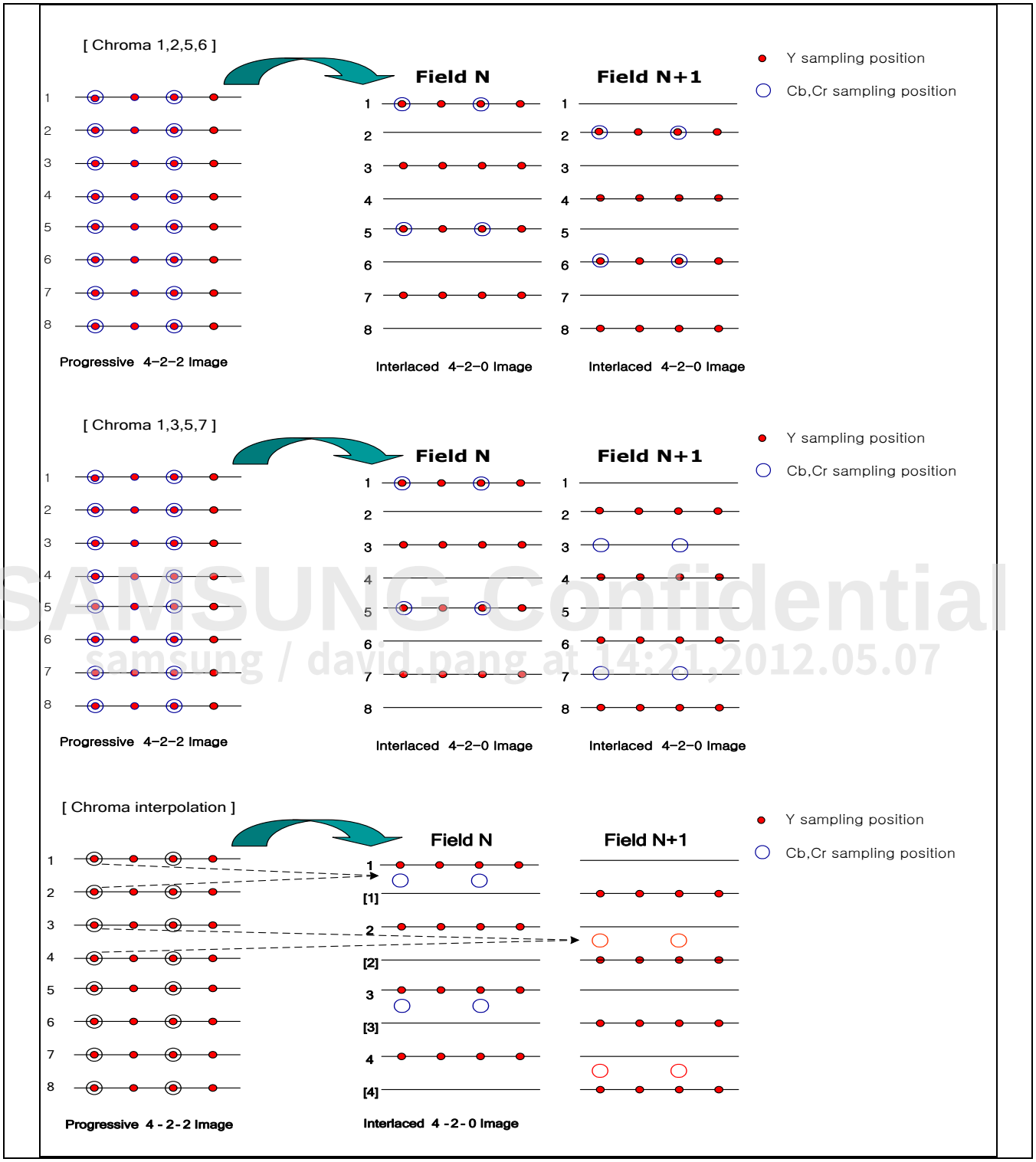


Figure 42-40 Interlace YCbCr 4:2:0 Output Chroma Position

42.8.1.54 CIFCNTSEQn (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x01FC, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
FrameCnt_Seq	[31:0]	RW	Specifies Output frame buffer sequence pattern. Value 0 means "skip frame buffer" and Value 1 means "use frame buffer". At least, Value 1 "use frame buffer" is set greater than 0. (> 0) If Weave out and interlace out function are enable, then skip value should be keep the same value between even and odd frame buffer i.e.) 1st buffer = 1 2nd buffer = 0 (weave out, incorrect setting) i.e.) 3rd buffer = 1 4th buffer = 1 (weave out, correct setting) (ML = OX)	0

42.8.1.55 CIOYSA5n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0200, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA5	[31:0]	RW	Output format = YCbCr 2/3 plane → 5th Y frame start address Output format = YCbCr 1 plane → 5th YCbCr frame start address Output format = RGB → 5th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA5[11:0] = 0x000. (ML = OX)	0

42.8.1.56 CIOYSA6n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0204, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA6	[31:0]	RW	Output format = YCbCr 2/3 plane → 6th Y frame start address Output format = YCbCr 1 plane → 6th YCbCr frame start address Output format = RGB → 6th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA6[11:0] = 0x000. (ML = OX)	0

42.8.1.57 CIOYSA7n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0208, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA7	[31:0]	RW	Output format = YCbCr 2/3 plane → 7th Y frame start address Output format = YCbCr 1 plane → 7th YCbCr frame start address Output format = RGB → RGB 7th frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA7[11:0] = 0x000. (ML = OX)	0

42.8.1.58 CIOYSA8n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x020C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA8	[31:0]	RW	Output format = YCbCr 2/3 plane → 8th Y frame start address Output format = YCbCr 1 plane → 8th YCbCr frame start address Output format = RGB → 8th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA8[11:0] = 0x000. (ML = OX)	0

42.8.1.59 CIOYSA9n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0210, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA9	[31:0]	RW	Output format = YCbCr 2/3 plane → 9th Y frame start address Output format = YCbCr 1 plane → 9th YCbCr frame start address Output format = RGB → 9th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA9[11:0] = 0x000. (ML = OX)	0

42.8.1.60 CIOYSA10n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0214, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA10	[31:0]	RW	Output format = YCbCr 2/3 plane → 10th Y frame start address Output format = YCbCr 1 plane → 10th YCbCr frame start address Output format = RGB → 10th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA10[11:0] = 0x000. (ML = OX)	0

42.8.1.61 CIOYSA11n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0218, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA11	[31:0]	RW	Output format = YCbCr 2/3 plane → 11th Y frame start address Output format = YCbCr 1 plane → 11th YCbCr frame start address Output format = RGB → 11th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA11[11:0] = 0x000. (ML = OX)	0

42.8.1.62 CIOYSA12n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x021C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA12	[31:0]	RW	Output format = YCbCr 2/3 plane → 12th Y frame start address Output format = YCbCr 1 plane → 12th YCbCr frame start address Output format = RGB → 12th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA12[11:0] = 0x000. (ML = OX)	0

42.8.1.63 CIOYSA13n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0220, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA13	[31:0]	RW	Output format = YCbCr 2/3 plane → 13th Y frame start address Output format = YCbCr 1 plane → 13th YCbCr frame start address Output format = RGB → 13th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA13[11:0] = 0x000. (ML = OX)	0

42.8.1.64 CIOYSA14n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0224, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA14	[31:0]	RW	Output format = YCbCr 2/3 plane → 14th Y frame start address Output format = YCbCr 1 plane → 14th YCbCr frame start address Output format = RGB → 14th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA14 [11:0] = 0x000. (ML = OX)	0

42.8.1.65 CIOYSA15n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0228, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA15	[31:0]	RW	Output format =YCbCr 2/3 plane → 15th Y frame start address Output format = YCbCr 1 plane → 15th YCbCr frame start address Output format = RGB → 15th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA15[11:0] = 0x000. (ML = OX)	0

42.8.1.66 CIOYSA16n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x022C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA16	[31:0]	RW	Output format = YCbCr 2/3 plane → 16th Y frame start address Output format = YCbCr 1 plane → 16th YCbCr frame start address Output format = RGB → 16th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA16[11:0] = 0x000. (ML = OX)	0

42.8.1.67 CIOYSA17n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0230, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA17	[31:0]	RW	Output format = YCbCr 2/3 plane → 17th Y frame start address Output format = YCbCr 1 plane → 17th YCbCr frame start address Output format = RGB → 17th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA17[11:0] = 0x000. (ML = OX)	0

42.8.1.68 CIOYSA18n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0234, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA18	[31:0]	RW	Output format = YCbCr 2/3 plane → 18th Y frame start address Output format = YCbCr 1 plane → 18th YCbCr frame start address Output format = RGB → 18th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA18[11:0] = 0x000. (ML = OX)	0

42.8.1.69 CIOYSA18n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0238, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA19	[31:0]	RW	Output format = YCbCr 2/3 plane → 19th Y frame start address Output format = YCbCr 1 plane → 19th YCbCr frame start address Output format = RGB → 19th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA19[11:0] = 0x000. (ML = OX)	0

42.8.1.70 CIOYSA20n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x023C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA20	[31:0]	RW	Output format = YCbCr 2/3 plane → 20th Y frame start address Output format = YCbCr 1 plane → 20th YCbCr frame start address Output format = RGB → 20th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA20[11:0] = 0x000. (ML = OX)	0

42.8.1.71 CIOYSA21n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0240, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA21	[31:0]	RW	Output format = YCbCr 2/3 plane → 21st Y frame start address Output format = YCbCr 1 plane → 21st YCbCr frame start address Output format = RGB → 21st RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA21[11:0] = 0x000. (ML = OX)	0

42.8.1.72 CIOYSA22n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0244, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA22	[31:0]	RW	Output format = YCbCr 2/3 plane → 22nd Y frame start address Output format = YCbCr 1 plane → 22nd YCbCr frame start address Output format = RGB → 22nd RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA22[11:0] = 0x000. (ML = OX)	0

42.8.1.73 CIOYSA23n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0248, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA23	[31:0]	RW	Output format = YCbCr 2/3 plane → 23rd Y frame start address Output format = YCbCr 1 plane → 23rd YCbCr frame start address Output format = RGB → 23rd RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA23[11:0] = 0x000. (ML = OX)	0

42.8.1.74 CIOYSA24n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x024C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA24	[31:0]	RW	Output format = YCbCr 2/3 plane → 24th Y frame start address Output format = YCbCr 1 plane → 24th YCbCr frame start address Output format = RGB → 24th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA24[11:0] = 0x000. (ML = OX)	0

42.8.1.75 CIOYSA25n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0250, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA25	[31:0]	RW	Output format = YCbCr 2/3 plane → 25th Y frame start address Output format = YCbCr 1 plane → 25th YCbCr frame start address Output format = RGB → 25th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA25[11:0] = 0x000. (ML = OX)	0

42.8.1.76 CIOYSA26n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0254, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA26	[31:0]	RW	Output format = YCbCr 2/3 plane → 26th Y frame start address Output format = YCbCr 1 plane → 26th YCbCr frame start address Output format = RGB → 26th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA26[11:0] = 0x000. (ML = OX)	0

42.8.1.77 CIOYSA27n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0258, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA27	[31:0]	RW	Output format = YCbCr 2/3 plane → 27th Y frame start address Output format = YCbCr 1 plane → 27th YCbCr frame start address Output format = RGB → 27th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA27[11:0] = 0x000. (ML = OX)	0

42.8.1.78 CIOYSA28n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x025C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA28	[31:0]	RW	Output format = YCbCr 2/3 plane → 28th Y frame start address Output format = YCbCr 1 plane → 28th YCbCr frame start address Output format = RGB → 28th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA28[11:0] = 0x000. (ML = OX)	0

42.8.1.79 CIOYSA29n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0260, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA29	[31:0]	RW	Output format = YCbCr 2/3 plane → 29th Y frame start address Output format = YCbCr 1 plane → 29th YCbCr frame start address Output format = RGB → 29th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA29[11:0] = 0x000. (ML = OX)	0

42.8.1.80 CIOYSA30n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0264, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA30	[31:0]	RW	Output format = YCbCr 2/3 plane → 30th Y frame start address Output format = YCbCr 1 plane → 30th YCbCr frame start address Output format = RGB → 30th RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA30[11:0] = 0x000. (ML = OX)	0

42.8.1.81 CIOYSA31n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_000000
- Address = Base Address + 0x0268, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA31	[31:0]	RW	Output format = YCbCr 2/3 plane → 31st Y frame start address Output format = YCbCr 1 plane → 31st YCbCr frame start address Output format = RGB → 31st RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA31[11:0] = 0x000 (ML = OX)	0

42.8.1.82 CIOYSA32n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x026C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOYSA32	[31:0]	RW	Output format = YCbCr 2/3 plane → 32nd Y frame start address Output format =YCbCr 1 plane → 32nd YCbCr frame start address Output format = RGB → 32nd RGB frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOYSA32[11:0] = 0x000 (ML = OX)	0

42.8.1.83 CIOCBSA5n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0270, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA5	[31:0]	RW	Output format = YCbCr 3 plane → 5th Cb frame start address Output format = YCbCr 2 plane → 5th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA5[11:0] = 0x000 (ML = OX)	0

42.8.1.84 CIOCBSA6n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0274, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA6	[31:0]	RW	Output format = YCbCr 3 plane → 6th Cb frame start address Output format = YCbCr 2 plane → 6th frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA6[11:0] = 0x000 (ML = OX)	0

42.8.1.85 CIOCBSA7n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0278, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA7	[31:0]	RW	Output format = YCbCr 3 plane → 7th Cb frame start address Output format = YCbCr 2 plane → 7th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA7[11:0] = 0x000 (ML = OX)	0

42.8.1.86 CIOCBSA8n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x027C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA8	[31:0]	RW	Output format = YCbCr 3 plane → 8th Cb frame start address Output format = YCbCr 2 plane → 8th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA8[11:0] = 0x000 (ML = OX)	0

42.8.1.87 CIOCBSA9n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0280, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA9	[31:0]	RW	Output format = YCbCr 3 plane → 9th Cb frame start address Output format = YCbCr 2 plane → 9th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA9[11:0] = 0x000 (ML = OX)	0

42.8.1.88 CIOCBSA10n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0284, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA10	[31:0]	RW	Output format = YCbCr 3 plane → 10th Cb frame start address Output format = YCbCr 2 plane → 10th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA10[11:0] = 0x000 (ML = OX)	0

42.8.1.89 CIOCBSA11n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0288, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA11	[31:0]	RW	Output format = YCbCr 3 plane → 11th Cb frame start address Output format = YCbCr 2 plane → 11th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA11[11:0] = 0x000 (ML = OX)	0

42.8.1.90 CIOCBSA12n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x028C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA12	[31:0]	RW	Output format = YCbCr 3 plane → 12th Cb frame start address Output format = YCbCr 2 plane → 12th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA12[11:0] = 0x000 (ML = OX)	0

42.8.1.91 CIOCBSA13n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0290, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA13	[31:0]	RW	Output format = YCbCr 3 plane → 13th Cb frame start address Output format = YCbCr 2 plane → 13th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA13[11:0] = 0x000. (ML = OX)	0

42.8.1.92 CIOCBSA14n (n = 0 to 3,)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0294, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA14	[31:0]	RW	Output format = YCbCr 3 plane → 14th Cb frame start address Output format = YCbCr 2 plane → 14th frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA14[11:0] = 0x000. (ML = OX)	0

42.8.1.93 CIOCBSA15n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0298, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA15	[31:0]	RW	Output format = YCbCr 3 plane → 15th Cb frame start address Output format =YCbCr 2 plane → 15th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA15[11:0] = 0x000. (ML = OX)	0

42.8.1.94 CIOCBSA16n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x029C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA16	[31:0]	RW	Output format = YCbCr 3 plane → 16th Cb frame start address Output format = YCbCr 2 plane → 16th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA16[11:0] = 0x000. (ML = OX)	0

42.8.1.95 CIOCBSA17n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA17	[31:0]	RW	Output format = YCbCr 3 plane → 17th Cb frame start address Output format = YCbCr 2 plane → 17th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA17[11:0] = 0x000. (ML = OX)	0

42.8.1.96 CIOCBSA18n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02A4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA18	[31:0]	RW	Output format = YCbCr 3 plane → 18th Cb frame start address Output format = YCbCr 2 plane → 18th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA1[11:0] = 0x000. (ML = OX)	0

42.8.1.97 CIOCBSA19n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02A8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA19	[31:0]	RW	Output format = YCbCr 3 plane → 19th Cb frame start address Output format = YCbCr 2 plane → 19th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA19[11:0] = 0x000. (ML = OX)	0

42.8.1.98 CIOCBSA20n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02AC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA20	[31:0]	RW	Output format = YCbCr 3 plane → 20th Cb frame start address Output format = YCbCr 2 plane → 20th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA20[11:0] = 0x000. (ML = OX)	0

42.8.1.99 CIOCBSA21n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02B0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA21	[31:0]	RW	Output format = YCbCr 3 plane → 21st Cb frame start address Output format = YCbCr 2 plane → 21st CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA21[11:0] = 0x000. (ML = OX)	0

42.8.1.100 CIOCBSA22n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02B4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA22	[31:0]	RW	Output format = YCbCr 3 plane → 22th Cb frame start address Output format = YCbCr 2 plane → 22th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA22[11:0] = 0x000. (ML = OX)	0

42.8.1.101 CIOCBSA23n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02B8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA23	[31:0]	RW	Output format = YCbCr 3 plane → 23th Cb frame start address Output format = YCbCr 2 plane → 23th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA23[11:0] = 0x000. (ML = OX)	0

42.8.1.102 CIOCBSA24n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02BC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA24	[31:0]	RW	Output format = YCbCr 3 plane → 24th Cb frame start address Output format = YCbCr 2 plane → 24th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA24[11:0] = 0x000. (ML = OX)	0

42.8.1.103 CIOCBSA25n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02C0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA25	[31:0]	RW	Output format = YCbCr 3 plane → 25th Cb frame start address Output format = YCbCr 2 plane → 25th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA25[11:0] = 0x000 (ML = OX)	0

42.8.1.104 CIOCBSA26n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02C4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA26	[31:0]	RW	Output format = YCbCr 3 plane → 26th Cb frame start address Output format = YCbCr 2 plane → 26th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA26[11:0] = 0x000 (ML = OX)	0

42.8.1.105 CIOCBSA27n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02C8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA27	[31:0]	RW	Output format = YCbCr 3 plane → 27th Cb frame start address Output format = YCbCr 2 plane → 27th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA27[11:0] = 0x000 (ML = OX)	0

42.8.1.106 CIOCBSA28n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02CC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA28	[31:0]	RW	Output format = YCbCr 3 plane → 28th Cb frame start address Output format = YCbCr 2 plane → 28th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA28[11:0] = 0x000. (ML = OX)	0

42.8.1.107 CIOCBSA29n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02D0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA29	[31:0]	RW	Output format = YCbCr 3 plane → 29th Cb frame start address Output format = YCbCr 2 plane → 29th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA29[11:0] = 0x000. (ML = OX)	0

42.8.1.108 CIOCBSA30n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02D4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA30	[31:0]	RW	Output format = YCbCr 3 plane → 30th Cb frame start address Output format =YCbCr 2 plane → 30th CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA30[11:0] = 0x000. (ML = OX)	0

42.8.1.109 CIOCBSA31n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02D8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA31	[31:0]	RW	Output format = YCbCr 3 plane → 31st Cb frame start address Output format = YCbCr 2 plane → 31st CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA31[11:0] = 0x000. (ML = OX)	0

42.8.1.110 CIOCBSA32n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02DC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCBSA32	[31:0]	RW	Output format = YCbCr 3 plane → 32nd Cb frame start address Output format = YCbCr 2 plane → 32nd CbCr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCBSA32[11:0] = 0x000. (ML = OX)	0

42.8.1.111 CIOCRSA5n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02E0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA5	[31:0]	RW	Output format = YCbCr 3 plane → 5th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA5[11:0] = 0x000. (ML = OX)	0

42.8.1.112 CIOCRSA6n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02E4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA6	[31:0]	RW	Output format = YCbCr 3 plane → 6th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA6[11:0] = 0x000 (ML = OX)	0

42.8.1.113 CIOCRSA7n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02E8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA7	[31:0]	RW	Output format = YCbCr 3 plane → 7th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA7[11:0] = 0x000 (ML = OX)	0

42.8.1.114 CIOCRSA8n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02EC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA8	[31:0]	RW	Output format = YCbCr 3 plane → 8th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA8[11:0] = 0x000 (ML = OX)	0

42.8.1.115 CIOCRSA9n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02F0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA9	[31:0]	RW	Output format = YCbCr 3 plane → 9th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA9[11:0] = 0x000 (ML = OX)	0

42.8.1.116 CIOCRSA10n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02F4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA10	[31:0]	RW	Output format = YCbCr 3 plane → 10th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA10[11:0] = 0x000 (ML = OX)	0

42.8.1.117 CIOCRSA11n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02F8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA11	[31:0]	RW	Output format = YCbCr 3 plane → 11th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA11[11:0] = 0x000 (ML = OX)	0

42.8.1.118 CIOCRSA12n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x02FC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA12	[31:0]	RW	Output format = YCbCr 3 plane → 12th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA12[11:0] = 0x000 (ML = OX)	0

42.8.1.119 CIOCRSA13n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0300, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA13	[31:0]	RW	Output format = YCbCr 3 plane → 13th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA13[11:0] = 0x000 (ML = OX)	0

42.8.1.120 CIOCRSA14n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0304, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA14	[31:0]	RW	Output format = YCbCr 3 plane → 14th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA14[11:0] = 0x000 (ML = OX)	0

42.8.1.121 CIOCRSA15n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0308, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA15	[31:0]	RW	Output format = YCbCr 3 plane → 15th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA15[11:0] = 0x000. (ML = OX)	0

42.8.1.122 CIOCRSA16n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x030C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA16	[31:0]	RW	Output format = YCbCr 3 plane → 16th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA16[11:0] = 0x000. (ML = OX)	0

42.8.1.123 CIOCRSA17n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0310, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA17	[31:0]	RW	Output format = YCbCr 3 plane → 17th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA17[11:0] = 0x000. (ML = OX)	0

42.8.1.124 CIOCRSA18n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0314, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA18	[31:0]	RW	Output format = YCbCr 3 plane → 18th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA18[11:0] = 0x000. (ML = OX)	0

42.8.1.125 CIOCRSA19n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0318, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA19	[31:0]	RW	Output format = YCbCr 3 plane → 19th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA19[11:0] = 0x000. (ML = OX)	0

42.8.1.126 CIOCRSA20n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x031C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA20	[31:0]	RW	Output format = YCbCr 3 plane → 20th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA20[11:0] = 0x000. (ML = OX)	0

42.8.1.127 CIOCRSA21n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0320, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA21	[31:0]	RW	Output format = YCbCr 3 plane → 21st Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA21[11:0] = 0x000. (ML = OX)	0

42.8.1.128 CIOCRSA22n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0324, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA22	[31:0]	RW	Output format = YCbCr 3 plane → 22th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA22[11:0] = 0x000. (ML = OX)	0

42.8.1.129 CIOCRSA23n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0328, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA23	[31:0]	RW	Output format = YCbCr 3 plane → 23th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA23[11:0] = 0x000. (ML = OX)	0

42.8.1.130 CIOCRSA24n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x032C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA24	[31:0]	RW	Output format = YCbCr 3 plane → 24th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA24[11:0] = 0x000. (ML = OX)	0

42.8.1.131 CIOCRSA25n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0330, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA25	[31:0]	RW	Output format = YCbCr 3 plane → 25th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA25[11:0] = 0x000. (ML = OX)	0

42.8.1.132 CIOCRSA26n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0334, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA26	[31:0]	RW	Output format = YCbCr 3 plane → 26th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA26[11:0] = 0x000. (ML = OX)	0

42.8.1.133 CIOCRSA27n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0338, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA27	[31:0]	RW	Output format = YCbCr 3 plane → 27th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA27[11:0] = 0x000. (ML = OX)	0

42.8.1.134 CIOCRSA28n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x033C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA28	[31:0]	RW	Output format = YCbCr 3 plane → 28th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA28[11:0] = 0x000. (ML = OX)	0

42.8.1.135 CIOCRSA29n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0340, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA29	[31:0]	RW	Output format = YCbCr 3 plane → 29th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA29[11:0] = 0x000. (ML = OX)	0

42.8.1.136 CIOCRSA30n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0344, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA30	[31:0]	RW	Output format = YCbCr 3 plane → 30th Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA30[11:0] = 0x000. (ML = OX)	0

42.8.1.137 CIOCRSA31n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x0348, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA31	[31:0]	RW	Output format = YCbCr 3 plane → 31st Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA31[11:0] = 0x000. (ML = OX)	0

42.8.1.138 CIOCRSA32n (n = 0 to 3)

- Base Address: 0x1180_0000, 0x1181_0000, 0x1182_0000, 0x1183_0000
- Address = Base Address + 0x034C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CIOCRSA32	[31:0]	RW	Output format = YCbCr 3 plane → 32nd Cr frame start address NOTE: In tile mode, you should align this value to 4 KB, that is CIOCRSA32[11:0] = 0x000. (ML = OX)	0

43 FIMC_LITE (Camera Interface)

43.1 Overview of Camera Interface in FIMC_LITE

The camera interface (CAMIF) in FIMC_LITE is a fully interactive mobile camera interface. CAMIF supports parallel I/F like ITU R BT-601 standard and MIPI (CSI) Slave I/F. The maximum input image size is 8192 × 8192 pixels.

CAMIF in FIMC_LITE is designed to perform some functions.

- T_PatternMux generates test pattern to calibrate input sync signals as HREF and VSYNC.
- Capture specifies the capturing signal.
- Window cut.
- Use the register settings to invert video sync signals and pixel clock polarity.

43.1.1 Block Diagram

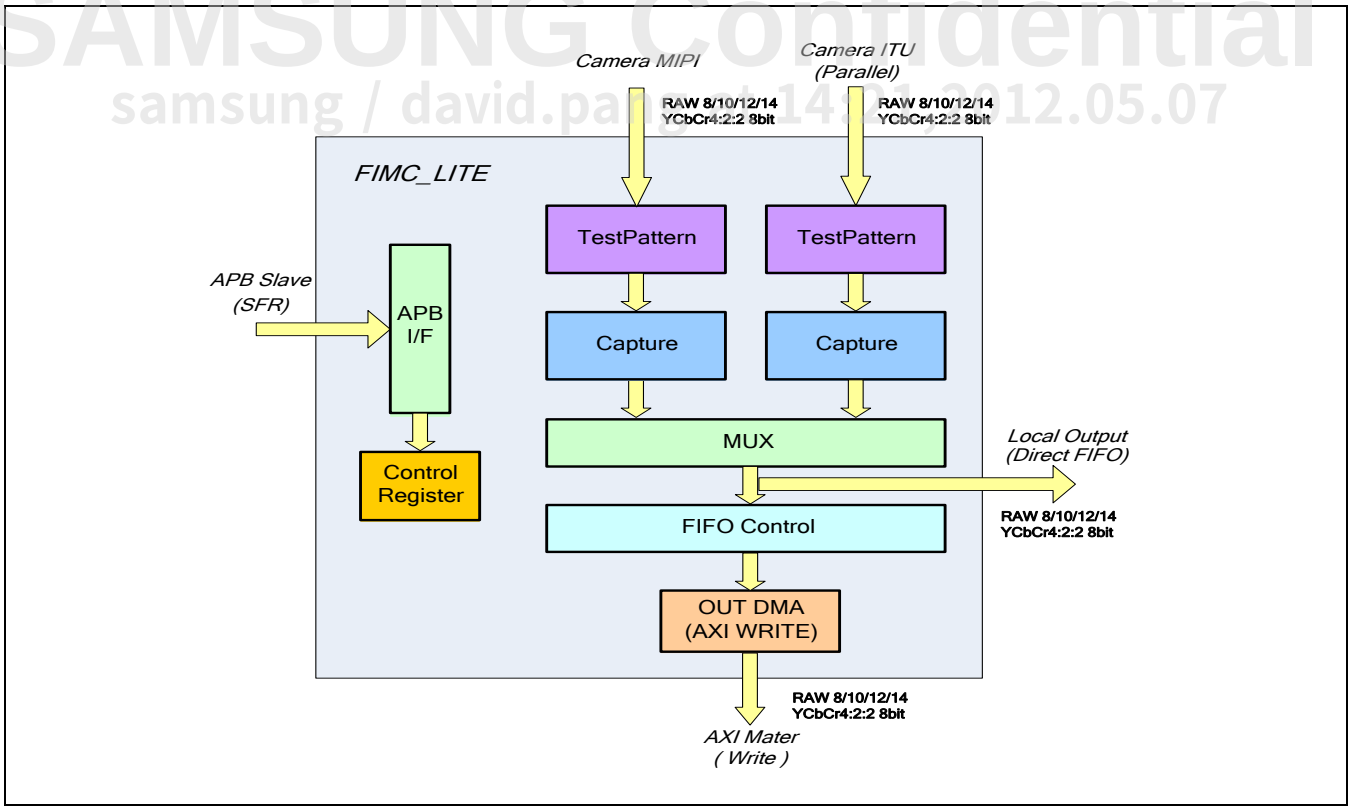


Figure 43-1 FIMC_LITE Block Diagram

43.2 Timing Diagram and Data Alignment of Camera

43.2.1 Parallel INTERFACE

CAMIF supports Parallel Interface like ITU601. Parallel Interface can support RAW 8/10/12/14-bits, YCbCr 4:2:2 8-bits, and JPEG 8-bits.

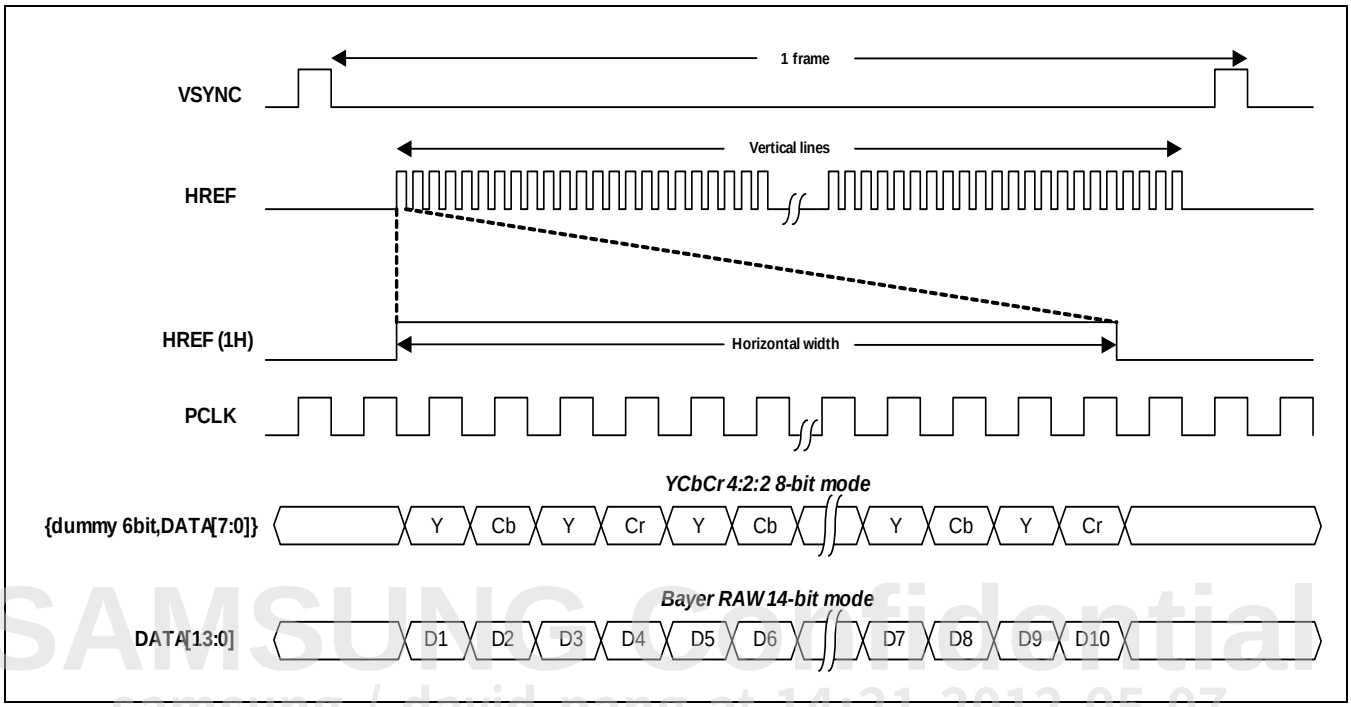


Figure 43-2 Parallel Interface (ITU) Timing Diagram (RAW and YCbCr 4:2:2)

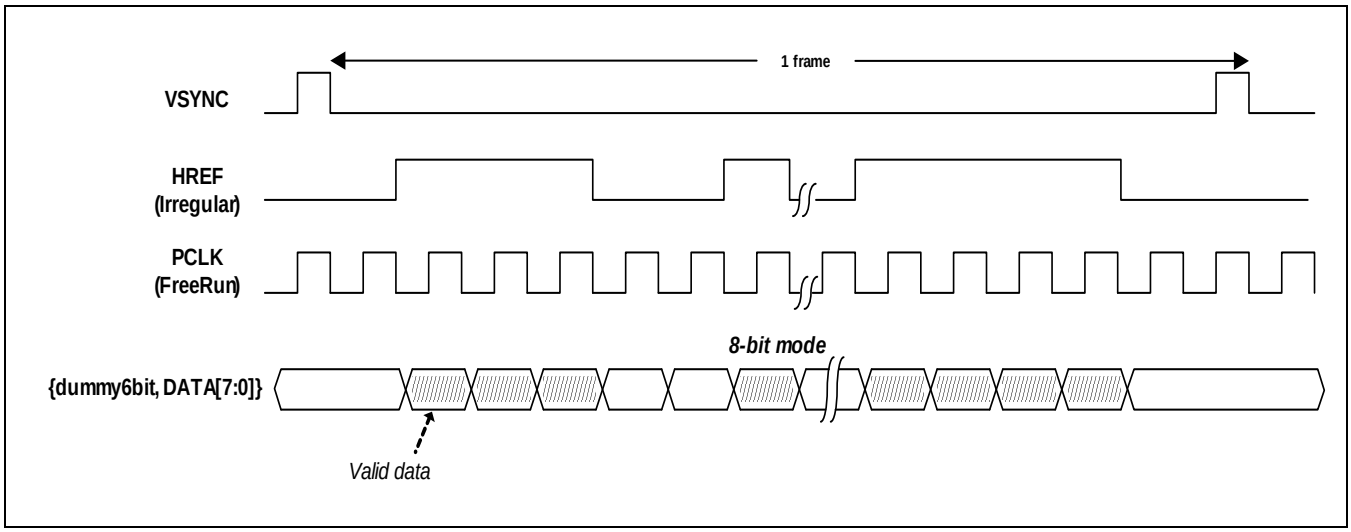


Figure 43-3 Parallel Interface (ITU) Timing Diagram (JPEG)

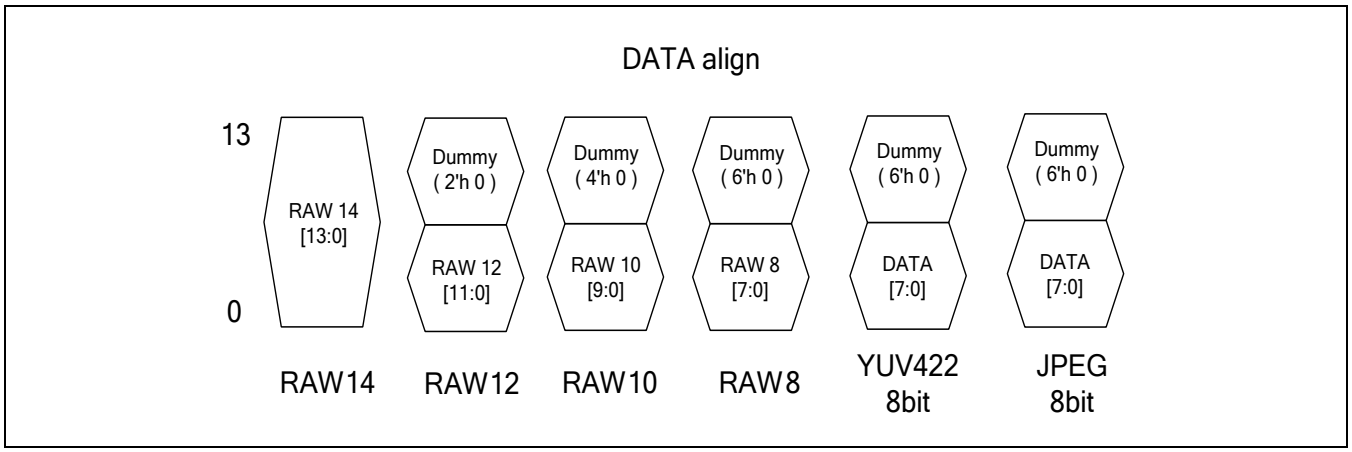


Figure 43-4 PARALLEL I/F Data Alignment

Caution: Do not combine all external camera interface IO signals with any other GPIO or bi-directional ports. It is recommended to use all external camera interface IOs as Shmitt-Trigger type IO for noise reduction.

43.2.2 MIPI CSI Slave Interface

CAMIF supports MIPI CSI Interface through MIPI CSI Slave. This interface can support RAW 8/10/12/14-bits, YCbCr 4:2:2 8-bits, and User-Defined 8-bits like JPEG 8-bits.

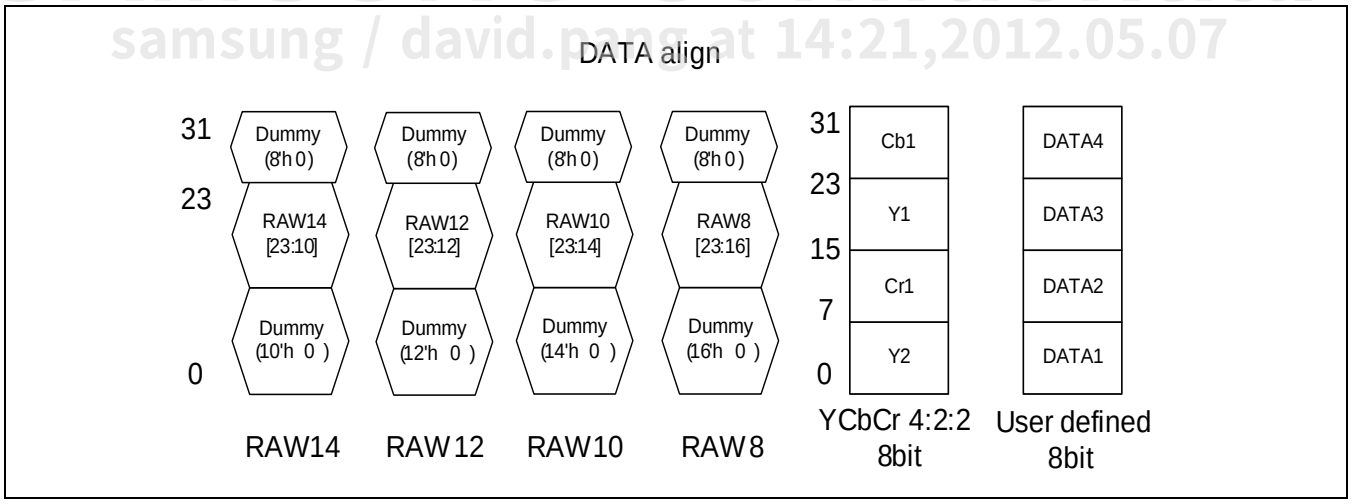


Figure 43-5 MIPI CSI Slave Data Alignment

43.2.3 Local Output Interface Data Alignment

CAMIF supports local output Interface through MIPI CSI Slave or Parallel I/F. This interface can support RAW 8/10/12/14-bits, YCbCr 4:2:2 8-bits, and User-Defined 8-bits like JPEG 8-bits.

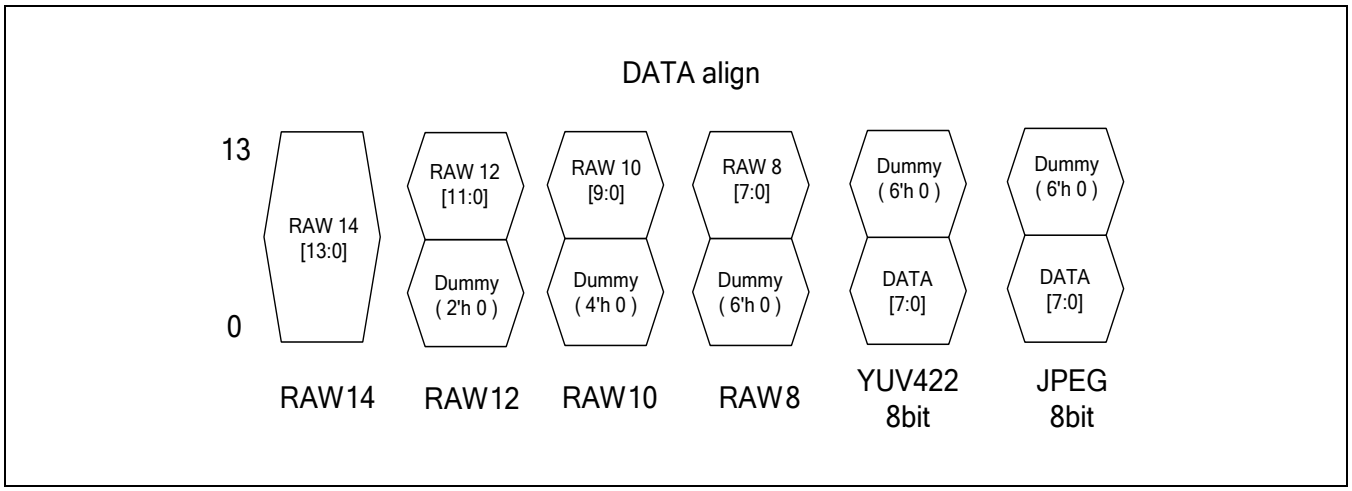


Figure 43-6 Local Out I/F Data Alignment

43.3 External Connection Guide

The CAMIF input signals must not result in inter-skewing of the pixel clock line. Therefore, it is recommended to use next pin location and routing.

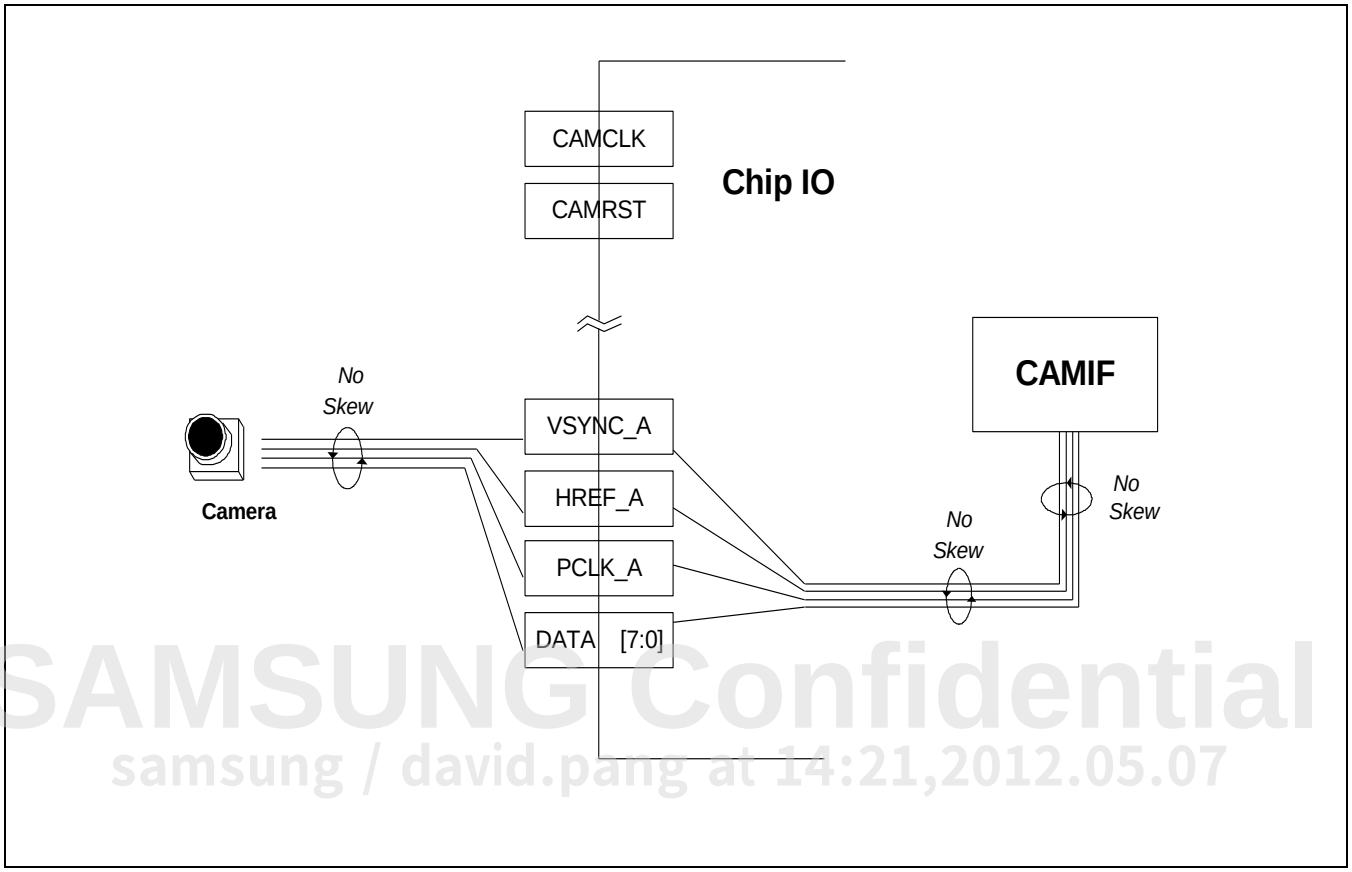


Figure 43-7 IO Connection Guide (Parallel Interface)

43.4 Input / Output Path

Each CAMIF consists of one input local path and two output paths, namely, Output local port and Output DMA port. The Input local port receives the image data from camera sensor. On the other hand, the Output local port sends the image data to another block or the Output DMA port stores the image data into memory.

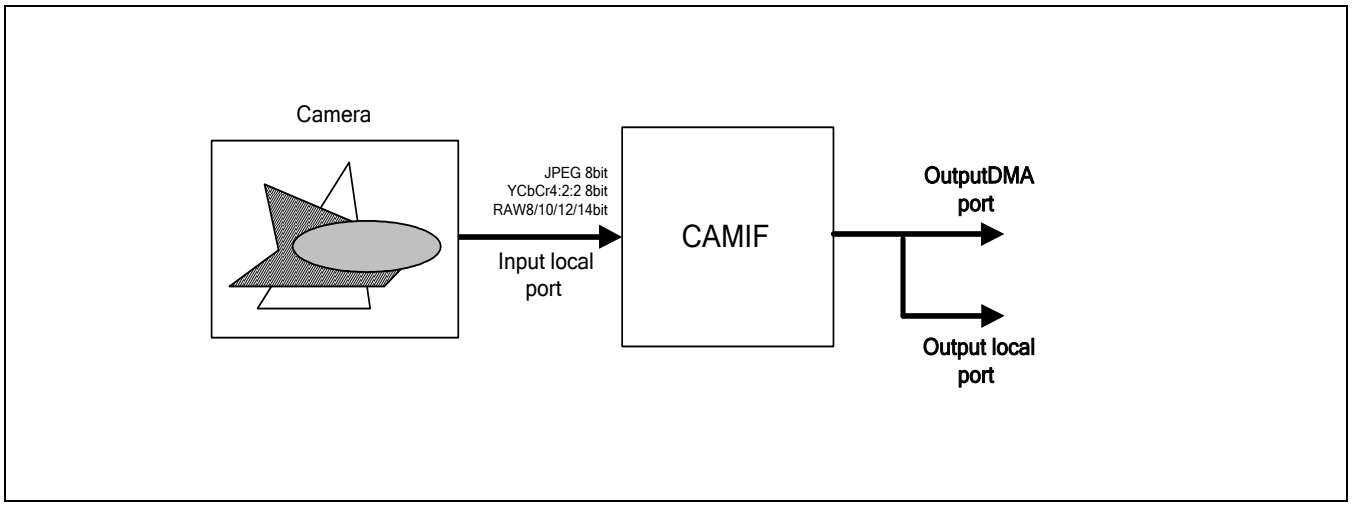


Figure 43-8 Input / Output Path

43.5 I/O Description

Signal	I/O	Description	PAD	Type
External Parallel ITU Camera Processor Interface Signal				
I_CAM_PCLK	I	Specifies the Pixel Clock driven by external Camera processor.	XciPCLK	Muxed
I_CAM_VSYNC	I	Specifies the Frame Sync driven by external Camera processor.	XciVSYNC	Muxed
I_CAM_HREF	I	Specifies the Horizontal Sync driven by external Camera processor.	XciHREF	Muxed
I_CAM _DATA[13:0]	I	Specifies the Pixel Data driven by external Camera processor.	XciDATA[13:0]	Muxed
CAM_MCLK	O	Specifies the Clock for external Camera processor.	XciCLKenb	Muxed

43.6 Register Description

43.6.1 Register Map

- Base Address: 0x1239_0000, 0x123A_0000

Register	Address	R/W	Description	Reset Value
CISRCSIZEn	0x0000	R/W	Source size register	0x00000000
CIGCTRLn	0x0004	R/W	Global control register	0x00000008
CIIMGCPTn	0x0008	R/W	Image Capture Enable register	0x00000000
CICPTSEQn	0x000C	R/W	Capture Sequence register	0xFFFFFFFF
CIWDOFSTn	0x0010	R/W	Window offset register	0x00000000
CIWDOFST2n	0x0014	R/W	Window offset register2	0x00000000
CIODMAFMTn	0x0018	R/W	Output DMA format register	0x00000000
CIOCANn	0x0020	RW	Camera output canvas register	0x00000000
CIOOFFn	0x0024	RW	Camera output DMA offset register.	0x00000000
CIOSAn	0x0030	R/W	Output DMA Start Address register	0x00000000
CISTATUSn	0x0040	R/W	Status register	0x00000000
CISTATUS2n	0x0044	R/W	Status register2	0x00000000
CITHOLDn	0x00F0	RW	Specifies QoS threshold register	0x00000000
CIGENERALn	0x00FC	R/W	General purpose register	0x00000001

43.6.1.1 CISRCSIZEn (n = 0 to 1)

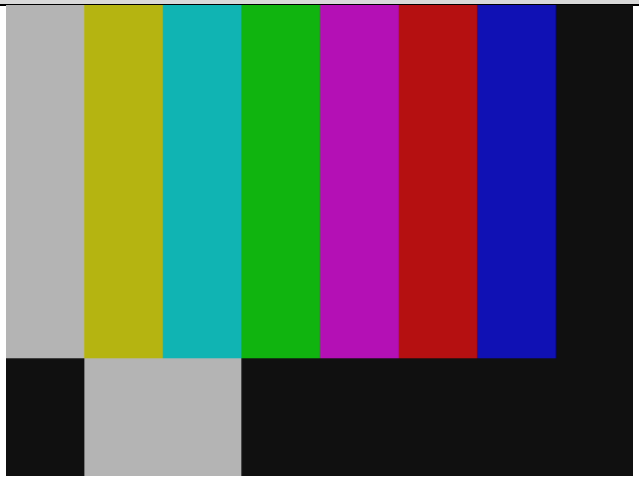
- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Description	Reset Value
RSVD	[31:30]	Reserved	0
SIZE_H	[29:16]	Specifies the source horizontal pixel number. In case of YCbCr 422 input, this value should be multiple of 2.	0
Order422_In	[15:14]	Specifies the camera input YCbCr order for 8-bit mode. 8-bit mode Data Flow → 00 = Y0Cb0Y1Cr0… 01 = Y0Cr0Y1Cb0… 10 = Cb0Y0Cr0Y1… 11 = Cr0Y0Cb0Y1…	0
SIZE_V	[13:0]	Specifies the source vertical line number.	0

43.6.1.2 CIGCTRLn (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0008

Name	Bit	Description	Reset Value
RSVD	[31:30]	Reserved	0
Format	[29:24]	Specifies the image format. If the JPEG format is image format, image format is selected User defined. 0x1E = YUV422 8-bit (1plane only) 0x2A = RAW8 0x2B = RAW10 0x2C = RAW12 0x2D = RAW14 0x30 to 3F = User defined (JPEG)	0
RSVD	[23:22]	Reserved	0
ShadowMask	[21]	Shadow register cannot update by Hardware frame start (input path should be local path). At the first start needs ShadowMask = "0" 0 = Enable (update is possible) 1 = Disable (update is impossible)	0
ODMAEnable	[20]	OutDMA enable register. If this register is set "Disable", Output DMA path cannot run except any setting. 0 = Enable 1 = Disable	0
SwRst_Req	[19]	Software reset request. When software reset is generated, SwRst_Req is automatically cleared.	0
SwRst_Rdy	[18]	IP safe mode and Software reset request acknowledgement status. When software reset is generated, SwRst_Rdy is automatically cleared. This signal is a read only signal.	0
SwRst	[17]	Software reset register. When software reset is generated, SwRst is automatically cleared. i.e.) [Normal software reset procedure] 1. Set SwRst_Req "1" → Check SwRst_Rdy "1" → Set SwRst "1" [Force software reset procedure] 1. Set SwRst "1" (without Software request)	0
RSVD	[16]	Reserved	0
TestPattern	[15]	This register should be set at YCbCr422 8-bit mode only. Source size should be 640 x 480 size and Order422_In register should be "00". 0 = External camera processor input (normal) 1 = Color bar test pattern	0

Name	Bit	Description	Reset Value
			
InvPolPCLK	[14]	0 = Normal 1 = Inverse the polarity of PCLK	0
InvPolVSYNC	[13]	0 = Normal 1 = Inverse the polarity of VSYNC	0
InvPolHREF	[12]	0 = Normal 1 = Inverse the polarity of HREF	0
RSVD	[11:9]	Reserved	0
IRQ_LastEn0	[8]	0 = Last capture end interrupt enable (Interrupt is generated after all frame capture finish) 1 = Last capture end interrupt disable	0
IRQ_EndEn0	[7]	0 = Frame end interrupt enable (Interrupt is generated after each frame capture finish) 1 = Frame end interrupt disable	0
IRQ_StartEn0	[6]	0 = Interrupt enables at Frame start point 1 = Interrupt disables at Frame start point	0
IRQ_OvfEn0	[5]	0 = Overflow interrupt enable 1 = Overflow interrupt disable (Interrupt is generated during overflow occurrence)	0
RSVD	[4]	Reserved	0
SelCam	[3]	Selects the External camera. 0 = Selects Parallel (ITU) Camera 1 = Selects MIPI Camera	1
RSVD	[2:0]	Reserved	0

43.6.1.3 CIIMGCPTn (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

CIIMGCPT	Bit	Description	Reset Value
ImgCptEn	[31]	Enables camera interface global capture. (Shadow) NOTE: ImgCptEn must be set at last. others SFR setting → ImgCptEn setting	0
RSVD	[30:26]	Reserved	0
Cpt_FrEn	[25]	Controls capture frame. 0 = Disable (FreeRun mode) 1 = Enable (Step-by-Step frame one shot mode) NOTE: User should not change this bit under capture enable status.	0
RSVD	[24]	Reserved	0
Cpt_FrPtr	[23:19]	Captures sequence turnaround pointer.	0
Cpt_FrMod	[18]	Captures frame control mode. 0 = Applies Cpt_FrEn mode (Captures frames along Cpt_FrSeq when Cpt_FrEn is high. This sequence repeats until capture frame control is disabled.) 1 = Applies Cpt_FrCnt mode (Captures Cpt_FrCnt frames along Cpt_FrSeq, after enabling capture DMA frame control. If Cpt_FrCnt = 0, then capture ends.)	0
Cpt_FrCnt	[17:10]	Specifies number of frames to be captured. If register reads, then you can see the value of a shadow register, which is downcounted if a frame is captured. In other words, Cpt_FrCnt has an initially loaded value after a frame is captured. NOTE: User have to disable and enable Cpt_FrEn register before starting CAMIF to use this (capture frame count) funciton	0
RSVD	[9:0]	Reserved	0

43.6.1.4 CICPTSEQn (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x000C, Reset Value = 0xFFFF_FFFF

CICPTSEQ	Bit	Description	Reset Value
Cpt_FrSeq	[31:0]	Specifies capture sequence pattern. This register is valid if Cpt_FrEn has a high value.	FFFF_FFFF

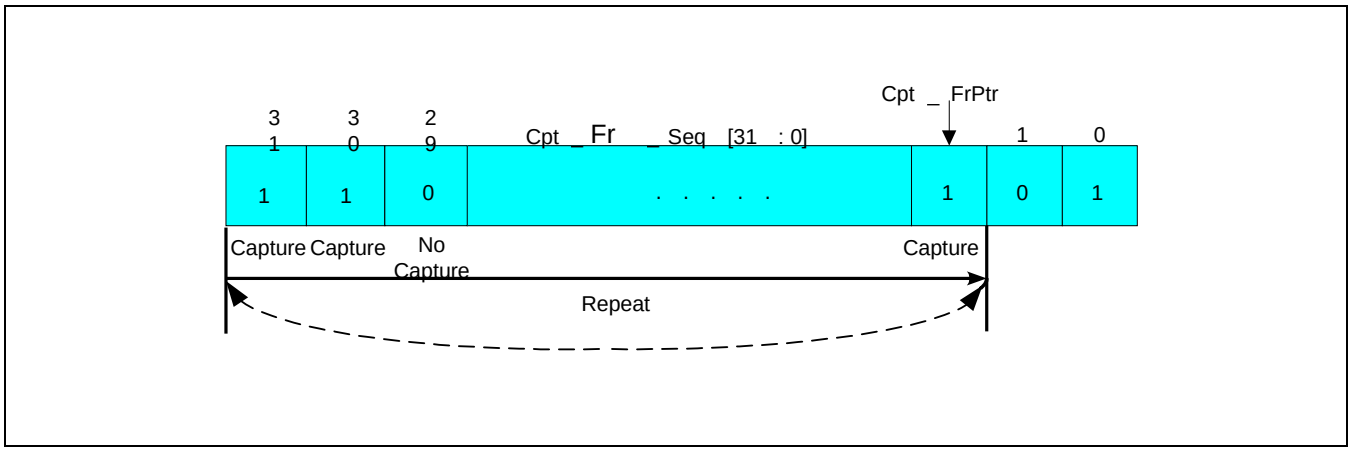


Figure 43-9 Capture Frame Control

※ For skipped frames, Interrupt is not generated.

43.6.1.5 CIWDOFSTn (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

CIWDOFST	Bit	Description	Reset Value
WinOfsEn	[31]	0 = No offset 1 = Enables window offset Note) If input format is JPEG or User-defined, this function should be set "0"	0
ClrOvFiY	[30]	0 = Normal 1 = Clears the overflow indication flag of input FIFO Y	0
RSVD	[29]	Reserved	0
WinHorOfst	[28:16]	Specifies window horizontal offset by pixel unit. In case of YCbCr 422 input, this value should be multiple of 2.	0
ClrOvFiCb	[15]	0 = Normal 1 = Clears the overflow indication flag of input FIFO Cb	0
ClrOvFiCr	[14]	0 = Normal 1 = Clears the overflow indication flag of input FIFO Cr	0
RSVD	[13]	Reserved	0
WinVerOfst	[12:0]	Specifies window vertical offset by pixel unit.	0

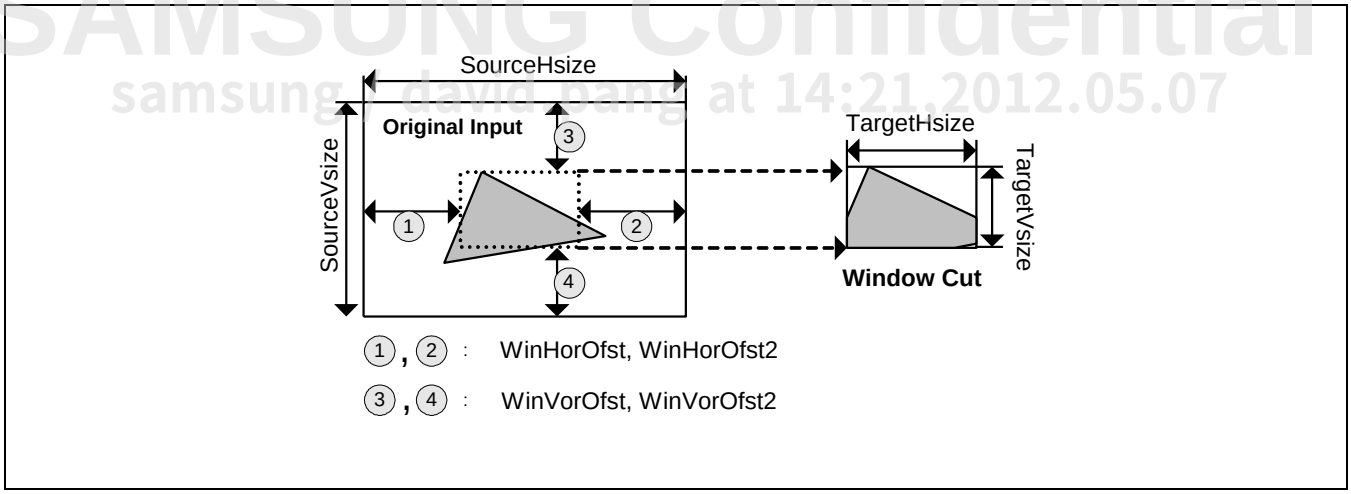


Figure 43-10 Camera Window Offset Scheme

43.6.1.6 CIWDOFST2n (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

CIWDOFST2	Bit	Description	Reset Value
RSVD	[31:29]	Reserved	0
WinHorOfst2	[28:16]	Specifies the window horizontal offset2 by pixel unit. In case of YCbCr 422 input, this value should be multiple of 2.	0
RSVD	[15:13]	Reserved	0
WinVerOfst2	[12:0]	Specifies the window vertical offset2 by pixel unit.	0

43.6.1.7 CIODMAFMTn (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

CIODMAFMT	Bit	Description	Reset Value
RSVD	[31]	Reserved	0
MODE_W	[30:29]	OUTPUT DMA address access style (RAW and JPEG format are available only Linear mode) 0 = Linear 1 = Reserved 2 = Reserved 3 = Reserved NOTE: If input format is either JPEG or RAW, User can not use tile mode.	0
RSVD	[28:16]	Reserved	0
RAW_Con	[15]	0 = Normal (2D DMA) 1 = Memory storing style of the RAW formats are 1D DMA	0
PACK12	[14]	0 = Normal (If RAW8, RAW14, JPEG or YCbCr is image format, PACK12 should be "0") 1 = Packed RAW10/12 format	0
RSVD	[13:6]	Reserved	0
ORDER1P_OUT	[5:4]	YCbCr 4:2:2 1plane output memory storing style order. bit: MSB LSB 00: Cr1Y3Cb1Y2Cr0Y1Cb0Y0 01: Cb1Y3Cr1Y2Cb0Y1Cr0Y0 10: Y3Cr1Y2Cb1Y1Cr0Y0Cb0 11: Y3Cb1Y2Cr1Y1Cb0Y0Cr0	0
RSVD	[3:0]	Reserved	0

43.6.1.8 CIOCANn (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

CIOCAN	Bit	Description	Reset Value
RSVD	[31:30]	Reserved	0
OCAN_V	[29:16]	Output DMA Canvas vertical register. This value is based on line unit. (OCAN_V >= Image height + offset height)	0
RSVD	[15:14]	Reserved	0
OCAN_H	[13:0]	Output DMA Canvas horizontal register. This value is based on pixel unit. (OCAN_H >= Image width + offset width)	0

43.6.1.9 CIOOFFn (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

CIOOFF	Bit	Description	Reset Value
RSVD	[31:30]	Reserved	0
OOFF_V	[29:16]	Output DMA vertical offset. Offset value is based on line unit.	0
RSVD	[15:14]	Reserved	0
OOFF_H	[13:0]	Output DMA horizontal offset. Offset value is based on pixel unit. (64bits align). i.e.) YCbCr422 1plane: 4's multiple RAW8,JPEG: 8's multiple RAW10: 5's multiple RAW12: 5's multiple RAW14: 4's multiple	0

43.6.1.10 CIOSAn (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

CIOSA	Bit	Description	Reset Value
OSA	[31:0]	Output DMA start address. (64 bits align, OSA[2:0] = 0x0)	0

43.6.1.11 CISTATUSn (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

CISTATUS	Bit	Description	Reset Value
RSVD	[31:23]	Reserved	0
MIPI_VVALID	[22]	MIPI CSI Vertical valid signal (R)	0
MIPI_HVALID	[21]	MIPI CSI Horizontal valid signal (R)	0
MIPI_DVALID	[20]	MIPI CSI Data valid signal (R)	0
RSVD	[19:15]	Reserved	0
ITU_VSYNC	[14]	Parallel ITU camera Vertical sync signal (R)	0
ITU_HREF	[13]	Parallel ITU camera Horizontal reference signal (R)	0
RSVD	[12:11]	Reserved	0
OvFiY	[10]	Overflow state of FIFO Y (R)	0
OvFiCb	[9]	Overflow state of FIFO Cb (R)	0
OvFiCr	[8]	Overflow state of FIFO Cr (R)	0
IRQ_SRC	[7:4]	Interrupt source status. When IRQ_CAM register is cleared, this register is automatically cleared. (R) IRQ_SRC[7] high: Interrupt by overflow source IRQ_SRC[6] high: Interrupt by last capture end source IRQ_SRC[5] high: Interrupt by frame start source IRQ_SRC[4] high: Interrupt by frame end source	0
RSVD	[3:1]	Reserved	0
IRQ_CAM	[0]	When interrupt is generated, this IRQ_CAM is high. IRQ_CAM is cleared by user setting "0" (R/W)	0

43.6.1.12 CISTATUS2n (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

CISTATUS2	Bit	Description	Reset Value
RSVD	[31:2]	Reserved	0
LastCaptureEnd	[1]	When ImgCptEn value is (1 → 0) and frame operation is done, Last frame capture status is high. LastCaptureEnd is cleared by user setting "0" (R/W)	0
FrameEnd	[0]	When each frame capture operation is done, FrameEnd status is high. FrameEnd is cleared by user setting "0" (R/W)	0

43.6.1.13 CITHOLDn (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x00F0, Reset Value = 0x0000_0000

CISTATUS2	Bit	Description	Reset Value
RSVD	[31]	Reserved	0
W_QoS_EN	[30]	0 = QoS disable for write DMA channel buffer 1 = QoS enable for write DMA channel buffer	0
RSVD	[29:7]	Reserved	0
WTh_QoS	[6:0]	Write buffer threshold register (related write output DMA) If WTh_Qos < buffer write count, write channel is generated no margin signal for bus high performance.	0

43.6.1.14 CIGENERALn (n = 0 to 1)

- Base Address: 0x1239_0000, 0x123A_0000
- Address = Base Address + 0x00FC, Reset Value = 0x0000_0001

CIGENERAL	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
General0	[0]	Selects the camera channel for general purpose. This register is valid when this register map is CAMIF A channel 0 = Selects Camera A channel 1 = Selects Camera B channel	1

44 MIPI-DSI Master

44.1 Overview

44.2 Features

The features of MIPI DSIM are:

- Complies with MIPI DSI Standard Specification V1.01r11:
  - Maximum resolution ranges up to WXGA (1280 × 800)
  - Supports 1, 2, 3, or 4 data lanes
  - Supports these pixel formats:
    - 16 bpp
    - 18 bpp packed
    - 18 bpp loosely packed (3 byte format)
    - 24 bpp
- Interfaces:
  - Complies with Protocol-to-PHY Interface (PPI) in MIPI D-PHY Specification version 0.90
  - Supports RGB Interface for Video Image Input from display controller
  - Supports I80 Interface for Command Mode Image input from display controller
  - Supports PMS control interface for PLL to configure byte clock frequency
  - Supports Prescaler to generate escape clock from byte clock

44.2.1 Block Diagram

Figure 44-1 illustrates MIPI DSI system block diagram.

44.2.1.1 Total System Block Diagram

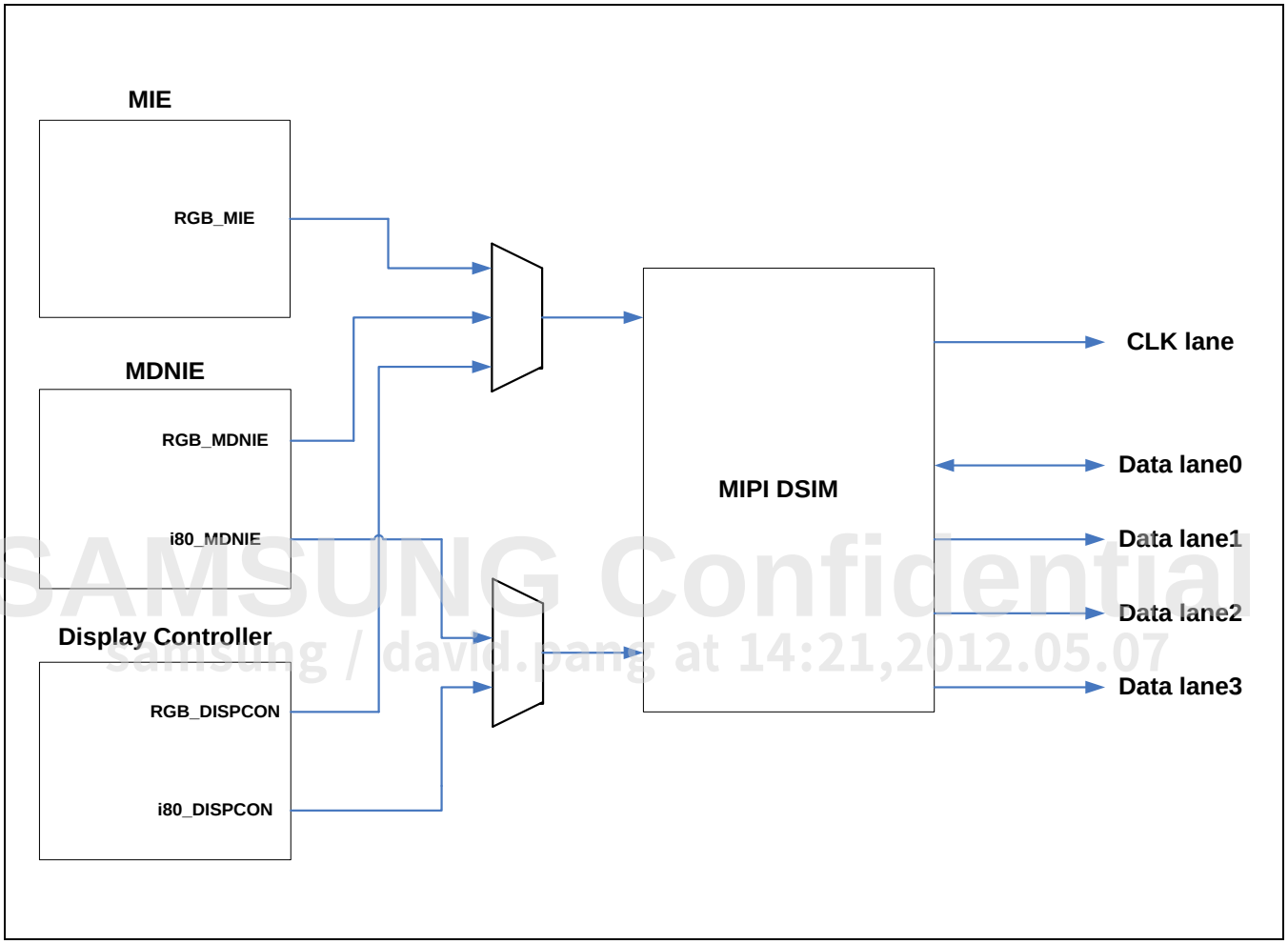


Figure 44-1 MIPI DSI System Block Diagram

NOTE:

1. DSIM receives data from the three different IPs, namely:
   - Mobile Image Enhancement (MIE),
   - MDNIE
   - Display Controller.
2. You can select one of these data paths by setting DISPLAY_CONTROL[1:0] (0x1001_0210).
   Refer to Chapter 12, System Register Controller, for more information.

44.2.1.2 Internal Primary FIFOs

Table 44-1 describes Internal Primary First In First Out (FIFO) list.

Table 44-1 Internal Primary FIFO List

Port	FIFO Type	Size	Description
Main display	Packet Header FIFO	3 byte × 64 depth	Specifies the packet header FIFO for main display.
	Payload FIFO	4 byte × 1024 depth	Specifies the payload FIFO for main display image.
Sub display for I80 interface image data	Packet Header FIFO	3 byte × 4 depth	Specifies the packet header FIFO for I80 interface sub display.
	Payload FIFO	4 byte × 512 depth	Specifies the payload FIFO for I80 interface sub display image.
Command for I80 interface command	Packet Header FIFO	3 byte × 16 depth	Specifies the packet header FIFO for I80 interface command packet.
	Payload FIFO	4 byte × 16 depth	Specifies the payload FIFO for I80 interface command long packet payload.
SFR for general packets	Packet Header FIFO	3 byte × 16 depth	Specifies the packet header FIFO for general packet.
	Payload FIFO	4 byte × 512 depth	Specifies the payload FIFO for general long packet.
Rx FIFO	Packet header and Payload FIFO	4 byte × 64 depth	Specifies Rx FIFO for LPDR. This FIFO is common for packet header and payload.

44.2.1.3 Packet Header Arbitration

There are four FIFO packet headers for Tx, namely:

- Main display
- Sub-display
- I80 INTERFACE command
- SFR FIFO

The main and sub display FIFO packet headers contain the image data, the I80 INTERFACE command FIFO packet header contains the command packets, and the SFR FIFO packet header contains command packets, sub-display image data (in video mode), and so on.

The packet header arbiter has a "Fixed priority" algorithm. Priority order is:

1. Main display
2. Sub display
3. I80 INTERFACE command
4. SFR FIFO packet header

In the video mode, DSIM does not use sub display and I80 INTERFACE command FIFO. The SFR FIFO packet header verifies if the main display FIFO is empty (no request) in not-active image region and then sends its request.

44.2.1.4 RxFIFO Structure

To Read the packets that are received through low-power data receiving mode, RxFIFO acts like an SFR. RxFIFO is an asynchronous FIFO with ByteClk and PCLK domains as input clock and output clock domains, respectively. The Rx data is synchronized to RxClk. RxBUF has four Rx Byte buffers for aligning byte to word.

The packet headers of all the packets that are stored in RxFIFO are word-aligned. This means that the first byte of a packet is always stored in Least Significant Byte (LSB). For example, when a long packet has 7-byte payload, DSIM fills the last byte with dummy byte and the DSIM fills the last byte stores the next packet in the next word.

Figure 44-2 illustrates the Rx data word alignment.

NOTE: CRC data is not stored in RxFIFO.

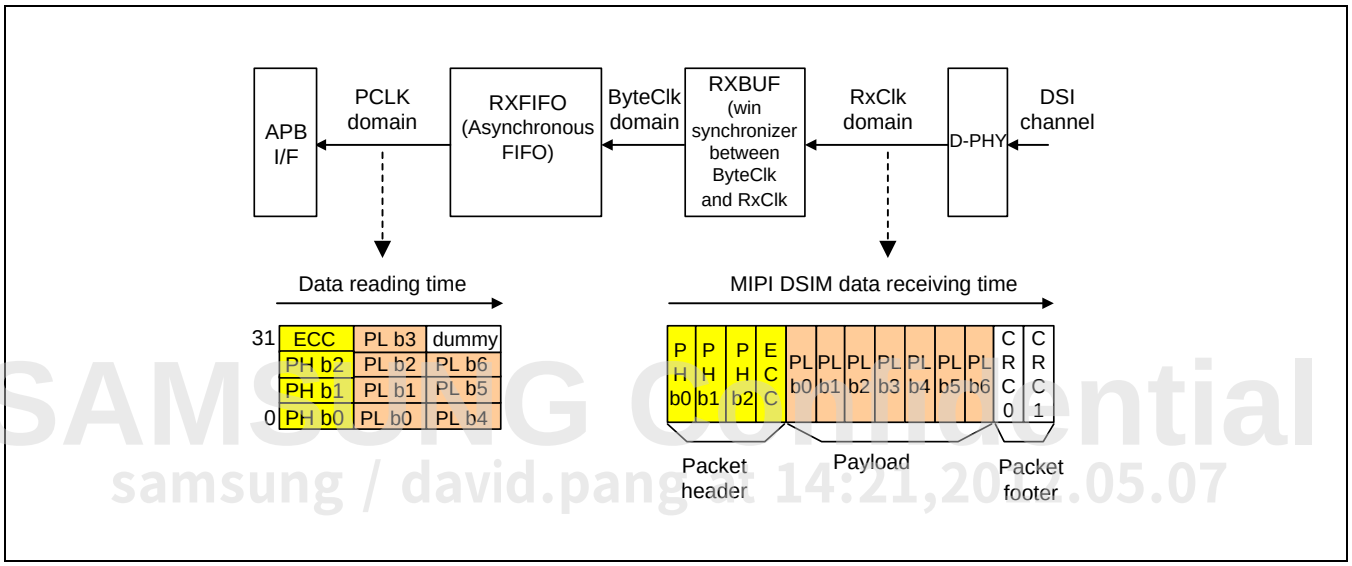


Figure 44-2 Rx Data Word Alignment

44.2.2 Interfaces and Protocol

Figure 44-3 illustrates Signal Converting Diagram in video mode between Display-controller and DSIM.

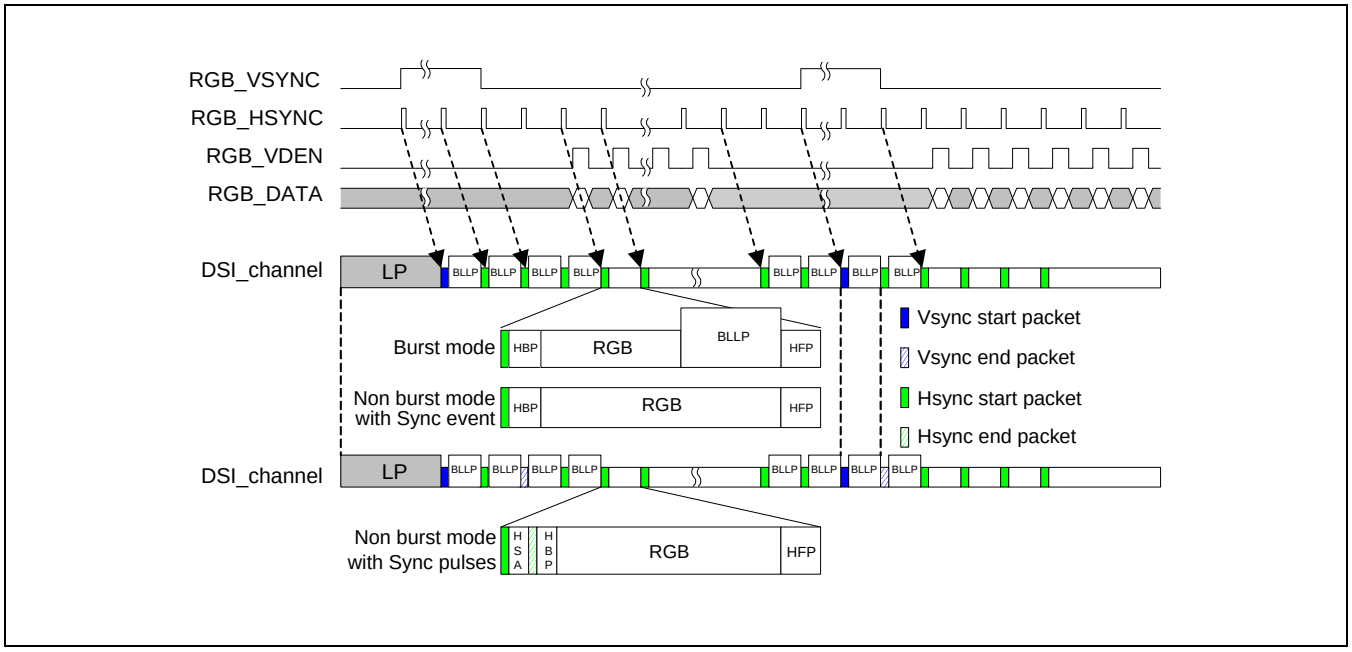


Figure 44-3 Signal Converting Diagram in Video Mode

44.2.2.1 Interface Timing and Protocol

This section includes:

- Display Controller Interface
- RGB Interface
- HSA Mode
- HSE Mode
- Transfer General Data in Video Mode
- MIPI DSIM Converts RGB Interface to Video Mode
- I80 Interface
- Relation between Input Transactions and DSI Transactions

44.2.2.1.1 Display Controller Interface

MIPI-DSIM has two-display controller interfaces, namely:

- RGB INTERFACE for main display
- CPU INTERFACE (I80 INTERFACE) for main/sub display

The video mode uses RGB INTERFACE, while the command mode uses CPU INTERFACE.

DSIM loads the RGB image data on the data bus of RGB INTERFACE and I80 INTERFACE with the same order: RGB_VD[23:0] or SYS_VDOUT[23:0] is {R[7:0], G[7:0], B[7:0]}. Each byte aligns to the most significant bit.

For example, in the 12-bit mode, only three 4-bit values are valid as R, G, and B each, that is, data[23:20], data[15:12], and data[7:4]. The DSIM ignores rest of the bits.

44.2.2.1.2 RGB Interface

Vsync, Hsync, and VDEN are active high signals. Among the three signals, Vsync and Hsync are the pulse types that require several video clocks. RGB_VD [23:0] is {R [7:0], G [7:0], and B [7:0]}. DSIM synchronizes all sync signals to the rising edge of RGB_VCLK. The display controller sends a minimum of one horizontal line length of Vsync pulse, V back porch, and V front porch. The width of the Hsync pulse should be longer than 1-byte clock cycle.

44.2.2.1.3 HSA Mode

HSA mode specifies the Horizontal Sync Pulse area disable mode.

Figure 44-4 illustrates the block timing diagram of HSA mode (HSA Mode Reset: DSIM_CONFIG[20] = 0)

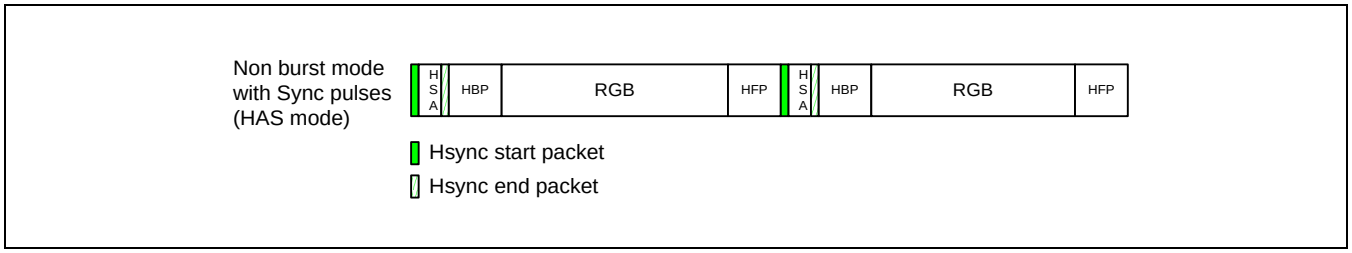


Figure 44-4 Block Timing Diagram of HSA Mode (HSA Mode Reset: DSIM_CONFIG[20] = 0)

Figure 44-5 illustrates the block timing diagram of HSA Mode (HSA Mode Set: DSIM_CONFIG[20] = 1).

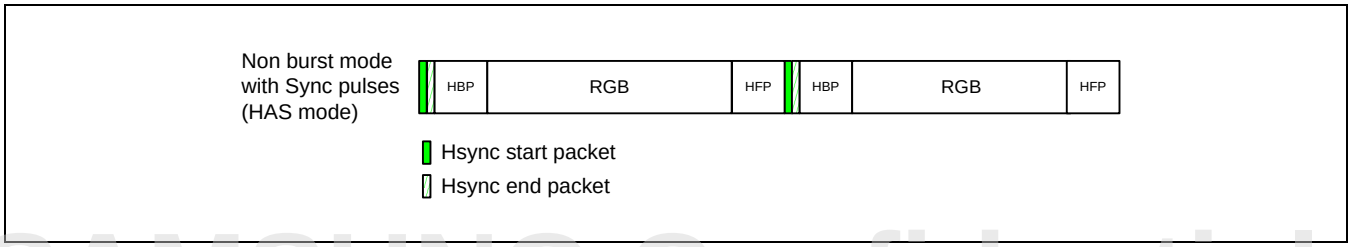


Figure 44-5 Block Timing Diagram of HSA Mode (HSA Mode Set: DSIM_CONFIG[20] = 1)

HBP mode specifies the Horizontal Back Porch disable mode.

Figure 44-6 illustrates the block timing diagram of HBP Mode (HBP Mode Reset: DSIM_CONFIG[21] = 0).

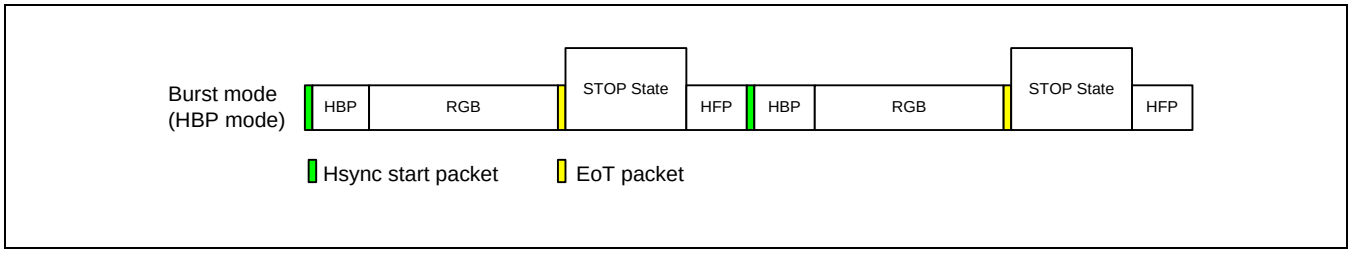


Figure 44-6 Block Timing Diagram of HBP Mode (HBP Mode Reset: DSIM_CONFIG[21] = 0)

Figure 44-7 illustrates the timing diagram of HBP mode (HBP Mode Set: DSIM_CONFIG[21] = 1).

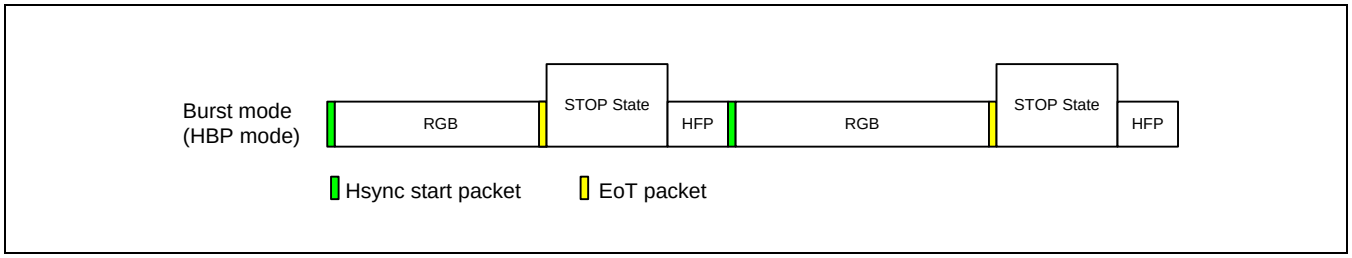


Figure 44-7 Block Timing Diagram of HBP Mode (HBP Mode Set: DSIM_CONFIG[21] = 1)

HFP mode specifies the Horizontal Front Porch disable mode.

Figure 44-8 illustrates the block timing diagram of HFP mode (HFP Mode Reset: DSIM_CONFIG [22] = 0).

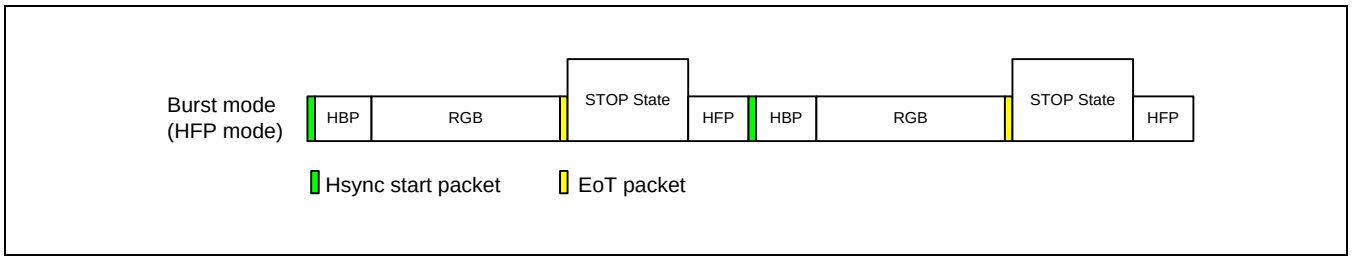


Figure 44-8 Block Timing Diagram of HFP Mode (HFP Mode Reset: DSIM_CONFIG[22] = 0)

Figure 44-9 illustrates the block timing diagram of HFP mode (HFP Mode Set: DSIM_CONFIG [22] = 1).

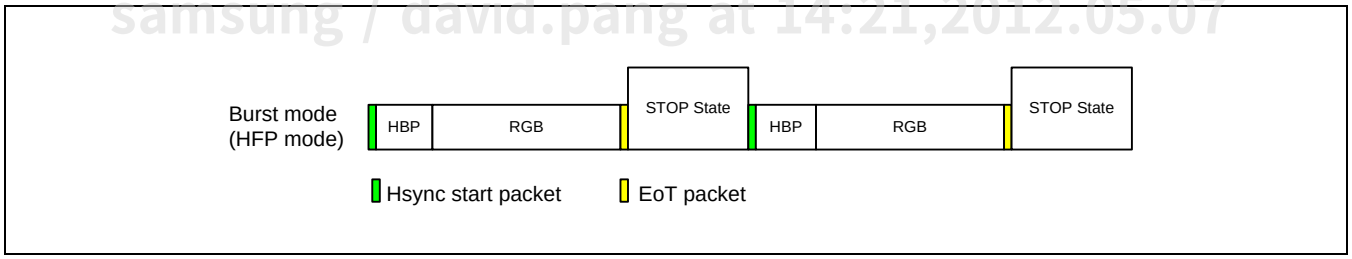


Figure 44-9 Block Timing Diagram of HFP Mode (HFP Mode Set: DSIM_CONFIG[22] = 1)

44.2.2.1.4 HSE Mode

HSE mode specifies the Horizontal Sync End Packet Enable mode in Vsync pulse or Vporch area.

Figure 44-10 illustrates the block timing diagram of HSE mode (HSE Mode Reset: DSIM_CONFIG [23] = 0).

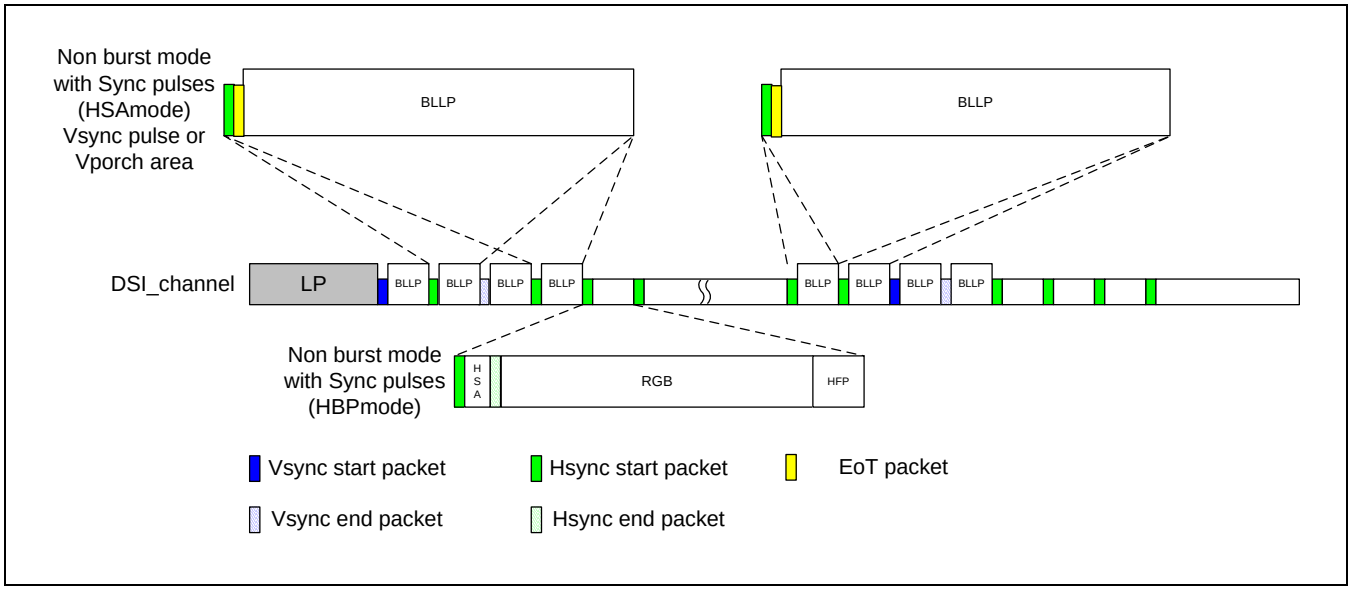


Figure 44-10 Block Timing Diagram of HSE Mode (HSE Mode Reset: DSIM_CONFIG[23] = 0)

Figure 44-11 illustrates the block timing diagram of HSE mode (HSE Mode Set: DSIM_CONFIG [23] = 1).

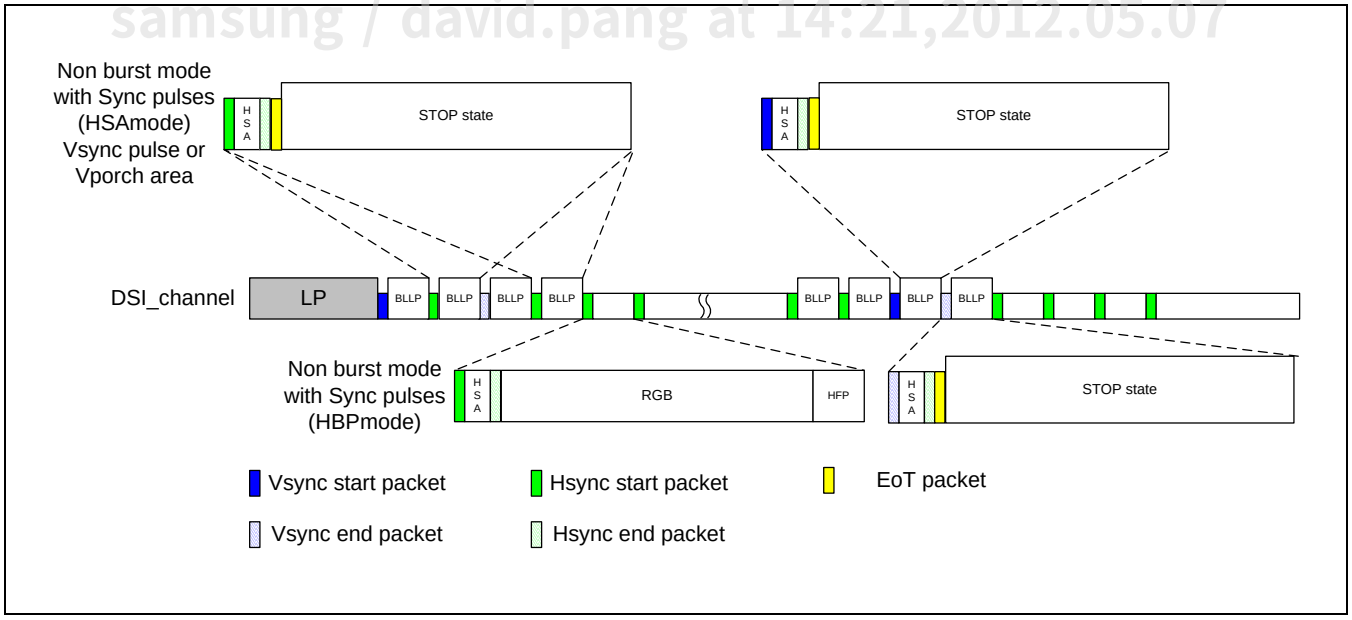


Figure 44-11 Block Timing Diagram of HSE Mode (HSE Mode Set: DSIM_CONFIG[23] = 1)

44.2.2.1.5 Transfer General Data in Video Mode

Figure 44-12 illustrates the stable VFP area before command transfer allowing area.

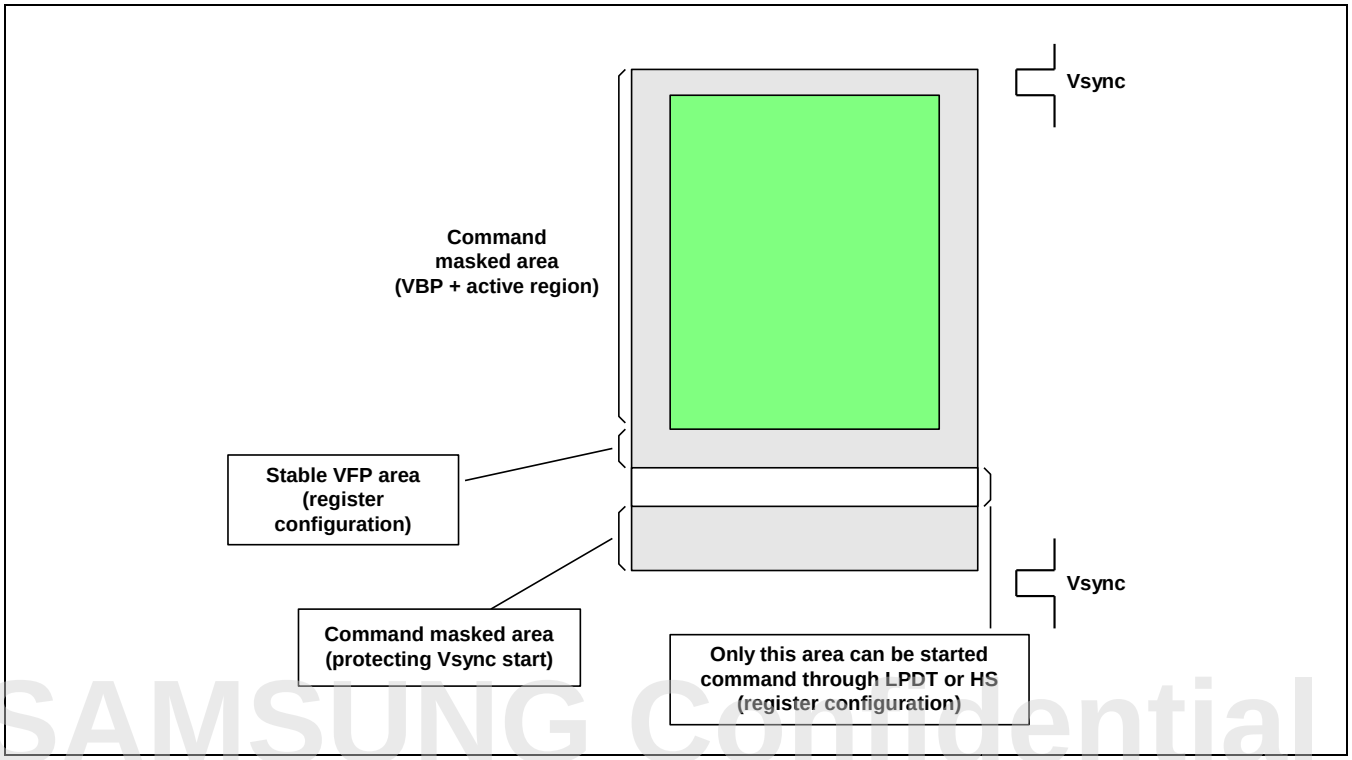


Figure 44-12 Stable VFP Area Before Command Transfer Allowing Area

44.2.2.1.6 MIPI DSIM Converts RGB Interface to Video Mode

Vsync and Hsync packets are extremely important to protect image in video mode. MIPI DSIM allows several lines in Vertical Front Porch (VFP) area to transfer general data transfer. In *Figure 44-12*, DSIM divides the vertical front porch into three areas, namely:

- Stable VFP area
- Command allowed area
- Command masked area

The register configures stable VFP area. Configuration boundary is 11'h000 to 11'h7FFF in DSIM_MVPORCH.

The register also configures the command allowed area. Configuration boundary is 4'h0 to 4'hF in DSIM_MVPORCH. DSIM allows only this area to start "command transfer" through HS mode or Low Power Data Transmit (LPDT). In LPDT, the data transferring requires a long time to complete (approximately hundreds of microseconds or more).
During this time, the Hsync packet does not arrive due to LPDT long packet. MIPI DSIM consists of a big size FIFO for the lost Hsync packet. After LPDT, MIPI DSIM transfers these Hsync packets immediately through HS mode.

To protect Vsync, mask the command mask area. This area is calculated by using LPDT bandwidth. For example, if EscClk is 10 MHz, the maximum long-packet payload size is 1 KB if LPDT mode, LPDT transferring time is 824 μs (packet size: 1030 byte, LPDT maximum bandwidth: 10 Mbps). When one line time is 20 μs, the line timing violation occurs in 42 lines. Therefore, the command masked area is larger than 42 + α. "α" Represents the transferring time of the violated Hsync packets.

You should configure the display controller in such a way that VFP lines are the total of stable vfp, command allowed area, and command masked area.

44.2.2.1.7 I80 Interface

Figure 44-13 illustrates the I80 Interface timing diagram.

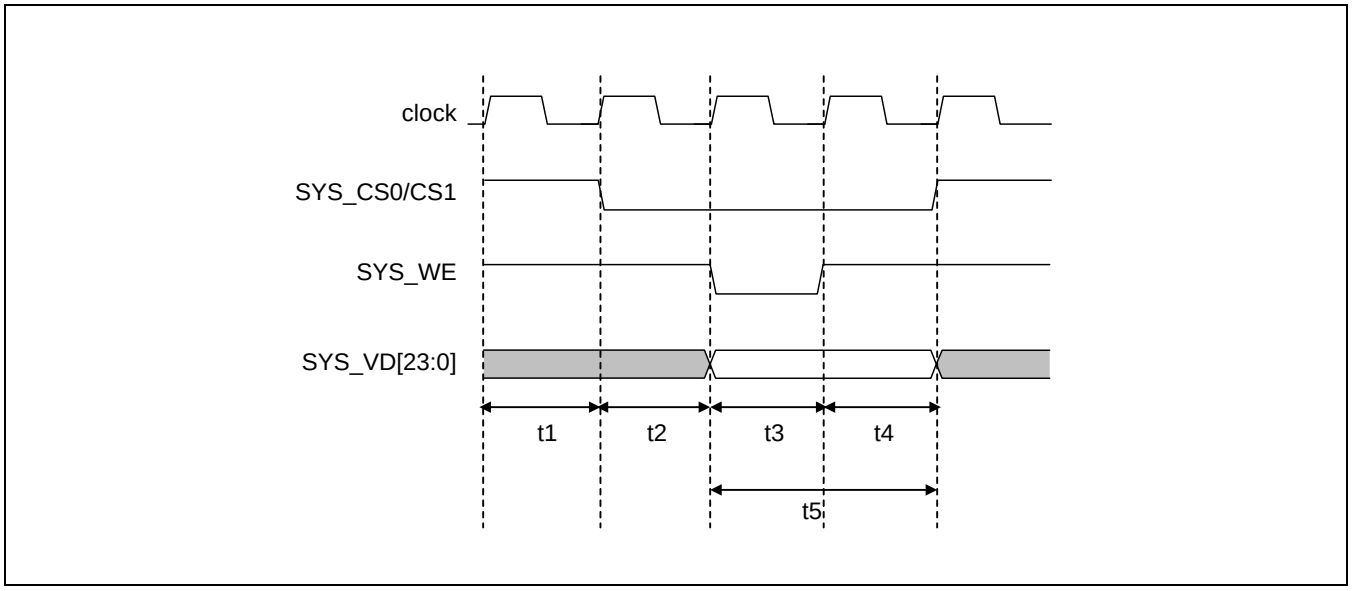


Figure 44-13 I80 Interface Timing Diagram

Minimum time should be:

- T1 ≥ 1 clock cycle.
- T2 ≥ 0 clock cycle.
- T3 ≥ 1 clock cycle.
- T4 ≥ 1 clock cycle.
- T5 ≥ 2 clock cycle.
- T2 + T3 + T4 > 1 cycle of byte clock

A display controller generates these signals with its internal clock: SYS_CS0/CS1, SYS_WE, and SYS_VD. MIPI-DSIM decodes the SYS_ADDR.

Table 44-2 describes the I80 INTERFACE address map.

Table 44-2 I80 Interface Address Map

SYS_ADDR[1:0]	Description
2'b00	Specifies the image data.
2'b01	Reserved
2'b10	Specifies the payload data.
2'b11	Specifies the packet header.

Figure 44-14 shows how MIPI-DSIM generates packet from the image data stream through I80 INTERFACE in command mode. MIPI-DSIM makes packet from the first line with "write_memory_start" Display Command Set (DCS) command and the other lines with "write_memory_continue" DCS command.

Figure 44-14 illustrates packetizing for MIPI DSI command mode from I80 interface.

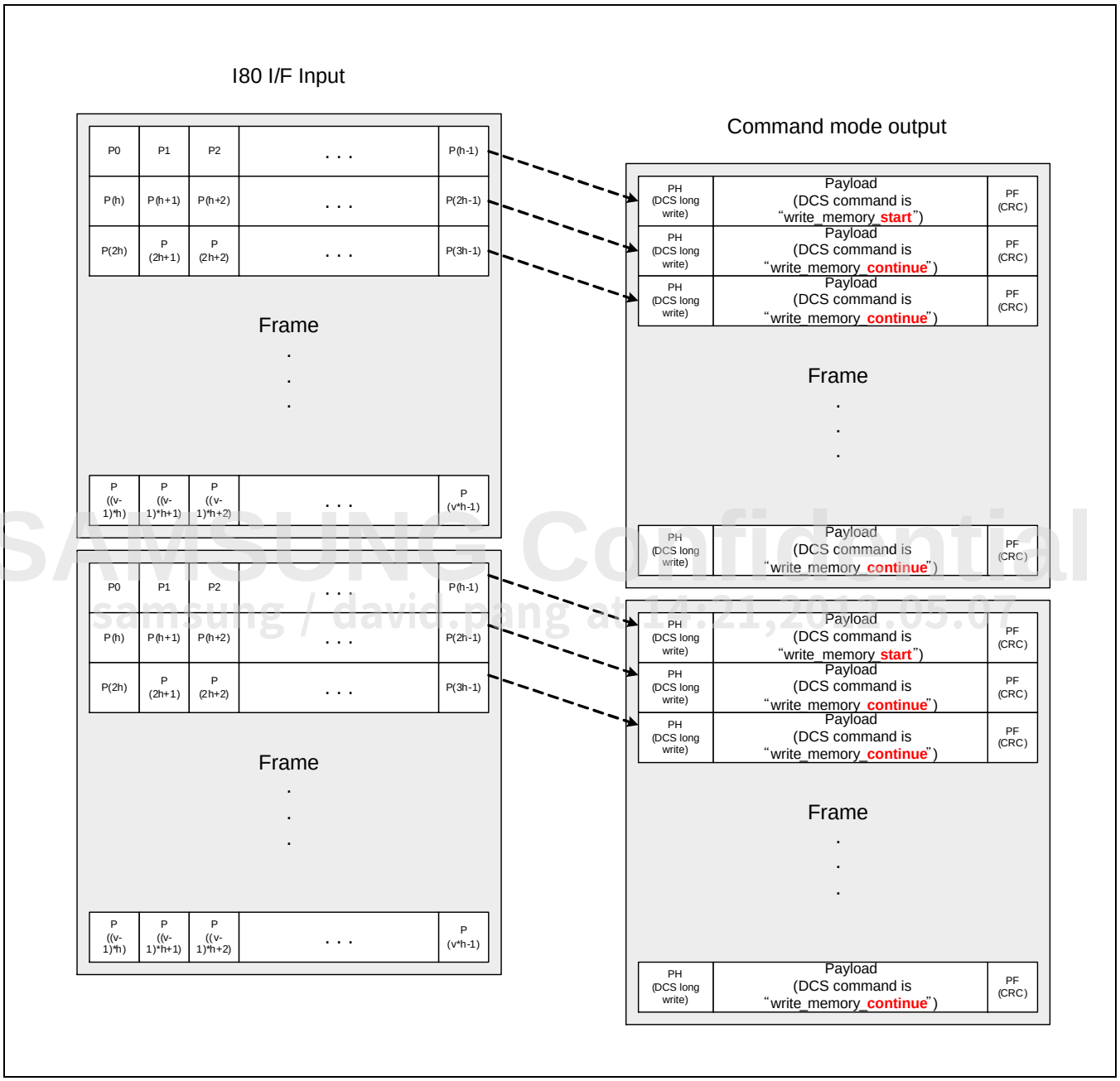


Figure 44-14 Packetizing for MIPI DSI Command Mode from I80 Interface

44.2.2.1.8 Relation between Input Transactions and DSI Transactions

Table 44-3 lists the relation between input transactions and DSI transactions.

Table 44-3 Relation between Input Transactions and DSI Transactions

Input Interface	Input Transaction	DSI Transaction
RGB	RGB transaction	Specifies the RGB Packet. You should specify 888, 666, 666 (loosely packed), and 565through register configuration.
I80	I80 Image Transaction	Specifies the Data type, that is, "DCS Long Write packet". (DCS command is "memory write start/continue".)
I80	I80 Command Transaction	Specifies any DSI packet. Bytes in I80 transaction should be the same bytes in DSI packets.
SFR	Header and Payload FIFO access	Specifies any DSI Packets. Bytes in APB transaction should be the same bytes in DSI packets.

44.2.3 Configuration

44.2.3.1 Video Mode versus Command Mode

MIPI DSI Master Block supports two modes, namely:

- Video mode
- Command mode

44.2.4 Dual Display versus Single Display

44.2.4.1 Dual Display

MIPI-DSIM supports dual display configuration in command mode only, that is, DSIM should transmit both main and sub-display image through I80 interface.

44.2.4.2 Single Display

For single display configuration, you should use video mode or command mode.

44.2.5 PLL

To transmit Image data, MIPI-DSIM Block requires high frequency clock (80 MHz to 1 GHz). PLL generates this high frequency clock.

PLL is embedded in PHY module. You should use other PLL in SoC if it meets the timing specification.

44.2.6 Buffer

In MIPI DSI standard specification, DSI Master sends image stream in burst mode. The image stream transmits in high-speed and bit-clock frequency. This mode enables the device to stay in stop state for longer duration to reduce power consumption. For this mode, MIPI DSIM has a dual line buffer to store one complete line and send it faster at the next line time.

44.3 I/O Description

Table 44-4 MIPI-DPHY Interface Slave Signal

Signal	I/O	Description	Pad	Type
MIPI_DP_0	B	Specifies the DP signal for MIPI-DPHY master data-lane 0.	XmipiDP[0]	Dedicated
MIPI_DN_0	B	Specifies the DN signal for MIPI-DPHY master data-lane 0.	XmipiDN[0]	Dedicated
MIPI_DP_1	O	Specifies the DP signal for MIPI-DPHY master data-lane 1.	XmipiDP[1]	Dedicated
MIPI_DN_1	O	Specifies the DN signal for MIPI-DPHY master data-lane 1.	XmipiDN[1]	Dedicated
MIPI_DP_2	O	Specifies the DP signal for MIPI-DPHY master data-lane 2.	XmipiDP[2]	Dedicated
MIPI_DN_2	O	Specifies the DN signal for MIPI-DPHY master data-lane 2.	XmipiDN[2]	Dedicated
MIPI_DP_3	O	Specifies the DP signal for MIPI-DPHY master data-lane 3.	XmipiDP[3]	Dedicated
MIPI_DN_3	O	Specifies the DN signal for MIPI-DPHY master data-lane 3.	XmipiDN[3]	Dedicated
MIPI_CLK_TX_P	O	Specifies the DP signal for MIPI-DPHY master clock-lane.	XmipiCLK_TX_P	Dedicated
MIPI_CLK_TX_N	O	Specifies the DN signal for MIPI-DPHY master clock-lane.	XmipiCLK_TX_N	Dedicated

NOTE:

1. I/O direction. I: input, O: output, and B: bi-direction.
2. Type field indicates whether the pads are dedicated to the signal or they are connected to the multiplexed signals.

44.4 Register Description

44.4.1 Register Map Summary

- Base Address: 0x11C8_0000

Register	Offset	Description	Reset Value
DSIM_STATUS	0x0000	Specifies the status register.	0x0010_0000
DSIM_SWRST	0x0004	Specifies the software reset register.	0x0000_0000
DSIM_CLKCTRL	0x0008	Specifies the clock control register.	0x0000_FFFF
DSIM_TIMEOUT	0x000C	Specifies the time out register.	0x00FF_FFFF
DSIM_CONFIG	0x0010	Specifies the configuration register.	0x0200_0000
DSIM_ESCMODE	0x0014	Specifies the escape mode register.	0x0000_0000
DSIM_MDRESOL	0x0018	Specifies the main display image resolution register.	0x0300_0400
DSIM_MVPORCH	0x001C	Specifies the main display VPORCH register.	0xF000_0000
DSIM_MHPORCH	0x0020	Specifies the main display HPORCH register.	0x0000_0000
DSIM_MSYNC	0x0024	Specifies the main display sync area register.	0x0000_0000
DSIM_SDRESOL	0x0028	Specifies the sub display image resolution register.	0x0300_0400
DSIM_INTSRC	0x002C	Specifies the interrupt source register.	0x0000_0000
DSIM_INTMSK	0x0030	Specifies the interrupt mask register.	0xB337_FFFF
DSIM_PKTHDR	0x0034	Specifies the packet header FIFO register.	0x0000_0000
DSIM_PAYLOAD	0x0038	Specifies the payload FIFO register.	0x0000_0000
DSIM_RXFIFO	0x003C	Specifies the read FIFO register.	0xXXXX_XXXX
DSIM_PLLCTRL	0x004C	Specifies the PLL control register.	0x0000_0000
DSIM_PLLTMR	0x0050	Specifies the PLL timer register.	0xFFFF_FFFF
DSIM_PHYACCHR	0x0054	Specifies the D-PHY AC characteristic register.	0x0000_0000
DSIM_PHYACCHR1	0x0058	Specifies the D-PHY AC characteristic register 1.	0x0000_0000

NOTE: M_RESETN at MIPI_PHY0_CONTROL (0x1002_0710) should be set to "1" before enabling DSIM0.

44.4.1.1 DSIM_STATUS

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0000, Reset Value = 0x0010_0000

Name	Bit	Type	Description	Reset Value
PllStable	[31]	R	D-PHY PLL generates stable byteclk.	0
RSVD	[30:21]	R	Reserved	0
SwRstRls	[20]	R	Specifies the software reset status. 0 = Reset state 1 = Release state	0
RSVD	[19:17]	R	Reserved	0
Direction	[16]	R	Specifies the data direction indicator. 0 = Forward direction 1 = Backward direction	1
RSVD	[15:11]	R	Reserved	0
TxReadyHsClk	[10]	R	Specifies the HS clock ready at clock lane. 0 = Not ready for transmitting HS data at clock lane 1 = Ready for transmitting HS data at clock lane	0
UlpsClk	[9]	R	Specifies the Ultra Low Power State (ULPS) indicator at clock lane. 0 = No ULPS in clock lane 1 = ULSP in clock lane	0
StopstateClk	[8]	R	Specifies the stop state indicator at clock lane. 0 = No-stop state in clock lane 1 = Stop state in clock lane	0
UlpsDat[3:0]	[7:4]	R	Specifies the ULPS indicator at data lanes. UlpsDat[0]: Data lane 0 UlpsDat[1]: Data lane 1 UlpsDat[2]: Data lane 2 UlpsDat[3]: Data lane 3 0 = No ULPS in each data lane 1 = ULPS in each data lane	0
StopstateDat[3:0]	[3:0]	R	Specifies the stop state indicator at data lane. StopstateDat[0]: Data lane 0StopstateDat[1]: Data lane 1 StopstateDat[2]: Data lane 2 StopstateDat[3]: Data lane 3 0 = No-stop state in each data lane 1 = Stop state in each data lane	0

44.4.1.2 DSIM_SWRST

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	RW	Reserved	–
FuncRst	[16]	RW	Specifies the software reset (High active). "Software reset" resets all FF in MIPI DSIM (except SFRs: STATUS, SWRST, CLKCTRL, TIMEOUT,CONFIG, ESCMODE (1), MDRESOL, MDVPORCH, MHPORCH, MSYNC, INTMSK, SDRESOL, FIFOTHLD, FIFOCTRL (2), MEMACCHR, PLLCTRL, PLLTMR, PHYACCHR, and VERINFORM). 0 = Standby 1 = Reset NOTE: 1. ForceStopstate, CmdLpdt, TxLpdt 2. Ninitrx, nInitSfr, nInitI80, nInitSub, nInitMD	0
RSVD	[15:1]	RW	Reserved	–
SwRst	[0]	RW	Specifies the software reset (High active). "Software reset" resets all FF in MIPI DSIM (except some SFRs: STATUS, SWRST, CLKCTRL, PLLCTRL, PLLTMR, and PHYTUNE). 0 = Standby 1 = Reset	0

44.4.1.3 DSIM_CLKCTRL

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_FFFF

Name	Bit	Type	Description	Reset Value
TxRequestHsClk	[31]	RW	Specifies the HS clock request for HS transfer at clock lane (Turn on HS clock)	0
RSVD	[30:29]	RW	Reserved	–
EscClkEn	[28]	RW	Enables the escape clock. 0 = Disables 1 = Enables	0
PLLBypass	[27]	RW	Sets the PLLBypass signal connected to D-PHY module input for selecting clock source bit. Refer to MIPI D-PHY specification for more information. 0 = PLL output 1 = External Serial clock	0
ByteClkSrc	[26:25]	RW	Selects byte clock source. (It must be 00.) 00 = D-PHY PLL (default). PLL_out clock is used to generate ByteClk by dividing 8.	0
ByteClkEn	[24]	RW	Enables byte clock. 0 = Disables 1 = Enables	0
LaneEscClkEn	[23:19]	RW	Enables escape clock for D-phy lane. LaneEscClkEn[0]: Clock lane LaneEscClkEn[1]: Data lane 0 LaneEscClkEn[2]: Data lane 1 LaneEscClkEn[3]: Data lane 2 LaneEscClkEn[4]: Data lane 3 0 = Disables 1 = Enables	0
RSVD	[18:16]	RW	Reserved	–
EscPrescaler	[15:0]	RW	Specifies the escape clock Prescaler value. The escape clock frequency range varies up to 20 MHz. NOTE: The requirement for Bus Turn Around (BTA) is that the Host Escclk frequency should range from 66.7 to 150 % of the peripheral escape clock frequency. EscClk = ByteClk/(EscPrescaler)	0xFFFF

44.4.1.4 DSIM_TIMEOUT

- Base Address: 0x11C8_0000
- Address = Base Address + 0x000C, Reset Value = 0x00FF_FFFF

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	RW	Reserved	–
BtaTout	[23:16]	RW	Specifies the timer for BTA. This register specifies the time-out from BTA request to change the direction with respect to Tx escape clock.	0xFF
LpdrTout	[15:0]	RW	Specifies the timer for LP Rx mode timeout. This register specifies the time-out on how long RxValid de-asserts after RxLpdt asserts with respect to Tx escape clock. RxValid specifies Rx data valid indicator. RxLpdt specifies an indicator that D-phy is under RxLpdt mode. RxValid and RxLpdt specifies signal from D-phy.	0xFFFF

44.4.1.5 DSIM_CONFIG

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0010, Reset Value = 0x0200_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	RW	Reserved	–
Mflush_VS	[29]	RW	Automatically flushes MD FIFO by using VSYNC pulse. Main display FIFO should be flushed for deleting garbage data. 0 = Enables (default) 1 = Disables	1
EoT_r03	[28]	RW	Disables EoT packet in HS mode. 0 = Enables EoT packet generation for V1.01r11 1 = Disables EoT packet generation for V1.01r03	0
SyncInform	[27]	RW	Selects either sync pulse or event mode in video mode. 0 = Event mode (non-burst, burst) 1 = Pulse mode (non-burst only) In command mode, this bit is ignored.	0
BurstMode	[26]	RW	Selects burst mode in video mode In non-burst mode, DSIM fills the RGB data area with RGB data and null packets according to input bandwidth of RGB interface. In burst mode, DSIM fills the RGB data area with RGB data only. 0 = Non-burst mode 1 = Burst mode In command mode, this bit is ignored.	0
VideoMode	[25]	RW	Specifies display configuration. 0 = Command mode 1 = Video mode	1
AutoMode	[24]	RW	Specifies auto vertical count mode. In video mode, the vertical line transition uses line counter. VSA, VBP, and Vertical resolution configured by VSA, VBP, and Vertical resolution. When this bit is set to "1", the line counter does not use VSA and VBP registers. 0 = Configuration mode 1 = Auto mode In command mode, this bit is ignored.	0
HseMode	[23]	RW	In Vsync pulse and Vporch area, MIPI DSIM transfers only Hsync start packet to MIPI DSI slave at MIPI DSI spec 1.1r02. This bit transfers Hsync end packet in Vsync pulse and Vporch area (optional). 0 = Disables transfer 1 = Enables transfer In command mode, this bit is ignored.	0

Name	Bit	Type	Description	Reset Value
HfpMode	[22]	RW	Specifies HFP disable mode. If this bit set, DSI master ignores HFP area in video mode. 0 = Enables 1 = Disables In command mode, this bit is ignored.	0
HbpMode	[21]	RW	Specifies HBP disable mode. If this bit set, DSI master ignores HBP area in Video mode. 0 = Enables 1 = Disables In command mode, this bit is ignored.	0
HsaMode	[20]	RW	Specifies HSA disable mode. If this bit set, DSI master ignores HSA area in Video mode. 0 = Enables 1 = Disables In command mode, this bit is ignored.	0
MainVc	[19:18]	RW	Specifies virtual channel number for main display.	0
SubVc	[17:16]	RW	Specifies virtual channel number for sub display.	0
RSVD	[15]	RW	Reserved	–
MainPixFormat	[14:12]	RW	Specifies pixel stream format for main display. 000 = 3 bpp (for command mode only) 001 = 8 bpp (for command mode only) 010 = 12 bpp (for command mode only) 011 = 16 bpp (for command mode only) 100 = 16-bit RGB (565) (for video mode only) 101 = 18-bit RGB (666: packed pixel stream) (for video mode only) 110 = 18-bit RGB (666: loosely packed pixel stream) for common 111 = 24-bit RGB (888) for common	0
RSVD	[11]	RW	Reserved	–
SubPixFormat	[10:8]	RW	Specifies pixel stream format for sub display. 000 = 3 bpp (for command mode only) 001 = 8 bpp (for command mode only) 010 = 12 bpp (for command mode only) 011 = 16 bpp (for command mode only) 100 = 16-bit RGB (565) (for video mode only) 101 = 18-bit RGB (666: packed pixel stream) for video mode only) 110 = 18-bit RGB (666: loosely packed pixel stream) for common) 111 = 24-bit RGB (888) (for common)	0
RSVD	[7]	RW	Reserved	–
NumOfDatLane	[6:5]	RW	Sets the data lane number. 00 = Data lane 0 (1 data lane) 01 = Data lane 0 to 1 (2 data lanes) 10 = Data lane 0 to 2 (3 data lanes)	0

Name	Bit	Type	Description	Reset Value
			11 = Data lane 0 to 3 (4 data lanes)	
RSVD	[4]	RW	Reserved	–
LaneEn[3:0]	[3:0]	RW	Enables the lane. If Lane_EN is disabled, the lane ignores input and drives initial value through output port. 0 = Lane is off. 1 = Lane is on. LaneEn[0]: Clock lane enabler LaneEn[1]: Data lane 0 enabler LaneEn[2]: Data lane 1 enabler LaneEn[3]: Data lane 2 enabler LaneEn[4]: Data lane 3 enabler	0

44.4.1.6 DSIM_ESCMODE

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
STOPstate_Cnt	[31:21]	RW	After transmitting Read packet or Write "set_tear_on" command, BTA requests to D-phy automatically. This counter value specifies the interval value between transmitting Read packet (or write "set_tear_on" command) and BTA request. 11'h000 = 2 EscClk 11'h001 = 2 EscClk + 1 EscClk to 11'h3FF = 2 EscClk + 1023 EscClk	0
ForceStopstate	[20]	RW	Forces Stopstate for D-PHY.	0
RSVD	[19:17]	RW	Reserved	–
ForceBta	[16]	RW	Forces Bus Turn Around. 1 = Sends the protocol layer request to D-PHY. MIPI DSI peripheral becomes master after BTA sequence. This bit automatically clears after receiving BTA acknowledge from MIPI DSI peripheral.	0
RSVD	[15:8]	RW	Reserved	–
CmdLpdt	[7]	RW	Specifies LPDT transfers command in SFR FIFO. 0 = HS Mode 1 = LP Mode	0
TxLpdt	[6]	RW	Specifies data transmission in LP mode (all data transfer in LPDT). 0 = HS Mode 1 = LP Mode	0
RSVD	[5]	RW	Reserved	–
TxTriggerRst	[4]	RW	Specifies remote reset trigger function. After trigger operation, these bits will be cleared automatically.	0
TxUlpsDat	[3]	RW	Specifies ULPS request for data lane. Manually clears after ULPS exit.	0
TxUlpsExit	[2]	RW	Specifies ULPS exit request for data lane. Manually clears after ULPS exit.	0
TxUlpsClk	[1]	RW	Specifies ULPS request for clock lane. Manually clears after ULPS exit.	0
TxUlpsClkExit	[0]	RW	Specifies ULPS exit request for clock lane. Manually clears after ULPS exit.	0

44.4.1.7 DSIM_MDRESOL

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0018, Reset Value = 0x0300_0400

Name	Bit	Type	Description	Reset Value
MainStandby	[31]	RW	Specifies standby for receiving DISPCON output in command mode after setting all configuration. 0 = Not ready 1 = Stand by This bit should be set after configuration (resolution, reqtype, pixelform, and so on). In video mode, when this bit value is set to 0, DSIM does not transfer the data.	0
RSVD	[30:27]	RW	Reserved	–
MainVResol[10:0]	[26:16]	RW	Specifies Vertical resolution (1 to 1024).	0x300
RSVD	[15:11]	RW	Reserved	–
MainHResol[10:0]	[10:0]	RW	Specifies Horizontal resolution (1 to 1024).	0x400

44.4.1.8 DSIM_MVPORCH

- Base Address: 0x11C8_0000
- Address = Base Address + 0x001C, Reset Value = 0xF000_0000

Name	Bit	Type	Description	Reset Value
CmdAllow	[31:28]	RW	Specifies the number of horizontal lines, where DSIM allows command packet transmission after Stable VFP period. Refer to *Figure 44-12* for more information.	0xF
RSVD	[27]	RW	Reserved	–
StableVfp[10:0]	[26:16]	RW	Specifies the number of horizontal lines, where DSIM does not allow command packet transmission after end of active region. Refer to *Figure 44-12* for more information. NOTE: In command mode, DSIM ignores these bits.	0
RSVD	[15:11]	RW	Reserved	–
MainVbp[10:0]	[10:0]	RW	Specifies vertical back porch width for video mode (line count). In command mode, DSIM ignores these bits.	0

NOTE: Transfers command packets after Stable VFP area. VFP lines should be set based on sum of these values: Stable VFP, command allowing area, and command masked area. Refer to the section, Transfer General Data in Video Mode, for more information.

NOTE:

44.4.1.9 DSIM_MHPORCH

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MainHfp[15:0]	[31:16]	RW	Specifies the horizontal front porch width for video mode. HFP is specified by using blank packet. These bits specify the word counts for blank packet in HFP. In command mode, these bits are ignored.	0
MainHbp[15:0]	[15:0]	RW	Specifies the horizontal back porch width for video mode. HBP is specified by using blank packet. These bits specify the word counts for blank packet in HBP. In command mode, these bits are ignored.	0

44.4.1.10 DSIM_MSYNC

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MainVsa[9:0]	[31:22]	RW	Specifies the vertical sync pulse width for video mode (Line count). In command mode, these bits are ignored.	0
RSVD	[21:16]	RW	Reserved	–
MianHsa[15:0]	[15:0]	RW	Specifies the horizontal sync pulse width for video mode. HSA is specified by using blank packet. These bits specify word counts for blank packet in HSA. In command mode, these bits are ignored.	0

44.4.1.11 DSIM_SDRESOL

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0028, Reset Value = 0x0300_0400

Name	Bit	Type	Description	Reset Value
SubStandby	[31]	RW	Specifies standby for receiving DISPCON output in command mode after setting all configuration. 0 = Not ready 1 = Standby This bit should be set after configuration (resolution, reqtype, pixelform, and so on) for command mode. In video mode, this bit is ignored.	0
RSVD	[30:27]	RW	Reserved	–
SubVResol[10:0]	[26:16]	RW	Specifies the Vertical resolution (1 to 1024).	0x300
RSVD	[15:11]	RW	Reserved	–
SubHResol[10:0]	[10:0]	RW	Specifies the Horizontal resolution (1 to 1024).	0x400

44.4.1.12 DSIM_INTSRC

- Base Address: 0x11C8_0000
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PllStable	[31]	RW	Indicates that D-phy PLL is stable.	0
SwRstRelease	[30]	RW	Releases the software reset.	0
SFRFifoEmpty	[29]	RW	Specifies the SFR payload FIFO empty.	0
SyncOverride	[28]	RW	Indicates that other DSI command transfer have overridden sync timing.	0
RSVD	[27:26]	RW	Reserved	–
BusTurnOver	[25]	RW	Indicates when bus grant turns over from DSI slave to DSI master.	0
FrameDone	[24]	RW	Indicates when MIPI DSIM transfers the whole image frame. NOTE: If DSIM does not receive Hsync during two-line times, it times out internal timer and flags this bit.	0
RSVD	[23:22]	RW	Reserved	–
LpdrTout	[21]	RW	Specifies the LP Rx timeout. Refer to Time out register (0x10) for more information.	0
TaTout	[20]	RW	Turns around Acknowledge Timeout. Refer to Time out register (0x10) for more information.	0
RSVD	[19]	RW	Reserved	–
RxDatDone	[18]	RW	Completes receiving data.	0
RxTE	[17]	RW	Receives TE Rx trigger.	0
RxAck	[16]	RW	Receives ACK Rx trigger.	0
ErrRxECC	[15]	RW	Specifies the ECC multi bit error in LPDR.	0
ErrRxCRC	[14]	RW	Specifies the CRC error in LPDR.	0
ErrEsc3	[13]	RW	Specifies the escape mode entry error lane 3. Refer to standard D-PHY specification for more information.	0
ErrEsc2	[12]	RW	Specifies the escape mode entry error lane 2. Refer to standard D-PHY specification.	0
ErrEsc1	[11]	RW	Specifies the escape mode entry error lane 1. Refer to standard D-PHY specification for more information.	0
ErrEsc0	[10]	RW	Specifies the escape mode entry error lane 0. Refer to standard D-PHY specification for more information.	0
ErrSync2	[9]	RW	Specifies the LPDT sync error lane 3. Refer to standard D-PHY specification for more information.	0
ErrSync2	[8]	RW	Specifies the LPDT sync error lane2. Refer to standard D-PHY specification for more information.	0
ErrSync1	[7]	RW	Specifies the LPDT sync error lane1. Refer to standard D-PHY specification for more information.	0

Name	Bit	Type	Description	Reset Value
ErrSync0	[6]	RW	Specifies the LPDT sync error lane0. Refer to standard D-PHY specification for more information.	0
ErrControl2	[5]	RW	Controls error lane3. Refer to standard D-PHY specification for more information.	0
ErrControl2	[4]	RW	Controls error lane2. Refer to standard D-PHY specification for more information.	0
ErrControl1	[3]	RW	Controls error lane1. Refer to standard D-PHY specification for more information.	0
ErrControl0	[2]	RW	Controls error lane0. Refer to standard D-PHY specification for more information.	0
ErrContentLP0	[1]	RW	Specifies the LP0 contention error (only lane0, because BTA occurs at lane0 only). Refer to standard D-PHY specification for more information.	0
ErrContentLP1	[0]	RW	Specifies the LP1 contention error (only lane0, because BTA occurs at lane0 only). Refer to standard D-PHY specification for more information.	0

This register identifies interrupt sources such as internal block error; data transmit interrupt, inter-layer (D_PHY) error, and so on. The bits are set even if DSIM_INTMSK_REG masks them off. To clear these bits, write "1".

44.4.1.13 DSIM_INTMSK

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0030, Reset Value = 0xB337_FFFF

Name	Bit	Type	Description	Reset Value
MskPllStable	[31]	RW	Indicates that D-PHY PLL is stable.	–
MskSwRstRelease	[30]	RW	Releases software reset.	0
MskSFRFifoEmpty	[29]	RW	Empties SFR payload FIFO.	1
RSVD	[28:26]	RW	Reserved	–
MskBusTurnOver	[25]	RW	Indicates when bus grant turns over from DSI slave to DSI master.	1
MskFrameDone	[24]	RW	Indicates when MIPI DSIM transfers whole image frame.	1
RSVD	[23:22]	RW	Reserved	–
MskLpdrTout	[21]	RW	Specifies LP Rx timeout. Refer to time out register (0x10) for more information.	1
MskTaTout	[20]	RW	Specifies turnaround acknowledge timeout. Refer to time out register (0x10) for more information.	1
RSVD	[19]	RW	Reserved	–
MskRxDatDone	[18]	RW	Specifies completion of data receiving.	1
MskRxTE	[17]	RW	Specifies receipt of TE Rx trigger.	1
MskRxAck	[16]	RW	Specifies receipt of ACK Rx trigger.	1
MskRxECC	[15]	RW	Specifies ECC multibit error in LPDR.	1
MskRxCRC	[14]	RW	Specifies CRC error in LPDR.	1
MskEsc3	[13]	RW	Specifies escape mode entry error in lane3. Refer to standard D-PHY specification for more information.	1
MskEsc2	[12]	RW	Specifies escape mode entry error in lane2. Refer to standard D-PHY specification for more information.	1
MskEsc1	[11]	RW	Specifies escape mode entry error in lane1. Refer to standard D-PHY specification for more information.	1
MskEsc0	[10]	RW	Specifies escape mode entry error in lane0. Refer to standard D-PHY specification for more information.	1
MskSync3	[9]	RW	Specifies LPDT sync error in lane3. Refer to standard D-PHY specification for more information.	1
MskSync2	[8]	RW	Specifies LPDT sync error in lane2. Refer to standard D-PHY specification for more information.	1
MskSync1	[7]	RW	Specifies LPDT sync error in lane1. Refer to standard D-PHY specification for more information.	1
MskSync0	[6]	RW	Specifies LPDT sync error in lane0. Refer to standard D-PHY specification for more information.	1
MskControl3	[4]	RW	Controls error in lane3. Refer to standard D-PHY specification for more information.	1

Name	Bit	Type	Description	Reset Value
MskControl2	[4]	RW	Controls error in lane2. Refer to standard D-PHY specification for more information.	1
MskControl1	[3]	RW	Controls error in lane1. Refer to standard D-PHY specification for more information.	1
MskControl0	[2]	RW	Controls error in lane0. Refer to standard D-PHY specification for more information.	1
MskContentLP0	[1]	RW	Specifies LP0 contention error. Refer to standard D-PHY specification for more information.	1
MskContentLP1	[0]	RW	Specifies LP1 contention error. Refer to standard D-PHY specification for more information.	1

44.4.1.14 DSIM_PKTHDR

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	W	Reserved	–
PacketHeader	[23:0]	W	Writes the packet header of Tx packet. [7:0]: DI [15:8]: Dat0 (Word count lower byte for long packet) [23:16]: Dat1 (Word count upper byte for long packet)	0

44.4.1.15 DSIM_PAYLOAD

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0038, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Payload	[31:0]	W	Writes the Payload of Tx packet.	0

44.4.1.16 DSIM_RXFIFO

- Base Address: 0x11C8_0000
- Address = Base Address + 0x003C, Reset Value = 0xXXXX_XXXX

Name	Bit	Type	Description	Reset Value
RxDat	[31:0]	R	In the Rx mode, you can read Rx data through this register. NOTE: DSIM does not store CRC in packet in RxFIFO.	Unknown

44.4.1.17 DSIM_PLLCTRL

- Base Address: 0x11C8_0000
- Address = Base Address + 0x004C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:28]	RW	Reserved. Should be set to 0.	–
FreqBand[3:0]	[27:24]	RW	Indicates bitclk frequency band for D-PHY global timing. Refer to *Table 44-7* for more information.	0
PllEn	[23]	RW	Enables PLL.	0
RSVD	[22:19]	RW	Reserved. Should be 0.	–
P	[18:13]	RW	Specifies the PLL P value. (1 to 63)	0
M	[12:4]	RW	Specifies the PLL M value. (20 to 511)	0
S	[3:1]	RW	Specifies the PLL S value. (0 to 5)	0
RSVD	[0]	RW	Reserved	0

44.4.1.18 DSIM_PLLTMR

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0050, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
PllTimer	[31:0]	RW	Specifies the PLL Timer for stability of the generated clock (System clock cycle base). When the timer value reaches 0x00000000, DSIM sets the clock stable bit of status and interrupt register.	0xFFFFFFFF

44.4.1.19 DSIM_PHYACCHR

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:15]	RW	Reserved. Should be set to 0.	0
AFC_EN	[14]	RW	Enables AFC. 0 = Disables 1 = Enables	0
RSVD	[13:8]	RW	Reserved. Should be set to 0.	0
AFC_CTL	[7:5]	RW	Specifies the AFC control value for MIPI DPHY. This value is meaningful when AFC_EN is set to 1. Refer to *Table 44-6* for more information.	0
RSVD	[4:0]	RW	Reserved. Should be set to 0.	0

44.4.1.20 DSIM_PHYACCCHR1

- Base Address: 0x11C8_0000
- Address = Base Address + 0x0058, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	RW	Reserved. Should be set to 0.	0
RSVD	[6:4]	RW	Reserved	0
RSVD	[3:2]	RW	Reserved	0
DpDnSwap_CLK	[1]	RW	Swaps Dp/Dn channel of clock lane. When you set this bit, Dp and Dn channel swap each other.	0
DpDnSwap_DAT	[0]	RW	Swaps Dp/Dn channel of Data lanes. When you set this bit, Dp and Dn channel swap each other.	0

44.5 DPHY PLL Control

DSIM has internal PLL:

44.5.1 PMS Setting Sample for MIPI PLL

This section includes:

- PMS setting
- Sample for F_{out} 80 MHz
- Sample for F_{out} 1000 MHz
- Sample for F_{out} 999 MHz

44.5.1.1 PMS Setting

Writes a value to PMS field in DSIM_PLLCTRL (0x11C8_004C) to set PMS value for DPHY PLL. Each P, M, and S resides in PMS [18:13], PMS [12:4] and PMS [3:1].

Before setting the PMS value, follow these instructions:

1. Do not set the P or M value as 0 because setting the P (00000) or M (0000000000) causes malfunction of the PLL.
2. Choose the selected M value within the range of 41 to 125 for PLL stability. The VCO output frequency range of MIPI_PLL varies from 500 MHz to 1000 MHz
3. Fin_pll frequency range varies from 6 MHz to 12 MHz.

Table 44-5 describes PMS and frequency constraint.

Table 44-5 PMS and Frequency Constraint

–	Function	Value	Description
Fin	Fin	6 to 200 MHz	Specifies PLL input frequency.
Fin_pll	Fin/P	6 to 12 MHz	Specifies PFD input frequency.
VCO_out	(M × Fin)/P	500 to 1000 MHz	Specifies VCO output frequency.
Fout	(M × Fin)/(P × 2^S)	15.625 to 1000 MHz	Specifies PLL output frequency.
P[5:0]	P	1 to 63	Specifies PMS[18:13].
M[8:0]	M	20 to 511	Specifies PMS[12:4].
S[2:0]	2^S	1, 2, 4, 8, 16, 32	Specifies PMS[3:1].

Table 44-6 lists AFC codes.

Table 44-6 AFC Codes

Serial Clock	AFC_CTL[2:0]
6 to 6.99 MHz	001
7 to 7.99 MHz	000
8 to 8.99 MHz	011
9 to 9.99 MHz	010
10 to 10.99 MHz	101
11 to 12 MHz	100

Table 44-7 lists band control setting.

Table 44-7 Band Control Setting

Serial Clock (= ByteClk × 8)	FreqBand[3:0]
to 99.99 MHz	0000
100 to 119.99 MHz	0001
120 to 159.99 MHz	0010
160 to 199.99 MHz	0011
200 to 239.99 MHz	0100
140 to 319.99 MHz	0101
320 to 389.99 MHz	0110
390 to 449.99 MHz	0111
450 to 509.99 MHz	1000
510 to 559.99 MHz	1001
560 to 639.99 MHz	1010
640 to 689.99 MHz	1011
690 to 769.99 MHz	1100
770 to 869.99 MHz	1101
870 to 949.99 MHz	1110
950 to 1000 MHz	1111

44.5.1.2 Sample for F_{out} 80 MHz

Refer to *Table 44-8* to set PMS value for F_{out} 80 MHz

Table 44-8 PMS value for 80 MHz

–	Case 1	Case 2
Fin	24 MHz	30 MHz
Fin_pll	8 MHz	10 MHz
P[5:0]	3	3
M[8:0]	80	64
S[2:0]	8	8
VCO_out	640 MHz	640 MHz
Fout	80 MHz	80 MHz

44.5.1.3 Sample for F_{out} 1000 MHz

Refer to *Table 44-9* to set PMS value for F_{out} 1000 MHz.

Table 44-9 PMS value for 1000 MHz

–	Case 1	Case 2
Fin	24 MHz	30 MHz
Fin_pll	8 MHz	10 MHz
P[5:0]	3	3
M[9:0]	125	100
S[2:0]	1	1
VCO_out	1000 MHz	1000 MHz
Fout	1000 MHz	1000 MHz

44.5.1.4 Sample for F_{out} 999 MHz

Refer to *Table 44-10* to set PMS value for F_{out} 999 MHz.

Table 44-10 PMS value for 999 MHz

–	Case 1	Case 2
Fin	27 MHz	9 MHz
Fin_pll	9 MHz	9 MHz
P[5:0]	3	1
M[9:0]	111	111
S[2:0]	1	1
VCO_out	999 MHz	999 MHz
Fout	999 MHz	999 MHz

45 MIPI-CSI Slave (MIPI-CSI)

45.1 Overview of MIPI CSIS

Exynos 4412 SCP has two MIPI CSIS units. The two units are:

- CSIS0
- CSIS1

CSIS0 controls MIPI_DPHY_4L (four lane mipi-dphy) and CSIS1 controls MIPI_DPHY_2L (two lane mipi-dphy).

45.2 Features

The features of MIPI CSIS are:

- Compliance to MIPI CSI2 standard specification version 1.0
  - CSIS0 supports 1, 2, 3, or 4 data lanes.
  - CSIS1 supports 1 or 2 data lanes.
  - Supports one channel.
  - Supports RAW8, RAW10, RAW12, RAW14, and YUV422 8-bit.
  - Supports all user defined byte-based data packet.
- Interfaces
  - Compatible with Protocol-to-PHY Interface (PPI) in MIPI D-PHY specification version 0.90.

45.3 Block Diagram

Figure 45-1 illustrates the CSIS0 system block diagram.

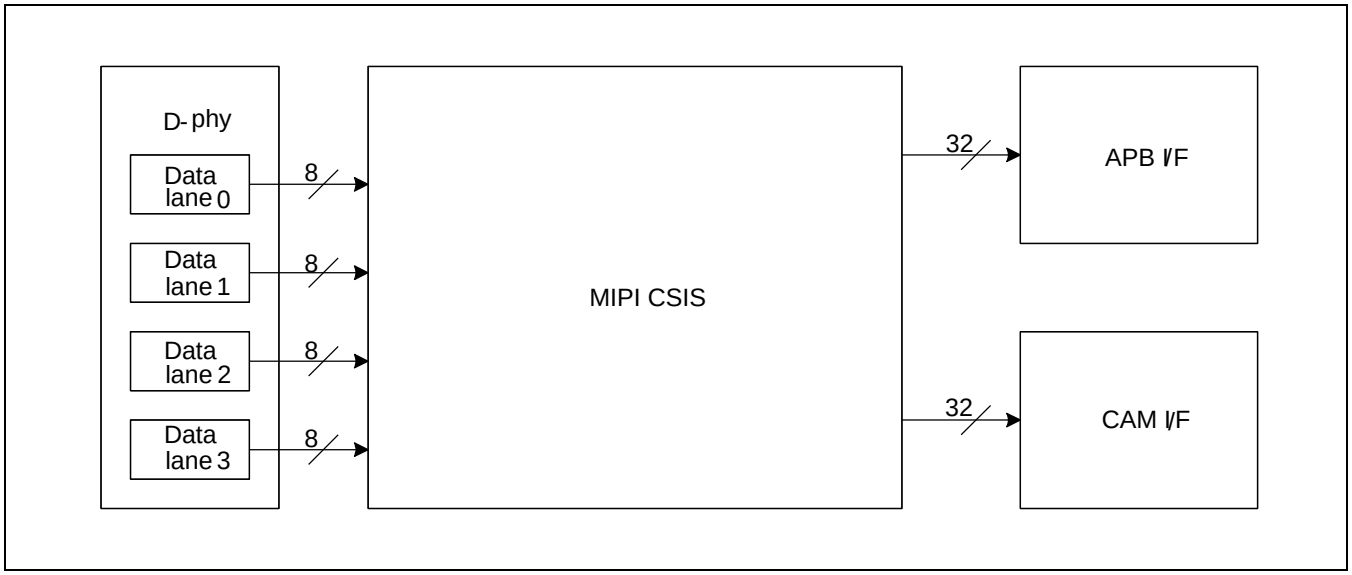


Figure 45-1 CSIS0 System Block Diagram

45.4 Interface and Protocol

Figure 45-2 illustrates the waveform of output data.

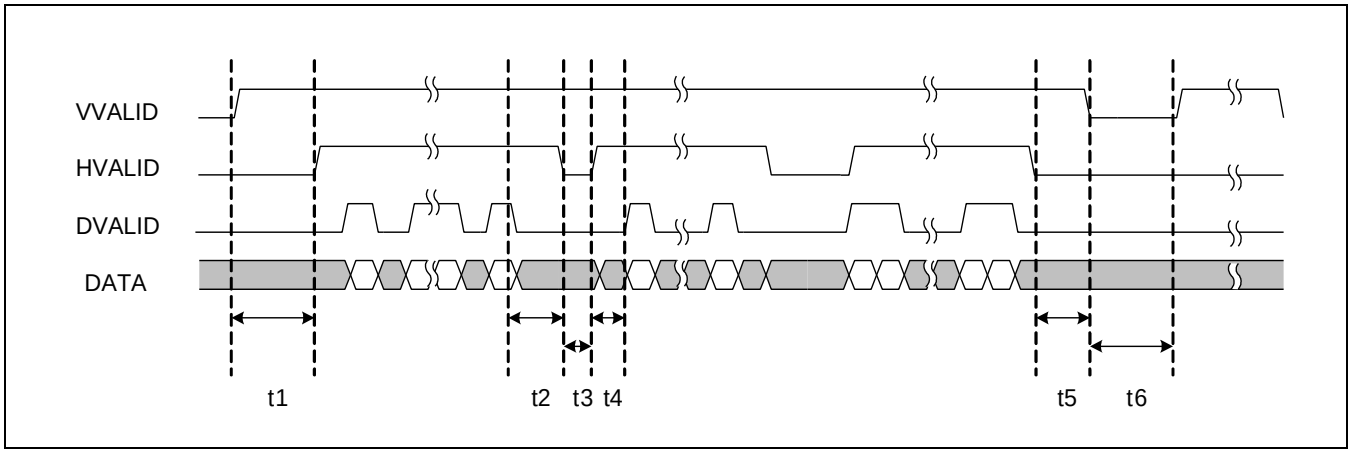


Figure 45-2 Waveform of Output Data

Table 45-1 describes the timing diagram of output data.

Table 45-1 Timing Diagram of Output Data

Symbol	Description	Minimum Cycle of Pixel Clock
t1	Specifies interval between rising of VVALID and first rising of HVALID.	Vsync_SIntv + 1 (1 to 64)
t2	Specifies interval between last falling of DVALID and falling of HVALID.	Hsync_LIntv + 2 (2 to 66)
t3	Specifies interval between falling of HVALID and rising of next HVALID.	1
t4	Specifies interval between rising of HVALID and first rising of DVALID.	0
t5	Specifies interval between last falling of HVALID and falling of VVALID.	Vsync_EIntv (0 to 4095)
t6	Specifies interval between falling of VVALID and rising of next VVALID.	1

45.5 Data Format

Data Format section includes:

- Data Alignment
- YUV422 8-bit Order

45.5.1 Data Alignment

CSIS supports two type of data alignment, as illustrated in *Figure 45-3*.

Figure 45-3 illustrates the MIPI CSIS data alignment.

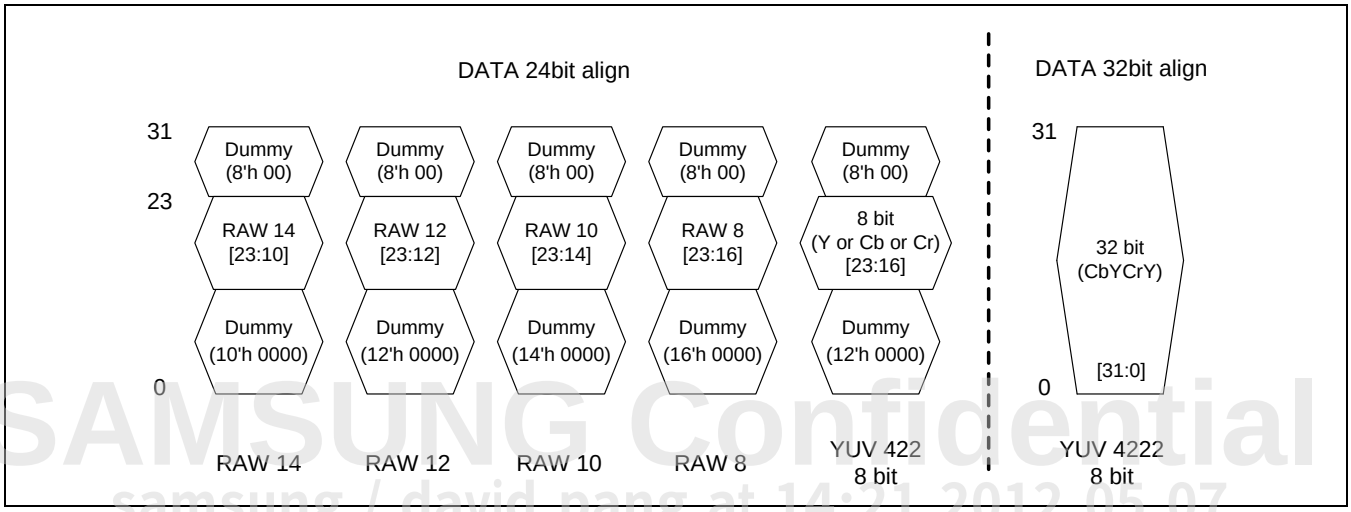


Figure 45-3 MIPI CSIS Data Alignment

45.5.2 YUV422 8-bit Order

CSIS stores YUV422 8-bit format data as a UYVY sequence.

Table 45-2 describes the data order of YUV422 alignment.

Table 45-2 Data Order of YUV422 Alignment

Format	Stream Order of content	24-bit Alignment	32-bit Alignment
YUV422 8-bit	U1 → Y1 → V1 → Y2 → ...	DATA1[23:16] = U1	DATA1[31:24] = U1
		DATA2[23:16] = Y1	DATA1[23:16] = Y1
		DATA3[23:16] = V1	DATA1[15:8] = V1
		DATA4[23:16] = Y2	DATA1[7:0] = Y2

45.6 I/O Description

Signal	I/O	Description	Pad	Type
DPDATA0_4L	B	Specifies DP signal for MIPI-DPHY_4L slave data-lane 0.	XmipiSDP0	Dedicated
DNDATA0_4L	B	Specifies DN signal for MIPI-DPHY_4L slave data-lane 0.	XmipiSDN0	Dedicated
DPDATA1_4L	B	Specifies DP signal for MIPI-DPHY_4L slave data-lane 1.	XmipiSDP1	Dedicated
DNDATA1_4L	B	Specifies DN signal for MIPI-DPHY_4L slave data-lane 1.	XmipiSDN1	Dedicated
DPDATA2_4L	B	Specifies DP signal for MIPI-DPHY_4L slave data-lane 2.	XmipiSDP2	Dedicated
DNDATA2_4L	B	Specifies DN signal for MIPI-DPHY_4L slave data-lane 2.	XmipiSDN2	Dedicated
DPDATA3_4L	B	Specifies DP signal for MIPI-DPHY_4L slave data-lane 3.	XmipiSDP3	Dedicated
DNDATA3_4L	B	Specifies DN signal for MIPI-DPHY_4L slave data-lane 3.	XmipiSDN3	Dedicated
DPCLK_4L	B	Specifies DP signal for MIPI-DPHY_4L slave clock-lane.	XmipiSDPCLK	Dedicated
DNCLK_4L	B	Specifies DN signal for MIPI-DPHY_4L slave clock-lane.	XmipiSDNCLK	Dedicated
DPDATA0_2L	B	Specifies DP signal for MIPI-DPHY_2L slave data-lane 0.	Xmipi2LSDP0	Dedicated
DNDATA0_2L	B	Specifies DN signal for MIPI-DPHY_2L slave data-lane 0.	Xmipi2LSDN0	Dedicated
DPDATA1_2L	B	Specifies DP signal for MIPI-DPHY_2L slave data-lane 1.	Xmipi2LSDP1	Dedicated
DNDATA1_2L	B	Specifies the DN signal for MIPI-DPHY_2L slave data-lane 1.	Xmipi2LSDN1	Dedicated
DPCLK_2L	B	Specifies DP signal for MIPI-DPHY_2L slave clock-lane.	Xmipi2LSDPCLK	Dedicated
DNCLK_2L	B	Specifies DN signal for MIPI-DPHY_2L slave clock-lane.	Xmipi2LSDNCLK	Dedicated

NOTE:

1. I/O direction. I = input, O = output, and B = bi-direction.
2. Type field indicates whether it dedicates pads to signal or connects to the multiplexed signals.

45.7 Register Description

45.7.1 Register Map Summary

- Base Address: 0x1188_0000
- Base Address: 0x1189_0000

Register	Offset	Description	Reset Value
CSIS_CONTROLn	0x0000	Specifies control register.	0x0010_0000
CSIS_DPHYCTRLn	0x0004	Specifies D-PHY control register.	0x0000_0000
CSIS_CONFIGn	0x0008	Specifies configuration register.	0x0000_0000
CSIS_DPHYSTSn	0x000C	Specifies D-PHY stop state register.	0x0000_00F1
CSIS_INTMSKn	0x0010	Specifies interrupt mask register.	0x0000_0000
CSIS_INTSRCn	0x0014	Specifies interrupt status register.	0x0000_0000
CSIS_RESOLn	0x002C	Specifies image resolution register.	0x8000_8000
SDW_CONFIGn	0x0038	Specifies shadow register of configuration.	0x0000_0000
SDW_RESOLn	0x003C	Specifies shadow register of resolution.	0x8000_8000
CSIS_PKTDATAn	0x2000 to 0x3FFC	Specifies memory area for storing non-image data. Odd frame = 0x2000 to 0x2FFC Even frame = 0x3000 to 0x3FFC	0xXXXX_XXXX

NOTE: Before enabling CSIS0, S_RESETN at MIPI_PHY0_CONTROL (0x1002_0710) should be set to "1" Before enabling CSIS1, S_RESETN at MIPI_PHY1_CONTRL (0x1002_0714) should be set to "1".

45.7.1.1 CSIS_CONTROLn (n = 0 to 1, Control Register)

- Base Address: 0x1188_0000, 0x1189_0000
- Address = Base Address + 0x0000, Reset Value = 0x0010_0000

Name	Bit	Type	Description	Reset Value
S_DpDn_Swap_Clk	[31]	RW	Swaps Dp channel and Dn channel of clock lanes. 0 = Default 1 = Swaps the channels.	0
S_DpDn_Swap_Dat	[30]	RW	Swaps Dp channel and Dn channel of data lanes. 0 = Default 1 = Swaps the channel.	0
RSVD	[29:21]	–	Reserved. NOTE: This bit should be set to 0.	0
Parallel	[20]	RW	Specifies data alignment size. Refer to *45.5 Data Format* 0 = Specifies 24-bit data alignment 1 = Specifies 32-bit data alignment Note : If data alignment is set to 24-bit, data format only supports RAW.	1
RSVD	[19:17]	–	Reserved. Should be 0.	0
Update_Shadow	[16]	RW	Updates the shadow registers. 0 = Default 1 = Updates the shadow registers You should set this bit for updating shadow registers after configuration. Clears this bit automatically after updating shadow registers.	0
Reserved	[15:9]	–	Reserved. NOTE: This bit should be set to 0	0
WCLK_Src	[8]	RW	Specifies wrapper clock source. 0 = Specifies PCLK clock (PCLK is APB clock for CSIS) 1 = Specifies SCLK_CSIS clock (SCLK_CSIS is CSIS operating clock generated from CMU) This bit determines the source of pixel clock, which transfers image data to CAMIF.	0
RSVD	[7:5]	–	Reserved. NOTE: This bit should be set to 0	0
SwRst	[4]	RW	Specifies software reset. 0 = No reset 1 = Resets the sofware. All writable registers in CSIS returns to their reset value. This bit de-asserts automatically after being active for three cycles. NOTE: Almost all MIPI CSIS blocks use "ByteClk" from D-PHY. "ByteClk" is not a continuous clock.	0

Name	Bit	Type	Description	Reset Value
			You should assert software reset if you turn the camera module off .	
RSVD	[3:1]	–	Reserved.	0
Enable	[0]	RW	Specifies the CSIS system on/ off. 0 = Specifies system is Off 1 = Specifies system is On If this bit is low even though the CSIS clock is alive, then it does not service any request from CSIS and keeps the request waiting. Once the main host disables CSIS, it should be reset by software or hardware before the main host enables CSIS again.	0

45.7.1.2 CSIS_DPHYCTRLn (n = 0 to 1, D-PHY Control Register)

- Base Address: 0x1188_0000, 0x1189_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved. NOTE: Do not change the value.	0
DPHYOn	[4:0]	RW	Enables D-PHY clock and data lane. [4]: Data lane 3 [3]: Data lane 2 [2]: Data lane 1 [1]: Data lane 0 [0]: Clock lane 0 = Disables D-PHY clock and data lane 1 = Enables D-PHY clock and data lane	0

NOTE: This register controls D-PHY.

45.7.1.3 CSIS_CONFIGn (n = 0 to 1, Configuration Register)

- Base Address: 0x1188_0000, 0x1189_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Hsync_LIntv	[31:26]	RW	Specifies interval between Hsync falling and Hsync rising (line interval). As illustrated in *Figure 45-2*, t2 specifies this interval. 6'h00 to 6'h3F cycle of Pixel clock.	0
Vsync_SIntv	[25:20]	RW	Specifies interval between Vsync rising and first Hsync rising. As illustrated in *Figure 45-2*, t1 specifies this interval. 6'h00 to 6'h3F cycle of Pixel clock	0
Vsync_EIntv	[19:8]	RW	Specifies interval between last Hsync falling and Vsync falling. As illustrated in *Figure 45-2*, t5 specifies this interval. 12'h000 to 12'hFFF cycle of Pixel clock	0
DataFormat[5:0]	[7:2]	RW	Specifies the image data format. 0x1E = YUV422 (8-bit) 0x2A = RAW8 0x2B = RAW10 0x2C = RAW12 0x30 = User defined 1 0x31 = User defined 2 0x32 = User defined 3 0x33 = User defined 4 Others = Reserved	0
NumOfDatLane	[1:0]	RW	Specifies the number of data lanes. 00 = 1 Data Lane 01 = 2 Data Lane 10 = 3 Data Lane 11 = 4 Data Lane	0

45.7.1.4 CSIS_DPHYSTSn (DPHY State Register)

- Base Address: 0x1188_0000, 0x1189_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_00F1

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved.	0
UlpsDat	[11:8]	R	Determines whether the data lane [3:0] is in ULPS. [11]: Data lane 3 [10]: Data lane 2 [9]: Data lane 1 [8]: Data lane 0 0 = Data lane is not ULPS 1 = Data lane is in ULPS	0
StopStateDat	[7:4]	R	Determines whether the data lane [3:0] is in Stop state. [7]: Data lane 3 [6]: Data lane 2 [5]: Data lane 1 [4]: Data lane 0 0 = Data lane is not in Stop state 1 = Data lane is in Stop state	F
RSVD	[3:2]	–	Reserved.	0
UlpsClk	[1]	R	Determines whether the clock lane is in ULPS. 0 = Clock lane is not ULPS 1 = Clock lane is in ULPS	0
StopStateClk	[0]	R	Determines whether the clock lane is in Stop state. 0 = Clock lane is not in Stop state 1 = Clock lane is in Stop state	1

45.7.1.5 CSIS_INTMSKn (n = 0 to 1, Interrupt Mask Register)

- Base Address: 0x1188_0000, 0x1189_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MSK_EvenBefore	[31]	RW	Receives non-image data before image at even frame. 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_EvenAfter	[30]	RW	Receives non-image data after image at even frame. 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_OddBefore	[29]	RW	Receives non-image data after image at even frame. 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_OddAfter	[28]	RW	Receives non-image data before image at odd frame. 0 = Disables Interrupt 1 = Enables Interrupt	0
RSVD	[27:13]	–	Reserved.	0
MSK_ERR_SOT_HS	[12]	RW	Specifies start of transmission error. 0 = Disables Interrupt 1 = Enables Interrupt	0
RSVD	[11:6]	–	Reserved.	0
MSK_ERR_LOST_FS	[5]	RW	Lost of Frame Start packet 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_ERR_LOST_FE	[4]	RW	Lost of Frame End packet 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_ERR_OVER	[3]	RW	Controls error. 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_ERR_ECC	[2]	RW	Specifies ECC error. 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_ERR_CRC	[1]	RW	Specifies CRC error. 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_ERR_ID	[0]	RW	Specifies unknown ID error. 0 = Disables Interrupt 1 = Enables Interrupt	0

NOTE: This register masks the interrupt sources.

45.7.1.6 CSIS_INTSRCn (n = 0 to 1, Interrupt Source Register)

- Base Address: 0x1188_0000, 0x1189_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
EvenBefore	[31]	RW	Receives non-image data before image at even frame. Write 1 = Clears status bit Write 0 = No effect	0
EvenAfter	[30]	RW	Receives non-image data after image at even frame. Write 1 = Clears status bit Write 0 = No effect	0
OddBefore	[29]	RW	Receives non-image data before image at odd frame. Write 1 = Clears status bit Write 0 = No effect	0
OddAfter	[28]	RW	Receives non-image data after image at odd frame. Write 1 = Clears status bit Write 0 = No effect	0
RSVD	[27:16]	–	Reserved.	0
ERR_SOT_HS	[15:12]	RW	Specifies start of transmission error.	0
RSVD	[11:6]	–	Reserved.	0
ERR_LOST_FS	[5]	RW	Indicates the lost of Frame Start packet	0
ERR_LOST_FE	[4]	RW	Indicates the lost of Frame End packet	0
ERR_OVER	[3]	RW	Specifies overflow caused in image FIFO. The outer bandwidth has to be faster than the input bandwidth. However, image FIFO can overflow due to user fault. There are two ways to prevent overflow: 1. Tune output pixel clock faster than current: You should set WCLK_Src in CSIS_CTRL register to 1 and then assign faster clock. 2. Tune input byte clock slower than current: Set register in camera module through I2C channel. When it generates the interrupt, turn the camera off. Assert software reset. If you do not assert software reset, MIPI CSIS will not receive any data. Tune the clock frequency and re-configure all the related registers. MIPI CSIS module is now ready for operation. Write 1 = Clears status bit Write 0 = No effect	0

Name	Bit	Type	Description	Reset Value
ERR_ECC	[2]	RW	Specifies ECC error. Write 1 = Clears status bit Write 0 = No effect	0
ERR_CRC	[1]	RW	Specifies CRC error. Write 1 = Clears status bit Write 0 = No effect	0
ERR_ID	[0]	RW	Specifies unknown ID error. Write 1 = Clears status bit Write 0 = No effect	0

NOTE: This register identifies interrupt sources.

45.7.1.7 CSIS_RESOLn (n = 0 to 1, Resolution Register)

- Base Address: 0x1188_0000, 0x1189_0000
- Address = Base Address + 0x002C, Reset Value = 0x8000_8000

Name	Bit	Type	Description	Reset Value
HResol	[31:16]	RW	Specifies horizontal image resolution. Input boundary of each image format is as follows: YUV422 (8-bit): 0x0001 to 0xFFFF RAW8: 0x0001 to 0xFFFF RAW10: 4n (where n is 1, 2, 3, ...) RAW12: 2n (where n is 1, 2, 3, ...) RAW14: 4n (n is 1, 2, 3,....)	0x8000
VResol	[15:0]	RW	Specifies vertical image resolution. Input boundary: 0x0001 to 0xFFFF	0x8000

45.7.1.8 SDW_CONFIGn (n = 0 to 1, Shadow Configuration Register)

- Base Address: 0x1188_0000, 0x1189_0000
- Address = Base Address + 0x0038, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Hsync_LIntv	[31:26]	R	Specifies current interval between Hsync falling and Hsync rising (Line interval).	0
Vsync_SIntv	[25:20]	R	Specifies current interval between Vsync rising and first Hsync rising.	0
Vsync_EIntv	[ ]	R	Specifies current interval between last Hsync falling and Vsync falling.	0
DataFormat	[ ]	R	Specifies current image data format.	0
NumOfDatLane	[ ]	R	Specifies current number of data lanes. These bits are always the same as the number of data lanes in CSIS_CONFIG register because these bits are static signals that do not change in operation.	0

45.7.1.9 SDW_RESOLn (n = 0 to 1, Shadow Resolution Register)

- Base Address: 0x1188_0000, 0x1189_0000
- Address = Base Address + 0x003C, Reset Value = 0x8000_8000

Name	Bit	Type	Description	Reset Value
HResol	[31:16]	R	Specifies current horizontal image resolution.	0x8000
VResol	[ ]	R	Specifies current vertical image resolution.	0x8000

45.7.1.10 CSIS_PKTDATAn (n = 0 to 1, Packet Data Register)

- Base Address: 0x1188_0000, 0x1189_0000
- Address = Base Address + 0x2000 to 0x3FFC, Reset Value = 0xXXXX_XXXX

Name	Bit	Type	Description	Reset Value
PktData	[31:0]	R	Specifies packet data.	Undefined

46 2D Graphic Accelerator

46.1 Overview

FIMG-2D is a 2D graphic accelerator (G2D) that supports Bit Block Transfer (BitBLT).

G2D performs these tasks:

1. Configures the rendering parameters, such as foreground color and coordinates data by setting the drawing-context registers.
2. Starts the rendering process by setting the relevant command registers accordingly.

46.2 Features

Features of G2D are:

- HOST Interface
- Primitives
- Per-pixel Operation
- Data For

46.2.1 Host Interface

- APB Slave
- AXI Master (DMA Mode):
  - Linked list of command lists

46.2.2 Primitives

- BitBLT
  - Stretched BitBLT supports that uses scale factor:
    - Nearest sampling
    - Smooth scaling:
      - Scale up/down: Bilinear sampling
  - Memory to memory
  - Reverse addressing (X Positive/Negative, Y Positive/Negative)
  - Various repeat type support: Repeat, Reflect, Pad, Clamp, and None

46.2.3 Per-pixel Operation

- Maximum 8000 × 8000 image size
- Window clipping
- 90°/180°/270° rotation
- X-flip/Y-flip
- Totally 4-operand Raster Operation (ROP4)
  - Mask, Pattern, Source, Destination
- Masking operation
  - 1-bit/4-bit/8-bit/16-bit/32-bit masking support
- Alpha blending
  - Alpha blending with a user-specified constant alpha
  - Per-pixel alpha blending
  - Alpha blending with both a constant alpha and per-pixel alpha
  - GL style blend function support
  - Porter/Duff rule support that uses GL style blending
  - Component blend with mask
- Color key
  - RGBA Color Key
- Dithering

46.2.4 Data Format

- 8/16/24/32 bpp, packed 24 bpp color format support
- Premultiplied alpha, Non-premultiplied alpha format support
- 1 bpp/4 bpp/8 bpp/16 bpp/32 bpp Mask format support

46.3 Host Interface: DMA Mode

FIMG-2D version 4.1 supports DMA Mode as host interface. Users can make command lists to reduce HOST (ARM) loads. FIMG-2D version 4.1 load own SFR data and the required data from set memory base to buffer. The interpreter of FIMG then reads data, sets SFR, and it can be revised "start core engine".

Figure 46-1 illustrates the data structure of command list.

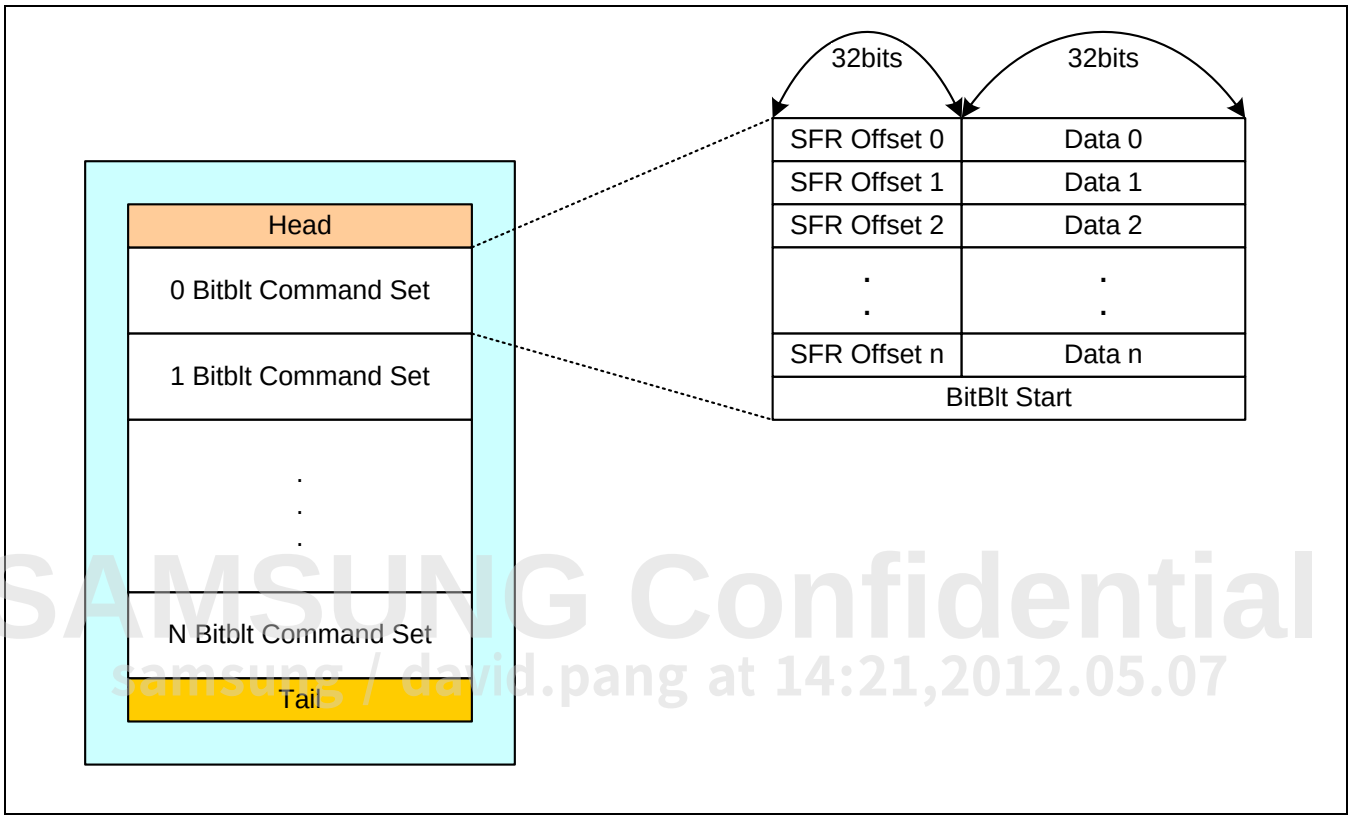


Figure 46-1 Data Structure of Command List

Data structure consists of:

- Header - This includes the number of commands: SFR Offset and SFR data.
- Tail - This links to the point base address of the other command lists.
- Command sets - This includes SFR Offsets and SFR data. A commands set includes BitBLT start (BITBLT_START_REG).

Figure 46-2 illustrates the detail description of one command list.

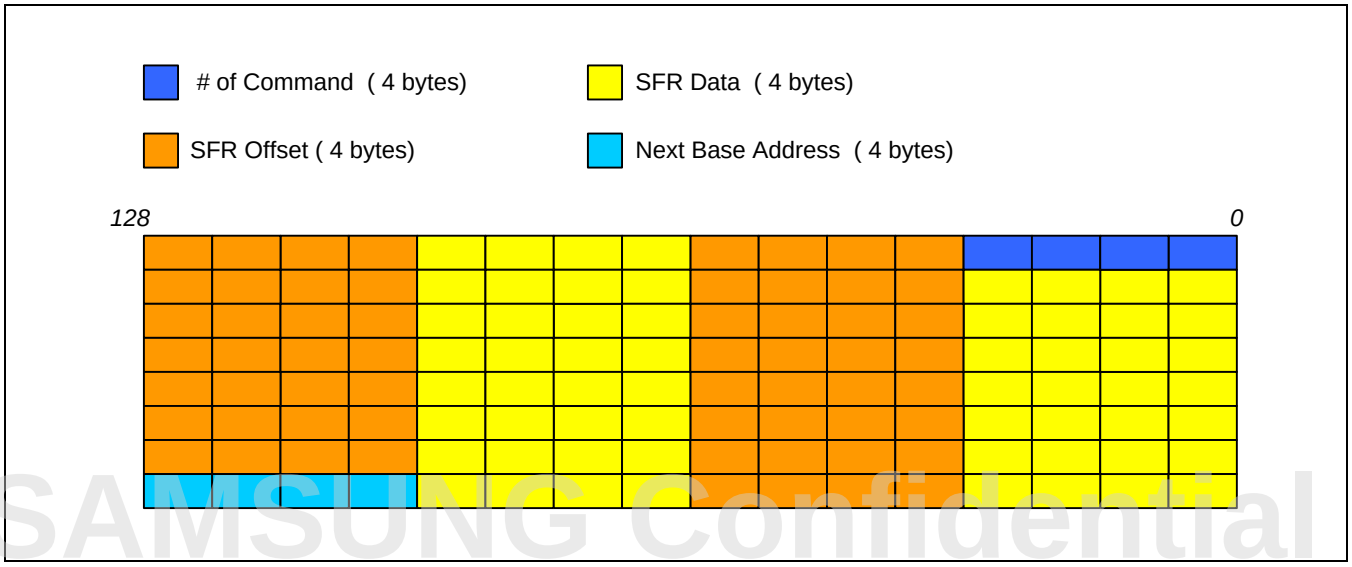


Figure 46-2 Detail Description of One Command List

DMA interpreter of FIMG-2D version 4.1 processes a variety of command list that uses the tail of data structure. As the tail points to the base address of other command list, interpreter successively read command lists. When the tail Reads the command list, the last commands list will have NULL tail.

Figure 46-3 illustrates linked list of multiple command lists.

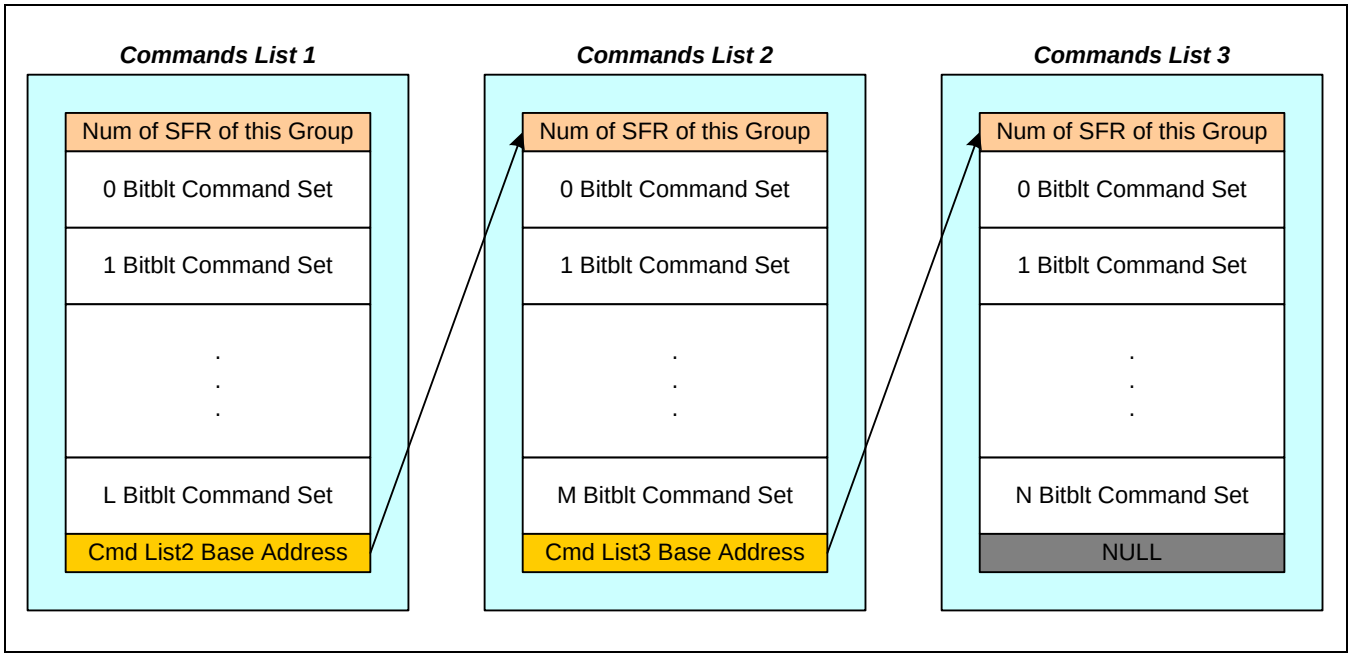


Figure 46-3 Linked List of Multiple Command Lists

Our hardware supports HOLD state for debugging or waiting at the BitBLT that user wants. You can utilize BitBLT_Hold and List_Hold of DMA_HOLD_CMD register for debugging. To hold at BitBLT that user wants, it should enable StartNHold of BITBLT_START on the condition that it enables User_Hold of DMA_HOLD_CMD register.

Figure 46-4 illustrates the HOLD scenario examples.

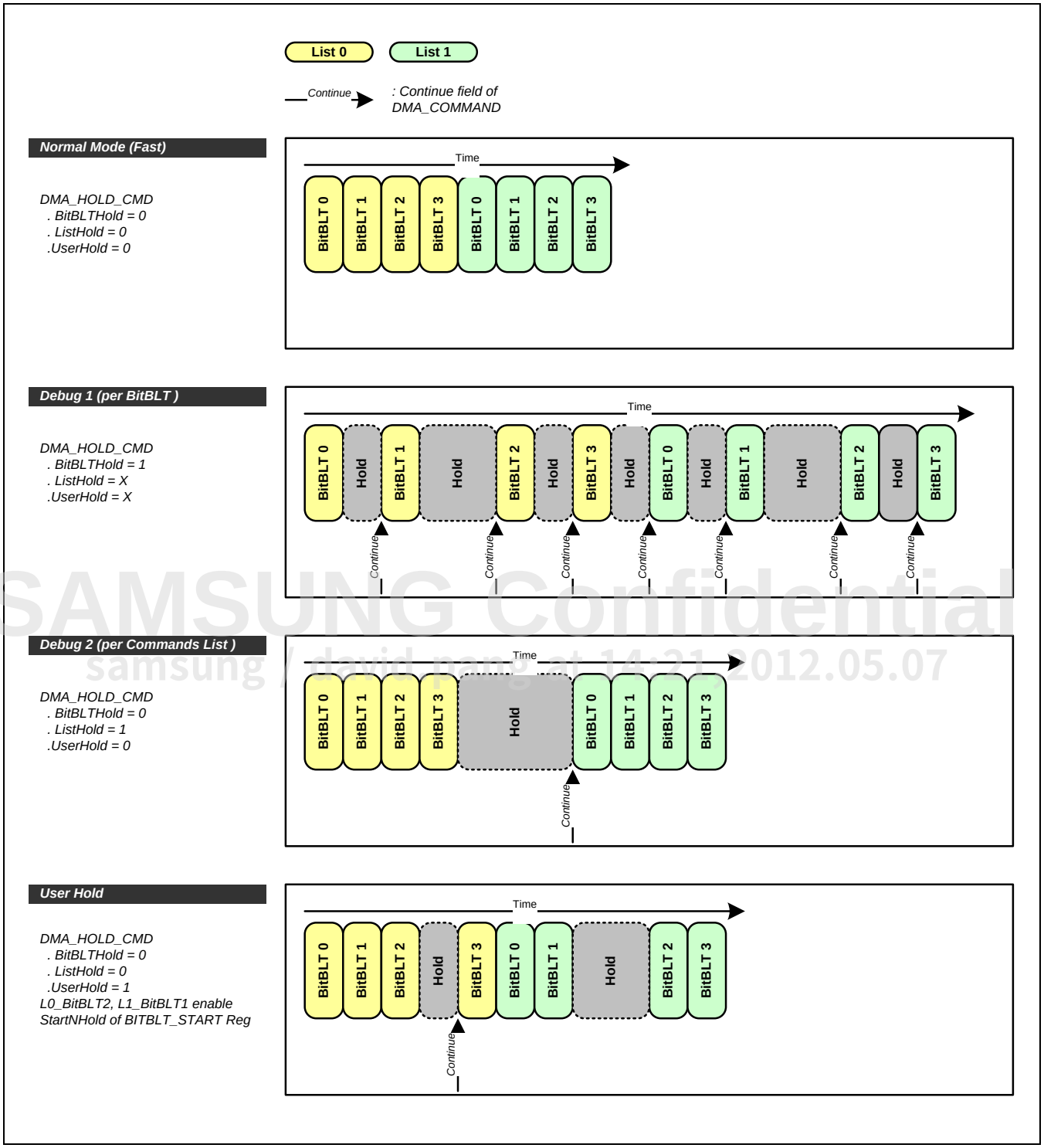


Figure 46-4 HOLD Scenario Examples

Related Registers

DMA_SFR_BASE_ADDR	First Commands list Base Address
DMA_COMMAND	Controls signal for DMA Mode start and halt.
DMA_EXE_LIST_NUM	DMA completes core operation when all BitBLT of this set command list is complete.
DMA_STATUS	This SFR shows Status of DMA operation (the number of commands list and BitBLT is complete)
DMA_HOLD_CMD	This SFR controls DMA Hold and Continue per BitBLT or Commands list

46.4 Color Format Conversion

This section includes:

- RGBA Format

46.4.1 RGBA Format

FIMG-2D version 4.1 supports:

- Eight RGBA formats
- Alpha format
- Luminance formats:
  - XRGB_8888
  - ARGB_8888
  - RGB_565
  - XRGB_1555
  - ARGB_1555
  - XRGB_4444
  - ARGB_4444
  - PACKED_RGB_888 and A_8
  - L_8

FIMG-2D version 4.1 supports four channel orders, namely:

- ARGB
- RGBA
- ABGR
- BGRA

- *Figure 46-5* illustrates the structure of each color format and RGBA format.

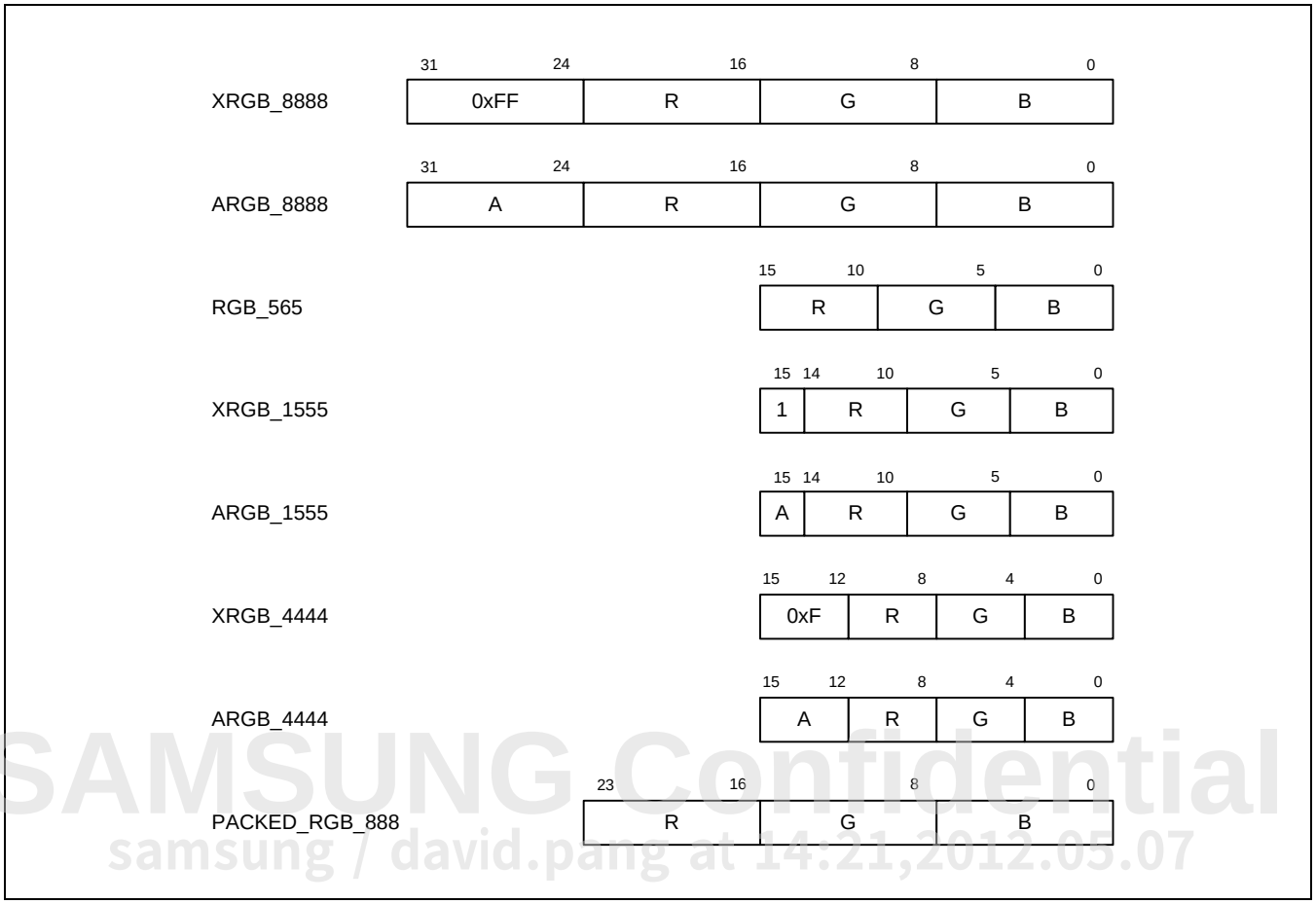


Figure 46-5 RGBA Format

When it converts 16-bit color data to 32-bit, it makes expanded color data to follow this rule:

It can be revised → Each field of data should shift for (8-x) bits to the left. Here, x is the bit-width of the field. It pads the least significant x bits of the new field data with the most significant x bits of the original field data. For example, when the R value in RGB_565 format is 5'b11010, it is converted to 8'b11010110 with three Least Significant Bits (LSBs) padded with three Most Significant Bits (MSBs) (3'b110) from the original R value.

Note that, the A field in RGBA_5551 and ARGB_1555 only has one bit. Therefore, it is converted to either 8'b00000000 or 8'b11111111 (A = 1'b1).

When it converts a 32-bit color data to 16-bit, it truncates the data of each data field to x bits. Here, x is the bit-width of the field in the new color format. For example, when the R value in ARGB_8888 format is 8'b11001110, it is converted to 5'b11001 in the RGB_565 format with the three discarded LSBs.

Note that, when the A field of the 32-bit color data is not 0, the A field in ARGB_1555 is 1'b1; A field in ARGB_1555 is 1'b0.

46.5 Rendering Pipeline

The remainder of this chapter introduces to the functionality and related registers of each stage.

Figure 46-6 illustrates the rendering pipeline of FIMG-2D version 4.1.

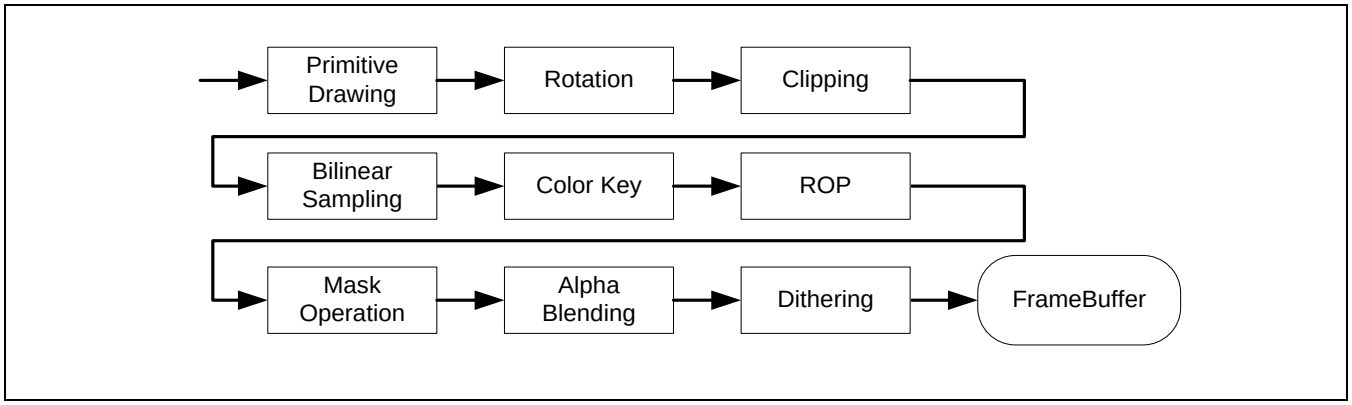


Figure 46-6 FIMG-2D Rendering Pipeline

46.5.1 Primitive Drawing

Primitive drawing determines the pixels to fill the framebuffer, and pass their coordinates to the next stage for further operations.

FIMG-2D version 4.1 supports BitBLT.

46.5.1.1 BitBLT

- A Bit Block Transfer is a transformation of a rectangular block of pixels. Typical applications of BitBLT include:
- Copying the off-screen pixel data to frame buffer
- Selecting one of two raster operations by mask value
- Combining the selected raster operation to bitmap patterns
- Changing the dimension of a rectangular image.

You can implement stretch BitBLT by using Digital Differential Analyzer (DDA) algorithm and the nearest sampling. For smooth scaling of Stretch BitBLT, use Bilinear sampling.

On-Screen Rendering

The on-screen BitBLT copies a rectangular block of pixels on screen to another position on the same screen. Note that the restrictions for on-screen rendering are:

1. SRC_BASE_ADDR_REG = DST_BASE_ADDR_REG
2. SRC_STRIDE_REG = DST_STRIDE_REG
3. SRC_COLOR_MODE_REG = DST_COLOR_MODE_REG

Off-Screen Rendering

The off-screen BitBLT copies pixel data from off-screen memory to frame buffer. It performs color format conversion automatically when SRC_COLOR_MODE_REG differs from DST_COLOR_MODE_REG.

Transparent Mode

FIMG-2D V 4.1 renders image in Transparent Mode. In this mode, it discards the pixels of same color with blue screen color (BS_COLOR_REG). This results in a transparent effect.

Figure 46-7 illustrates the function of Transparent Mode, in which the BS_COLOR_REG is set to white.

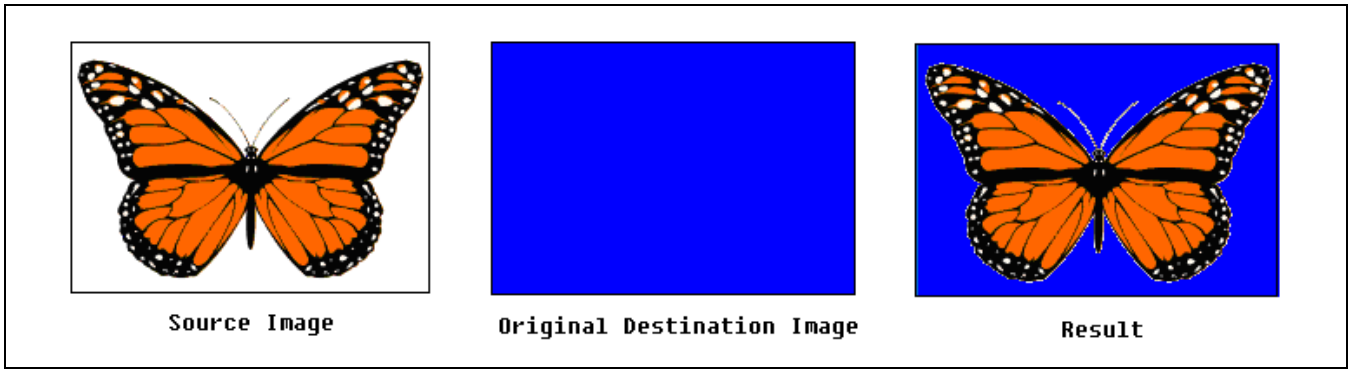


Figure 46-7 Function of Transparent Mode

FIMG-2D version 4.1 also supports Blue Screen Mode. In this mode, it replaces the pixels of same color with blue screen color (BS_COLOR_REG) by the background color (BG_COLOR_REG).

FIMG-2D version 4.1 supports memory-to-memory mode of blt.

46.5.1.2 Related Registers

Register	Description
SRC_LEFT_TOP_REG	Coordinate the top-left of the source image.
SRC_RIGHT_BOTTOM_REG	Coordinate the bottom-right of the source image.
SRC_SELECT_REG	Selects one of the these cases: 2'b00: Normal mode (source image in the memory) 2'b01: Foreground color 2'b10: Background color.
DST_LEFT_TOP_REG	Coordinates the leftmost topmost coordinate of the destination image.
DST_RIGHT_BOTTOM_REG	Coordinates the rightmost bottommost coordinate of the destination image.
DST_SELECT_REG	Selects one of the these cases: 2'b00: Normal mode (destination image in the memory) 2'b01: Foreground color 2'b10: Background color.
SRC_BASE_ADDR_REG	The base address of the source image (when it uses normal mode in SRC_SELECT_REG).
DST_BASE_ADDR_REG	The base address of the destination image (usually the frame buffer base address).
DST_STRIDE_REG	The stride of the destination image.
SRC_COLOR_MODE_REG	The color format, the channel order and the number of plane of the source image.
DST_COLOR_MODE_REG	The color format, the channel order, and the number of plane of the destination image.
FG_COLOR_REG	Foreground color value.
BG_COLOR_REG	Background color value.
BS_COLOR_REG	Blue screen color value.
BITBLT_COMMAND_REG	Enable/disable Transparent Mode or Blue Screen Mode. Enable/disable Alpha Blending, Dithering, Mask, and ROP4.

46.5.2 Rotation and Addressing Direction (Flip)

You can rotate the pixels can be rotated by 90 degree counter clockwise or flipped around X-axis or Y-axis. You can perform the flip operation by direction of source read and destination read.

Table 46-1 describes the effects of rotation and flip options.

Related Registers

Register	Description
ROTATE_REG	Enables 90 degree rotation of source image, mask, and pattern image
SRC_MSK_DIRECT_REG	Addressing direction of source/mask memory to read
DST_PAT_DIRECT_REG	Addressing direction of destination/pattern memory to read and write

Table 46-1 Rotation Effect

	Effect
0 °	Rotated X = Original X Rotated Y = Original Y
90 °	Rotated X = Original Y Rotated Y = Original Width – 1 – Original X

Table 46-2 describes the effects of addressing direction.

Table 46-2 Addressing Direction Effect

	Effect
Src X Direction = Dst X Direction	No flip over X-axis
Src X Direction ≠ Dst X Direction	Horizontal flip
Src Y Direction = Dst Y Direction	No flip over Y-axis
Src Y Direction ≠ Dst Y Direction	Vertical flip

Figure 46-8 illustrates an example of rotation and flip.

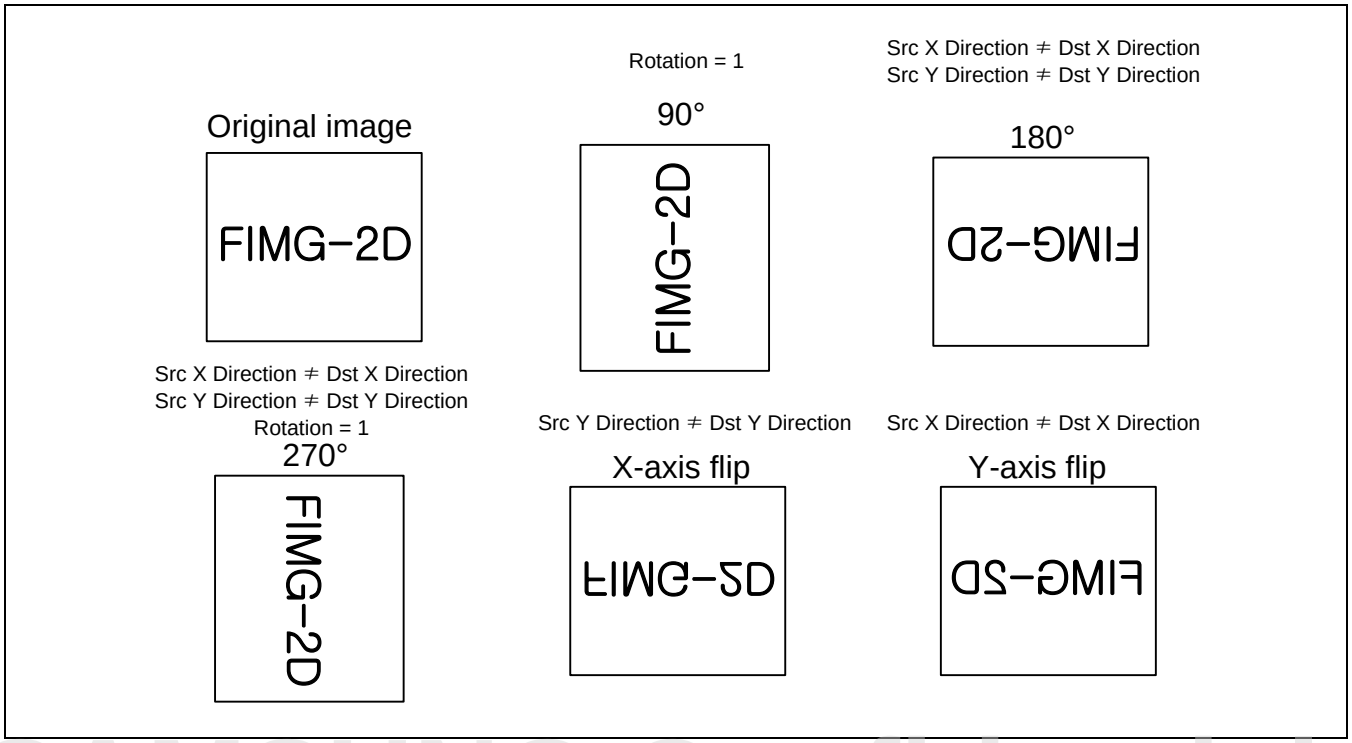


Figure 46-8 Rotation and Flip Example

46.5.3 Clipping

Clipping discards the pixels after rotation outside the clipping window. The discarded pixels do not go through the rest of rendering pipelines.

NOTE: That the clipping windows must reside totally inside the screen. If you set the clipping window the same size as the screen, it will disable the clipping effect. It does not allow a clipping window bigger than the screen size.

Related Registers

Register	Description
BITBLT_COMMAND_REG	Enables/disables clipping window (CWEn Field)
CW_LT_REG	Coordinate the top-left of the clipping window
CW_RB_REG	Coordinate the bottom-right of the clipping window

46.5.4 Color Key

Color Key conditionally discards a pixel based on the outcome. This outcome is of a comparison between the color value of the pixel of the source/destination image and the DR (Minimum)/DR (Maximum) values.

If each field (R, G, B, A) or (Y, Cb, Cr) of the color value are set to range of [DR (Min.), DR (Max.)], it passes this pixel to the next state otherwise, it discards the pixel. User can disable the Stencil Test configuration on a specific field by clearing the corresponding bits in SRC_COLORKEY_CTRL_REG and DST_COLORKEY_CTRL_REG.

Related Registers

Register	Description
BITBLT_COMMAND_REG	Enables/disables Colorkey (ColorKeyMode Field)
SRC_COLORKEY_CTRL_REG	RGBA source Stencil Test configurations, such as enables/disables the test and so on.
SRC_COLORKEY_DR_MIN_REG	Sets the minimum source DR value for each field (R, G, B, A)
SRC_COLORKEY_DR_MAX_REG	Sets the maximum source DR value for each field (R, G, B, A)
DST_COLORKEY_CTRL_REG	RGBA Destination Stencil Test configurations, such as enables/disables the test and so on.
DST_COLORKEY_DR_MIN_REG	Sets the DR (Min.) value for each field (R, G, B, A)
DST_COLORKEY_DR_MAX_REG	Sets the DR (Max.) value for each field (R, G, B, A)

46.5.5 Raster Operation

Raster operation performs Boolean operations on these four operands:

- Mask
- Third operand
- Source
- Destination

These operands are according to two 8-bit-ROP3 values specified by the user. User can select unmasked ROP3 values and masked ROP3 values with binary mask image.

Table 46-3 lists the truth table of ROP3.

Table 46-3 Truth table of ROP3

Third Operand	Source	Destination	ROP Value
0	0	0	Bit[0]
0	0	1	Bit[1]
0	1	0	Bit[2]
0	1	1	Bit[3]
1	0	0	Bit[4]
1	0	1	Bit[5]
1	1	0	Bit[6]
1	1	1	Bit[7]

The third operand can be pattern, foreground color, or background color; which are configurable by THIRD_OPERAND_ REG.

The pattern supports RGBA formats of source image or destination image except A_8 and L_8 formats. Use this equation to calculate the pattern pixel coordinate (PatX, PatY):

PatX = (PatternOffsetX + DstX) % PatternWidth
PatY = (PatternOffsetY + DstY) % PatternHeight

Where PatternOffsetX and PatternOffsetY are the offset values. It specifies these values in register PAT_OFFSET_REG. PatternWidth and PatternHeight are sizes of the pattern. It specifies these values in register PAT_SIZE_REG.

Examples on how to use the ROP3 value to perform the operations:

- Final Data = Source. Only the Source data matter, so ROP Value = "0xCC".
- Final Data = Destination. Only the destination data matter, so ROP Value = "0xAA".
- Final Data = Pattern. Only the Pattern data matter, so ROP Value = "0xF0".
- Final Data = Source AND Destination. ROP Value = "0xCC" & "0xAA" = "0x88"
- Final Data = Source or Pattern. ROP Value = "0xCC" | "0xF0" = "0xFC".

Related Registers

Register	Description
PAT_BASE_ADDR_REG	Base address of the pattern image
PAT_SIZE_REG	Size of the pattern
PAT_COLOR_MODE_REG	Color channel order and color format of the pattern
PAT_OFFSET_REG	Coordinates offset of the pattern
MASK_BASE_ADDR_REG	Base address of the mask image
THIRD_OPERAND_REG	Third operand selection for unmasked ROP3 and masked ROP3
ROP4_REG	ROP4 value

46.5.6 Mask Operation

When MaskROP4En register is 2'b1x, FIMG-2D supports three types of Mask Operation as:

1. Uses only alpha channel as mask (MskOpType == 2'b00)

SrcA_New = SrcA × MskA;	**SrcAlpha_A = SrcA × MskA;**
SrcR_New = SrcR × MskA;	**SrcAlpha_R = SrcAlpha_A;**
SrcG_New = SrcG × MskA;	**SrcAlpha_G = SrcAlpha_A ;**
SrcB_New = SrcB × MskA;	**SrcAlpha_B = SrcAlpha_A;**

2. Blends components with mask (MskOpType == 2'b01)

SrcA_New = SrcA × MskA;	**SrcAlpha_A = SrcA × MskA;**
SrcR_New = SrcR × MskR;	**SrcAlpha_R = SrcA × MskR;**
SrcG_New = SrcG × MskG;	**SrcAlpha_G = SrcA × MskG;**
SrcB_New = SrcB × MskB;	**SrcAlpha_B = SrcA × MskB;**

3. Channel mask (MskOpType == 2'b10)

SrcA_New = SrcA × MskA;	**SrcAlpha_A = SrcA × MskA;**
SrcR_New = SrcR × MskR;	**SrcAlpha_R = SrcAlpha_A;**
SrcG_New = SrcG × MskG;	**SrcAlpha_G = SrcAlpha_A;**
SrcB_New = SrcB × MskB;	**SrcAlpha_B = SrcAlpha_A;**

4. Blends Ex. SRC-OVER component with mask.

DstA = SrcA_New + (1 – SrcAlpha_A) × DstA;
DstR = SrcR_New + (1 – SrcAlpha_R) × DstR;
DstG = SrcG_New + (1 – SrcAlpha_G) × DstG;
DstB = SrcB_New + (1 – SrcAlpha_B) × DstB;

46.5.7 Alpha Blending

Alpha Blending combines the source color and the destination color in the frame buffer to get the new destination color.

The equation for Alpha Blending equation (blend function) is:

[source coefficients × source color + destination coefficients × destination color]

This equation supports Porter/Duff rule and some extra blend rule. Alpha blending supports Global Alpha (Constant alpha) and Global color (Constant color) by selecting coefficients to GB_ALPHA, GB_COLOR, SRCPGB_ALPHA, SRCMGB_ALPHA, DSTPGB_ALPHA, or DSTMGB_ALPHA.

Alpha blending supports Darken or Lighten rules when Darken or Lighten fields are set. The pixel fill rate is almost half because it requires two blending pipelines per pixel.

Here are the definitions of Porter/Duff rules:

```
Source color: Sc[Sa, Sr, Sg, Sb]
Destination color: Dc[Da, Dr, Dg, Db]
Result color: Rc[Ra, Rr, Rg, Rb]

[The equations about Porter/Duff Rule & Extra blend Rule]

0. NONE        [Rc]           = [Sc × Sa + (1 − Sa) × Dc]
1. CLEAR       [Rc]           = [0];
2. SRC         [Rc]           = [Sc];
3. DST         [Rc]           = [Dc];
4. SRC-OVER    [Rc]           = [Sc + (1 − Sa) × Dc];
5. DST-OVER    [Rc]           = [Dc + (1 − Da) × Sc];
6. SRC-IN      [Rc]           = [Sc × Da];
7. DST-IN      [Rc]           = [Sa × Dc];
8. SRC-OUT     [Rc]           = [Sc × (1 − Da)];
9. DST-OUT     [Rc]           = [Dc × (1 − Sa)];
10. SRC-ATOP   [Rc]           = [Da, Sc × Da + (1 − Sa) × Dc];
11. DST-ATOP   [Rc]           = [Sa, Sa × Dc+Sc × (1 − Da)];
12. XOR        [Rc]           = [Sc × (1 − Da) + (1 − Sa) × Dc];
13. PLUS       [Rc]           = [Sc + Dc];
14. MULTIPLY   [Rc]           = [Sc × Dc];
15. SCREEN     [Rc]           = [Sc + (1 − Sc) × Dc];
16. DARKEN     [Ra, Rc]              = [Sa + (1 − Sa) ×Da, if (Sc × Da < Dc × Sa) {Sc + (1 − Sa) × Dc},
                              else {Dc + (1 − Da) × Sc}];
17. LIGHTEN    [Ra, Rc]              = [Sa + (1 − Sa) × Da, if (Sc × Da > Dc × Sa) {Sc + (1 − Sa) ×
Dc},
                              else {Dc + (1 − Da) × Sc}];
18. DISJOINT_SRC_OVER
                [Rc]          = [Sc + Min. (1, ((1 − Sa)/Da) × Dc)]; (if Da == 0, [Rc] = [Sc + Dc])
19. DISJOINT_DST_OVER (SATURATE)
                [Rc]          = [Dc + Min. (1, ((1 − Da)/Sa) × Sc)]; (if Sa == 0, [Rc] = [Sc + Dc])
20. DISJOINT_SRC_IN
```

```
              [Rc]          = [Max. (0, 1 - ((1 - Da)/Sa) × Sc)]; (if Sa == 0, [Rc] = [0])
21. DISJOINT_DST_IN
              [Rc]          = [Max. (0, 1 - ((1 - Sa)/Da) × Dc)]; (if Da == 0, [Rc] = [0])
22. DISJOINT_SRC_OUT
              [Rc]          =[Min. (1, ((1 - Da)/Sa) × Sc)]; (if Sa ==0, [Rc] = [Sc])
23. DISJOINT_DST_OUT
              [Rc]          = [Min. (1, ((1 - Sa)/Da) × Dc)]; (if Da == 0, [Rc] = [Dc])
24. DISJOINT_SRC_ATOP
              [Rc]          = [Max. (0, 1-((1 - Da)/Sa) × Sc) + Min. (1, ((1 - Sa)/Da) × Dc)];
                            (if Sa == 0 && Da == 0, [Rc] = [Dc] else Sa == 0,
              [Rc]          = [Min. (1, ((1 - Sa)/Da) × Dc)] else Da == 0,
              [Rc]          = [Max. (0, 1-((1 - Da)/Sa) × Sc) + Dc])
25. DISJOINT_DST_ATOP
              [Rc]          = [Min. (1, ((1 - Da)/Sa) × Sc) + Max. (0, 1-((1 - Sa)/Da) × Dc)];
                            (if Sa == 0 && Da == 0, [Rc] = [Sc]else Sa == 0,
              [Rc]          = [Sc + Max. (0, 1 - ((1 - Sa)/Da) × Dc)]else Da == 0,
              [Rc]          = [Min. (1, ((1-Da)/Sa) × Sc)])
26. DISJOINT_XOR
              [Rc]          = [Min. (1, ((1-Da)/Sa) × Sc) + Min. (1, ((1-Sa)/Da) × Dc)];
                            (if Sa == 0 && Da == 0, [Rc] = [Sc + Dc] else Sa == 0,
              [Rc]          = [Sc + Min. (1, ((1 - Sa)/Da) × Dc)] else Da == 0,
              [Rc]          = [Min. (1, ((1 - Da)/Sa) × Sc) + Dc])
27. CONJOINT_SRC_OVER
              [Rc]          = [Sc + Max.  (0, 1 - ((Sa/Da) × Dc))]; (if Da == 0, [Rc] = [Sc])
28. CONJOINT_DST_OVER
              [Rc]          = [Dc + Max. (0, 1 - ((Da/Sa) × Sc)]; (if Sa == 0, [Rc] = [Dc])
29. CONJOINT_SRC_IN
              [Rc]          = [Min. (1, (Da/Sa) × Sc)]; (if Sa == 0, [Rc] = [Sc])
30. CONJOINT_DST_IN
              [Rc]          = [Min. (1, (Sa/Da) × Dc)]; (if Da ==0, [Rc] = [Dc])
31. CONJOINT_SRC_OUT
              [Rc]          = [Max. (0, 1 - ((Da/Sa) × Sc)]; (if Sa ==0, [Rc] = [0])
32. CONJOINT_DST_OUT
              [Rc]          = [Max. (0, 1 - ((Sa/Da) × Dc)]; (if Da ==0, [Rc] = [0])
33. CONJOINT_SRC_ATOP
              [Rc]          = [Min. (1, (Da/Sa) × Sc) + Max. (0, 1 - (Sa/Da) × Dc)];
                            (if Sa == 0 && Da == 0,
              [Rc]          = [Sc] else Sa == 0, [Rc]
                            = [Sc + Max. (0, 1 - (Sa/Da) × Dc)] else Da == 0,
              [Rc]          = [Min. (1, (Da/Sa) × Sc)])
34. CONJOINT_DST_ATOP
              [Rc]          = [Max. (0, 1 - (Da/Sa) × Sc) + Min. (1, (Sa/Da) × Dc)];
                            (if Sa == 0 && Da == 0,
              [Rc]          = [Dc] else Sa == 0, [Rc] = [Min. (1, (Sa/Da) × Dc)] else Da == 0,
              [Rc]          = [Max. (0, 1 - (Da/Sa) × Sc) + Dc]
35. CONJOINT_XOR
              [Rc]          = [Max. (0, 1 - (Da/Sa) × Sc) + Max. (0, 1 - (Sa/Da) × Dc)];
                            (if Sa == 0 && Da == 0,
              [Rc]          = [0] else Sa == 0, [Rc] = [Max. (0, 1 - (Sa/Da) × Dc)] else Da == 0,
              [Rc]          = [Max. (0, 1 - (Da/Sa) × Sc)]
```

This example shows how to use the blend function (select coefficient) for Porter/Duff rule:

```
[Example] [Sourc Color Coefficient, Destination Color Coefficient]
0. None                      [SRC_ALPHA, INV_SRC_ALPHA]
1. CLEAR                     [ZERO, ZERO]
2. SRC                       [ONE, ZERO]
3. DST                       [ZERO, ONE]
4. SRC-OVER                  [ONE, INV_SRC_ALPHA]
5. DST-OVER                  [INV_DST_ALPHA, ONE]
6. SRC-IN                    [DST_ALPHA, ZERO]
7. DST-IN                    [ZERO, SRC_ALPHA]
8. SRC-OUT                   [INV_DST_ALPHA, ZERO]
9. DST-OUT                   [ZERO, INV_SRC_ALPHA]
10. SRC-ATOP                 [DST_ALPHA, INV_SRC_ALPHA]
11. DST-ATOP                 [INV_DST_ALPHA, SRC_ALPHA]
12. XOR                      [INV_DST_ALPHA, INV_SRC_ALPHA]
13. PLUS                     [ONE, ONE]
14. MULTIPLY                 [DST_COLOR, ZERO]
15. SCREEN                   [INV_DST_COLOR, ONE]
16. DARKEN                   Set Darken field on BLEND_FUNCTION_REG
17. LIGHTEN                  Set Lighten field on BLEND_FUNCTION_REG
18. DISJOINT_SRC_OVER        [ONE, DISJOINT_S]
19. DISJOINT_DST_OVER        [DISJOINT_D, ONE]
20. DISJOINT_SRC_IN          [INV_DISJOINT_D, ZERO]
21. DISJOINT_DST_IN          [ZERO, INV_DISJOINT_S]
22. DISJOINT_SRC_OUT         [DISJOINT_D, ZERO]
23. DISJOINT_DST_OUT         [ZERO, DISJOINT_S]
24. DISJOINT_SRC_ATOP        [INV_DISJOINT_D, DISJOINT_S]
25. DISJOINT_DST_ATOP        [DISJOINT_D, INV_DISJOINT_S]
26. DISJOINT_XOR             [DISJOINT_D, DISJOINT_S]
27. CONJOINT_SRC_OVER        [ONE, INV_CONJOINT_S]
28. CONJOINT_DST_OVER        [INV_CONJOINT_D, ONE]
29. CONJOINT_SRC_IN          [CONJOINT_D, ZERO]
30. CONJOINT_DST_IN          [ZERO, CONJOINT_S]
31. CONJOINT_SRC_OUT         [INV_CONJOINT_D, ZERO]
32. CONJOINT_DST_OUT         [ZERO, INV_CONJOINT_S]
33. CONJOINT_SRC_ATOP        [CONJOINT_D, INV_CONJOINT_S]
34. CONJOINT_DST_ATOP        [INV_CONJOINT_D, CONJOINT_D]
35. CONJOINT_XOR             [INV_CONJOINT_D, INV_CONJINT_S]
```

Related Registers

Register	Description
BITBLT_COMMAND_REG	Blending configurations Disables/enables alpha blending
ALPHA_REG	Global alpha value and global color value.
BLEND_FUNCTION_REG	Source/Destination coefficient
ROUND_MODE_REG	Round mode Premultiply, Blending, Depremultiply

46.5.8 Fast Solid Color Fill

FIMG-2D version 4.1 supports Fast Solid Color Fill operation to accelerate solid color fill.

Register	Description
BITBLT_COMMAND_REG	Disables/enables Solid Color fill
DST_BASE_ADDR_REG	Base address of the destination image
DST_STRIDE_REG	Destination stride (2's complement value)
DST_COLOR_MODE_REG	Destination color format, channel order
DST_LEFT_TOP_REG	Destination left top coordinate register
DST_RIGHT_BOTTOM_REG	Destination right bottom coordinate register
SF_COLOR_REG	Solid fill color register

46.6 Register Description

46.6.1 Register Map Summary

- Base Address: 0x1080_0000

Register	Offset	Description	Reset Value
General Registers			
SOFT_RESET_REG	0x0000	Software reset register	0x0000_0000
INTEN_REG	0x0004	Interrupt enable register	0x0000_0000
INTC_PEND_REG	0x000C	Interrupt control pending register	0x0000_0000
FIFO_STAT_REG	0x0010	Command FIFO status register	0x0000_0001
AXI_MODE_REG	0x001C	AXI Mode register	0x0200_0000
DMA_SFR_BASE_ADDR_REG	0x0080	DMA base address register	0x0000_0000
DMA_COMMAND_REG	0x0084	DMA command register	0x0000_0000
DMA_EXE_LIST_NUM_REG	0x0088	DMA list done count register	0x0000_0000
DMA_STATUS_REG	0x008C	DMA status register	0x0000_0001
DMA_HOLD_CMD_REG	0x0090	DMA hold register	0x0000_0000
Command Registers			
BITBLT_START_REG	0x0100	BitBLT start register	Undefined
BITBLT_COMMAND_REG	0x0104	Command register for BitBLT	0x0000_0000
BLEND_FUNCTION_REG	0x0108	Blend function for alpha blending	0x0040_0000
ROUND_MODE_REG	0x010C	Round mode selection	0x0000_0233
Parameter Setting Registers			
Rotate & Direction			
ROTATE_REG	0x0200	Rotation register	0x0000_0000
SRC_MSK_DIRECT_REG	0x0204	Source and mask direction register	0x0000_0000
DST_PAT_DIRECT_REG	0x0208	Destination and pattern direction register	0x0000_0000
Source			
SRC_SELECT_REG	0x0300	Source image selection register	0x0000_0001
SRC_BASE_ADDR_REG	0x0304	Source image base address register	0x0000_0000
SRC_STRIDE_REG	0x0308	Source stride register	0x0000_0000
SRC_COLOR_MODE_REG	0x030C	Source image color mode register	0x0000_0000
SRC_LEFT_TOP_REG	0x0310	Source left top coordinate register	0x0000_0000
SRC_RIGHT_BOTTOM_REG	0x0314	Source right bottom coordinate register	0x0000_0000
SRC_REPEAT_MODE_REG	0x031C	Source repeat mode register	0x0000_0000
SRC_PAD_VALUE_REG	0x0320	Source pad value register	0x0000_0000
SRC_A8_RGB_EXT_REG	0x0324	Source A8 RGB Extension register	0x0000_0000
SRC_SCALE_CTRL_REG	0x0328	Source scale control register	0x0000_0000
SRC_XSCALE_REG	0x032C	Source X scale factor register	0x0001_0000

Register	Offset	Description	Reset Value
SRC_YSCALE_REG	0x0330	Source Y scale factor register	0x0001_0000
Destination			
DST_SELECT_REG	0x0400	Destination image selection register	0x0000_0001
DST_BASE_ADDR_REG	0x0404	Destination image base address register	0x0000_0000
DST_STRIDE_REG	0x0408	Destination stride register	0x0000_0000
DST_COLOR_MODE_REG	0x040C	Destination image color mode register	0x0000_0000
DST_LEFT_TOP_REG	0x0410	Destination left top coordinate register	0x0000_0000
DST_RIGHT_BOTTOM_REG	0x0414	Destination right bottom coordinate register	0x0000_0000
DST_A8_RGB_EXT_REG	0x041C	Destination A8 RGB extension register	0x0000_0000
Pattern			
PAT_BASE_ADDR_REG	0x0500	Pattern image base address register	0x0000_0000
PAT_SIZE_REG	0x0504	Pattern image size register	0x0001_0001
PAT_COLOR_MODE_REG	0x0508	Pattern image color mode register	0x0000_0000
PAT_OFFSET_REG	0x050C	Pattern left top coordinate register	0x0000_0000
PAT_STRIDE_REG	0x0510	Pattern stride register	0x0000_0000
Mask			
MSK_BASE_ADDR_REG	0x0520	Mask base address register	0x0000_0000
MSK_STRIDE_REG	0x0524	Mask stride register	0x0000_0000
MSK_LEFT_TOP_REG	0x0528	Mask left top coordinate register	0x0000_0000
MSK_RIGHT_BOTTOM_REG	0x052C	Mask right bottom coordinate register	0x0000_0000
MSK_MODE_REG	0x0530	Mask mode register	0x0000_0000
MSK_REPEAT_MODE_REG	0x0534	Mask repeat mode register	0x0000_0000
MSK_PAD_VALUE_REG	0x0538	Mask pad value register	0x0000_0000
MSK_SCALE_CTRL_REG	0x053C	Mask scale control register	0x0000_0000
MSK_XSCALE_REG	0x0540	Mask X scale factor register	0x0001_0000
MSK_YSCALE_REG	0x0544	Mask Y scale factor register	0x0001_0000
Clipping Window			
CW_LT_REG	0x0600	Coordinates LeftTop of clip window	0x0000_0000
CW_RB_REG	0x0604	Coordinates RightBottom of clip window	0x0000_0000
ROP & Alpha Setting			
THIRD_OPERAND_REG	0x0610	Third operand selection register	0x0000_0011
ROP4_REG	0x0614	Raster operation register	0x0000_CCCC
ALPHA_REG	0x0618	Global alpha value, global color value	0x0000_00FF
Color			
FG_COLOR_REG	0x0700	Foreground color register	0x0000_0000
BG_COLOR_REG	0x0704	Background color register	0x0000_0000
BS_COLOR_REG	0x0708	Blue screen color register	0x0000_0000

Register	Offset	Description	Reset Value
SF_COLOR_REG	0x070C	Solid fill color register	0x0000_0000
Color Key			
C_COLORKEY_CTRL_REG	0x0710	Source colorkey control register	0x0000_0000
SRC_COLORKEY_DR_MIN_REG	0x0714	Source colorkey decision reference Minimum register	0x0000_0000
SRC_COLORKEY_DR_MAX_REG	0x0718	Source colorkey decision reference maximum register	0xFFFF_FFFF
DST_COLORKEY_CTRL_REG	0x071C	Destination colorkey control register	0x0000_0000
DST_COLORKEY_DR_MIN_REG	0x0720	Destination colorkey decision reference minimum register	0x0000_0000
DST_COLORKEY_DR_MAX_REG	0x0724	Destination colorkey decision reference maximum register	0xFFFF_FFFF
Gamma Table			
GAMMA_TABLE0_0_REG	0x0800	Entry 0 of gamma table 0	0x0000_0000
GAMMA_TABLE0_1_REG	0x0804	Entry 1 of gamma table 0	0x0000_0000
GAMMA_TABLE0_2_REG	0x0808	Entry 2 of gamma table 0	0x0000_0000
GAMMA_TABLE0_3_REG	0x080C	Entry 3 of gamma table 0	0x0000_0000
GAMMA_TABLE0_4_REG	0x0810	Entry 4 of gamma table 0	0x0000_0000
GAMMA_TABLE0_5_REG	0x0814	Entry 5 of gamma table 0	0x0000_0000
GAMMA_TABLE0_6_REG	0x0818	Entry 6 of gamma table 0	0x0000_0000
GAMMA_TABLE0_7_REG	0x081C	Entry 7 of gamma table 0	0x0000_0000
GAMMA_TABLE0_8_REG	0x0820	Entry 8 of gamma table 0	0x0000_0000
GAMMA_TABLE0_9_REG	0x0824	Entry 9 of gamma table 0	0x0000_0000
GAMMA_TABLE0_10_REG	0x0828	Entry 10 of gamma table 0	0x0000_0000
GAMMA_TABLE0_11_REG	0x082C	Entry 11 of gamma table 0	0x0000_0000
GAMMA_TABLE0_12_REG	0x0830	Entry 12 of gamma table 0	0x0000_0000
GAMMA_TABLE0_13_REG	0x0834	Entry 13 of gamma table 0	0x0000_0000
GAMMA_TABLE0_14_REG	0x0838	Entry 14 of gamma table 0	0x0000_0000
GAMMA_TABLE0_15_REG	0x083C	Entry 15 of gamma table 0	0x0000_0000
GAMMA_TABLE1_0_REG	0x0840	Entry 0 of gamma table 1	0x0000_0000
GAMMA_TABLE1_1_REG	0x0844	Entry 1 of gamma table 1	0x0000_0000
GAMMA_TABLE1_2_REG	0x0848	Entry 2 of gamma table 1	0x0000_0000
GAMMA_TABLE1_3_REG	0x084C	Entry 3 of gamma table 1	0x0000_0000
GAMMA_TABLE1_4_REG	0x0850	Entry 4 of gamma table 1	0x0000_0000
GAMMA_TABLE1_5_REG	0x0854	Entry 5 of gamma table 1	0x0000_0000
GAMMA_TABLE1_6_REG	0x0858	Entry 6 of gamma table 1	0x0000_0000
GAMMA_TABLE1_7_REG	0x085C	Entry 7 of gamma table 1	0x0000_0000
GAMMA_TABLE1_8_REG	0x0860	Entry 8 of gamma table 1	0x0000_0000

Register	Offset	Description	Reset Value
GAMMA_TABLE1_9_REG	0x0864	Entry 9 of gamma table 1	0x0000_0000
GAMMA_TABLE1_10_REG	0x0868	Entry 10 of gamma table 1	0x0000_0000
GAMMA_TABLE1_11_REG	0x086C	Entry 11 of gamma table 1	0x0000_0000
GAMMA_TABLE1_12_REG	0x0870	Entry 12 of gamma table 1	0x0000_0000
GAMMA_TABLE1_13_REG	0x0874	Entry 13 of gamma table 1	0x0000_0000
GAMMA_TABLE1_14_REG	0x0878	Entry 14 of gamma table 1	0x0000_0000
GAMMA_TABLE1_15_REG	0x087C	Entry 15 of gamma table 1	0x0000_0000
GAMMA_REF_COLOR_REG	0x0880	Gamma table reference color register	0x0000_0000

46.6.2 General Registers

46.6.2.1 SOFT_RESET_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	W	Reserved	0x0
SFRClear	[1]	W	SFR Clear When this bit is set, all SFR registers except related to DMA and some global registers have initial value (reset value). Clears this bit automatically after one clock cycle. (Except Registers: DMA_SFR_BASE_ADDR_REG, DMA_COMMAND_REG, DMA_EXE_LIST_NUM_REG, DMA_STATUS_REG, DMA_HOLD_CMD_REG, AXI_MODE_REG, SOFT_RESET_REG, INTEN_REG)	0x0
R	[0]	W	Software Reset Write to this bit results in a one-cycle reset signal to FIMG2D graphics engine. It assigns the "Reset Value" to every command register and parameter setting register.	0x0

46.6.2.2 INTEN_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	RW	Reserved	0x0
INT_TYPE	[4]	RW	Interrupt type 0 = Level 1 = Edge	0x0
ACF	[3]	RW	All command lists finished interrupt enable. When this bit is set and when the graphics engine finishes the execution of all command lists, an interrupt occurs. The INTP_ACMD_FIN flag in INTC_PEND_REG will be set. (This field is valid when the mode of host interface is DMA Mode.)	0x0
UCF	[2]	RW	One BitBLT with enabled StartNHold finished interrupt enable. When this bit is set and when the graphics engine finishes the execution of one BitBLT with Enabled StartNHold, an interrupt occurs. The INTP_UCMD_FIN flag in INTC_PEND_REG will be set. (This field is valid when the mode of host interface is DMA Mode.)	0x0
GCF	[1]	RW	A command lists finished interrupt enable. When this bit is set and when the graphics engine finishes the execution of a command list, an interrupt occurs. The INTP_GCMD_FIN flag in INTC_PEND_REG will be set. (This field is valid when the mode of host interface is DMA Mode.)	0x0
SCF	[0]	RW	A BitBLT finished interrupt enable. When this bit is set and when the graphics engine finishes the execution of a BitBLT, an interrupt occurs. The INTP_SCMD_FIN flag in INTC_PEND_REG will be set.	0x0

46.6.2.3 INTC_PEND_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	RW	Reserved	0x0
INTP_ACMD_FIN	[3]	RW	All command lists finished interrupt flag. Writing "1" to this bit clears this flag Clear this bit before Start_DMA because of previous Start_DMA's residue. (This field is valid when the mode of host interface is DMA Mode.)	0x0
INTP_UCMD_FIN	[2]	RW	One BilBLT with StartNHold finished interrupt flag. Writing "1" to this bit clears this flag Clear this bit before Start_BITBLT because of previous Start_BITBLT's residue. (This field is valid when the mode of host interface is DMA Mode.)	0x0
INTP_GCMD_FIN	[1]	RW	A Command list finished interrupt flag. Writing "1" to this bit clears this flag (This field is valid when the mode of host interface is DMA Mode.)	0x0
INTP_SCMD_FIN	[0]	RW	A BilBLT finished interrupt flag, Writing "1" to this bit clears this flag Clear this bit before Start_BITBLT because of previous Start_BITBLT's residue.	0x0

46.6.2.4 FIFO_STAT_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	R	Reserved	0x0
CMD_FIN	[0]	R	0 = In the middle of rendering process. 1 = The graphics engine finishes the execution of command. NOTE: In case that the AXI clock (Core clock) is slower than APB clock, it must perform the read operation of this register twice to confirm the state of this bit.	0x1

46.6.2.5 AXI_MODE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x001C, Reset Value = 0x0200_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	RW	Reserved	0x0
MaxBurstLength	[25:24]	RW	AXI MASTER I/F of FIMG-2D version 4.1 generates maximum burst length 2'b00 = 2 burst: This means the range of burst length value is from 4'b0000 to 4'b0001) 2'b01 = 4 burst: This means the range of burst length value is from 4'b0000 to 4'b0011) 2'b10 = 8 burst: This means the range of burst length value is from 4'b0000 to 4'b0111) 2'b11 = 16 burst: This means the range of burst length value is from 4'b0000 to 4'b1111)	0x2
RSVD	[23:21]	RW	Reserved	0x0
AWUSERS	[20:16]	RW	These bits decide AWUSERS signal of AXI MASTER I/F. The meaning of values is same as that of AMBA AXI specification.	0x0
RSVD	[15:13]	RW	Reserved	0x0
ARUSERS	[12:8]	RW	These bits decide ARUSERS signal of AXI MASTER I/F. The meaning of values is same as that of AMBA AXI Specification.	0x0
AWCACHE	[7:4]	RW	These bits decide AWCACHE signal of AXI MASTER I/F. The meaning of values is same as that of AMBA AXI Specification.	0x0
ARCACHE	[3:0]	RW	These bits decide ARCACHE signal of AXI MASTER I/F. The meaning of values is same as that of AMBA AXI Specification.	0x0

46.6.2.6 DMA_SFR_BASE_ADDR_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DMA BaseAddr	[31:0]	RW	First Command list Base Address	0x0000_0000

46.6.2.7 DMA_COMMAND_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0084, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	W	Reserved	0x0000_0000
DMA Halt	[2]	W	Halts the DMA mode after running the current BitBLT is complete. This is different from Software Reset that immediately stops running BitBLT. As soon as it asserts this signal, it clears INTC_PEND_REG. It clears automatically after one-clock cycle.	0x0
DMA Continue	[1]	W	When this bit is set, the holding hardware continues to operate next BitBlt after clearing INTC_PEND_REG. It clears automatically after one-clock cycle.	0x0
DMA Start	[0]	W	Starts BitBlt Operation as DMA mode. When this bit is set, It clears automatically after one-clock cycle.	0x0

46.6.2.8 DMA_EXE_LIST_NUM_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0088, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	RW	Reserved	0x0000_0000
DMA Execution List Number for Done	[7:0]	RW	Finishes DMA mode when last BitBlt of set list by this field is complete. If this field is "0", it finishes DMA Mode when last BitBlt of last list is complete. List IDs automatically and sequentially are assigned from "1" be N.	0x0

46.6.2.9 DMA_STATUS_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x008C, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	R	Reserved	0x0000_0000
DMA List Done Count	[24:17]	R	After DMA starts, when all BitBLT of one commands list is complete, it counts this field. This field is cleared when DMA Start is set	0x0
DMA BITBLT Done Count	[16:1]	R	After DMA starts, when a BitBLT is complete, it counts this field. This field is cleared when DMA Start is set.	0x0
DMA Done	[0]	R	When all BitBLT by DMA_EXE_LIST_NUM_REG are complete, this bit becomes 1'b1.	0x1

46.6.2.10 DMA_HOLD_CMD_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	RW	Reserved	0x0000_0000
User Hold	[2]	RW	When this is 1'b1, the hardware will be on hold when BitBLT with enabling StartNHold is complete. When "DMA continue" of DMA_COMMAND is set, it runs the next BitBLT.	0x0
LIST Hold	[1]	RW	When this is 1'b1, the hardware will be on hold after all BitBLT of a List is complete. When "DMA continue" of DMA_COMMAND is set, it runs the next BitBLT of next list.	0x0
BITBLT Hold	[0]	RW	If this is 1'b1, the hardware becomes holding per a BitBLT t. When "DMA continue" of DMA_COMMAND is set, it runs the next BitBLT.	0x0

46.6.3 Command Registers

46.6.3.1 BITBLT_START_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0100, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	W	Reserved	0x0
StartCaseSel	[2]	W	This bit decides when current BitBLT starts. 1'b0 = Waits Write Response of previous BitBLT 1'b1 = Does not wait Write Response of previous BitBLT	0x0
StartNHold	[1]	W	This bit decides whether it holds state after current BitBLT or immediately starts next BitBLT 1'b0 = Does not wait "DMA_continue" of DMA_COMMAND 1'b1 = Waits "DMA_continue" of DMA_COMMAND for running next BitBLT	0x0
Start_BitBLT	[0]	W	Starts BitBLT operation. When this bit is set, it automatically clears after one-clock cycle.	0x0

46.6.3.2 BITBLT_COMMAND_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0104, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0-
FastSolidColorFill	[28]	RW	Fast solid color fill 1'b0 = No fast solid color fill. 1'b1 = Fast solid color fill, ignores below register setting	0x0
DstWDepre	[27]	RW	Final alpha depremultiplies final RGBs. 1'b0 = Does not depremultiply 1'b1 = Depremultiplies	0x0
DstPre	[26]	RW	Destination alpha premultiplies destination read RGBs 1'b0 = Does not premultiply 1'b1 = Premultiplies	0x0
PatPre	[25]	RW	Pattern alpha premultiplies pattern RGBs. 1'b0 = Does not premultiply 1'b1 = Premultiplies	0x0
SrcPre	[24]	RW	Source alpha premultiplies source RGBs. 1'b0 = Does not premultiply 1'b1 = Premultiplies	0x0
RSVD	[23:21]	RW	Reserved	–
AlphaBlendMode	[20]	RW	Alpha Blending Mode 1'b0 = No Alpha Blending 1'b1 = Alpha Blending	0x0
ColorKeyMode	[19:16]	RW	4'b0000 = Disables colorkey 4'bXXX1 = Enables source RGBA colorkey 4'bXX1X = Enables destination RGBA colorkey	0x0
RSVD	[15:14]	RW	Reserved	0x0
Transparent Mode	[13:12]	RW	2'b00 = Opaque mode 2'b01 = Transparent mode 2'b10 = BlueScreen mode 2'b11 = Reserved	0x0
RSVD	[11:9]	RW	Reserved	0x0
CWEn	[8]	RW	Enables clipping window	0x0
RSVD	[7:4]	RW	Reserved	0x0
DitherEn	[3]	RW	1'b0 = Disables Dithering 1'b1 = Enables Dithering	–
ROPAlphaEn	[2]	RW	1'b0 = Selects source alpha for alpha blending 1'b1 = Selects ROP alpha for alpha blending	0x0
MaskROP4En	[1:0]	RW	2'b00 = Disables masking operation and mask for ROP4 2'b01 = Enables mask for ROP4 (ROP selection enable) 2'b10 = Enables masking operation 2'b11 = Enables masking operation and mask for ROP4	0x0

ColorKey Mode and Transparent Mode Limitation: If you select source code as the foreground color or the background color, it disables the source colorkey. It regards transparent mode as opaque mode because the colorkeying for the foreground color and the background color set by user is meaningless. When you select the destination color as foreground color or the background color, it also disables the destination colorkey.

46.6.3.3 BLEND_FUNCTION_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0108, Reset Value = 0x0040_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:23]	RW	Reserved	–
WinCESrcOver	[22]	RW	When "1" Src_Over blending mode for WinCE Ignores Below Register setting, Auto Generates Coefficients and Round Mode in hardware for WinCE conformance.	0x1
Darken	[21]	RW	When "1" Darken blending mode Ignores Below Register setting, Auto Generates Coefficients and Round Mode in hardware (Almost half pixel rate performance)	0x0
Lighten	[20]	RW	When "1" Lighten blending mode Ignores Below Register setting, Auto Generates Coefficients and Round Mode in hardware (Almost half pixel rate performance)	0x0
RSVD	[19]	RW	Reserved	0x0
InvDstCoeff	[18]	RW	Inverse destination color coefficient 0 = Does not inverse 1 = Inverses destination color coefficient (0xFF: Destination color coefficient)	0x0
RSVD	[17]	RW	Reserved	0x0
InvSrcCoeff	[16]	RW	Inverse source color coefficient 0 = Does not inverse 1 = Inverses source color coefficient (0xFF: Source color coefficient)	0x0
DstCoeffDstAlpha	[15:14]	RW	Destination alpha for destination coefficient 2'b00 = Destination alpha 2'b01 = Destination alpha + Global alpha 2'b10 = Destination alpha × Global alpha 2'b11 = Reserved	0x0
DstCoeffSrcAlpha	[13:12]	RW	Source alpha for destination coefficient 2'b00 = Source alpha 2'b01 = Source alpha + Global alpha 2'b10 = Source alpha × Global alpha 2'b11 = Reserved	0x0
DstCoeff	[11:8]	RW	Destination color coefficient 4'b0000 = ONE	0x0

Name	Bit	Type	Description	Reset Value
			4'b0001 = ZERO 4'b0010 = SRC_Alpha 4'b0011 = SRC_COLOR 4'b0100 = DST_ALPHA 4'b0101 = DST_COLOR 4'b0110 = GB_ALPHA 4'b0111 = GB_COLOR 4'b1000 = DISJOINT_S ((1 – Source alpha)/Destination alpha) 4'b1001 = DISJOINT_D ((1 – Destination alpha)/Source alpha) 4'b1010 = CONJOINT_S (Source alpha/Destination alpha) 4'b1011 = CONJOINT_D (Destination Alpha/Source alpha) 4'b1100 to 4'b1111 = Reserved	
SrcCoeffDstAlpha	[7:6]	RW	Destination alpha for source coefficient 2'b00 = Destination alpha 2'b01 = Destination alpha + Global alpha 2'b10 = Destination alpha × Global alpha 2'b11 = Reserved	0x0
SrcCoeffSrcAlpha	[5:4]	RW	Source alpha for source coefficient 2'b00 = Source alpha 2'b01 = Source alpha + Global alpha 2'b10 = Source alpha × Global alpha 2'b11 = Reserved	0x0
SrcCoeff	[3:0]	RW	Source coefficient 4'b0000 = ONE 4'b0001 = ZERO 4'b0010 = SRC_ALPHA 4'b0011 = SRC_COLOR 4'b0100 = DST_ALPHA 4'b0101 = DST_COLOR 4'b0110 = GB_ALPHA 4'b0111 = GB_COLOR 4'b1000 = DISJOINT_S ((1-Source alpha)/Destination alpha) 4'b1001 = DISJOINT_D ((1-Destination alpha)/Source alpha) 4'b1010 = CONJOINT_S (Source alpha/Destination alpha) 4'b1011 = CONJOINT_D (Destination alpha/Source alpha) 4'b1100 to 4'b1111 = Reserved	0x0

46.6.3.4 ROUND_MODE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x010C, Reset Value = 0x0000_0233

Name	Bit	Type	Description	Reset Value
RSVE	[31:10]	RW	Reserved	0x0
RoundModeDepre	[9:8]	RW	Round mode in alpha depremultiply 2'b00 = Result = ((A + 1) × B) >> 8; 2'b01 = Result = ((A+ (A >>7)) × B) >> 8; 2'b10 = Result_tmp = A × B + 0x80; Result = (Result_tmp + (Result_tmp >> 8))>> 8; 2'b11 = Reserved	0x2
RSVD	[7:6]	RW	Reserved	0x0
RoundModePre	[5:4]	RW	Round mode in alpha premultiply 2'b00 = Result = (A × B) >> 8; 2'b01 = Result = ((A + 1) × B) >> 8; 2'b10 = Result = ((A+ (A>>7)) × B) >> 8; 2'b11 = Result_tmp = A × B + 0x80; Result = (Result_tmp + (Result_tmp >> 8)) >> 8;	0x3
RSVD	[3:2]	RW	Reserved	0x0
RoundModeBl	[1:0]	RW	Round mode in blending 2'b00 = Result = ((A + 1) × B) >> 8; 2'b01 = Result = ((A + (A >> 7)) × B) >> 8; 2'b10 = Result_tmp = A × B + 0x80; Result = (Result_tmp + (Result_tmp >> 8)) >> 8; 2'b11 = Result_A = A × B; (16 bpp) Result_B = C × D; (16 bpp) Result_tmp = Result_A + Result_B + 0x80; Result = (Result_tmp + (Result_tmp >> 8)) >> 8;	0x3

46.6.4 Parameter Setting Registers

46.6.4.1 ROTATE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0200, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	RW	Reserved	0x0
MskRotate	[8]	RW	0 = No Rotation 1 = 90 Degree rotation	0x0
RSVD	[7:5]	RW	Reserved	0x0
PatRotate	[4]	RW	0 = No rotation 1 = 90 Degree rotation	0x0
RSVD	[3:1]	RW	Reserved	0x0
SrcRotate	[0]	RW	0 = No Rotation 1 = 90 Degree rotation	0x0

46.6.4.2 SRC_MSK_DIRECT_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0204, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	RW	Reserved	0x0
MskYDirect	[5]	RW	0 = Y Positive 1 = Y Negative	0x0
MskXDirect	[4]	RW	0 = X Positive 1 = X Negative	0x0
RSVD	[3:2]	RW	Reserved	0x0
SrcYDirect	[1]	RW	0 = Y Positive 1 = Y Negative	0x0
SrcXDirect	[0]	RW	0 = X Positive 1 = X Negative	0x0

46.6.4.3 DST_PAT_DIRECT_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0208, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	RW	Reserved	0x0
PatYDirect	[5]	RW	0 = Y Positive 1 = Y Negative	0x0
PatXDirect	[4]	RW	0 = X Positive 1 = X Negative	0x0
RSVD	[3:2]	RW	Reserved	0x0
DstYDirect	[1]	RW	0 = Y Positive 1 = Y Negative	0x0
DstXDirect	[0]	RW	0 = X Positive 1 = X Negative	0x0

46.6.5 Source

46.6.5.1 SRC_SELECT_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0300, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	RW	Reserved	0x0
SrcSelect	[1:0]	RW	Select source 2'b00 = Normal mode (Using source image in the external memory) 2'b01 = Uses foreground color as source image 2'b10 = Uses background color as source image 2'b11 = Reserved	0x1

46.6.5.2 SRC_BASE_ADDR_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0304, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SrcAddr	[31:0]	RW	Base address of the source image in RGBA color mode	0x0

46.6.5.3 SRC_STRIDE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0308, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	RW	Reserved	0x0
SrcStride	[15:0]	RW	Source stride (2's complement value).	0x0

46.6.5.4 SRC_COLOR_MODE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x030C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	RW	Reserved	0x0
SrcChannelOrder	[5:4]	RW	2'b00 = {A, X}RGB 2'b01 = RGB{A, X} 2'b10 = {A, X}BGR 2'b11 = BGR{A, X}	0x0
SrcColorFormat	[3:0]	RW	4'b0000 = XRGB_8888 4'b0001 = ARGB_8888 4'b0010 = RGB_565 4'b0011 = XRGB_1555 4'b0100 = ARGB_1555 4'b0101 = XRGB_4444 4'b0110 = ARGB_4444 4'b0111 = PACKED_RGB_888 4'b1011 = A8 (Alpha 8 bits) 4'b1100 = L8 (Luminance 8 bits = gray color)	0x0

46.6.5.5 SRC_LEFT_TOP_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0310, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
SrcTopY	[28:16]	RW	Left Top Y coordinate of source image Range = 0 to 8000 (Requirement: SrcTopY < SrcBottomY)	0x0
RSVD	[15:13]	RW	Reserved	0x0
SrcLeftX	[12:0]	RW	Left Top X coordinate of source image Range = 0 to 8000 (Requirement: SrcLeftX < SrcRightX)	0x0

46.6.5.6 SRC_RIGHT_BOTTOM_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0314, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
SrcBottomY	[28:16]	RW	Right bottom Y coordinate of source image Range = 0 to 8000	0x0
RSVD	[15:13]	RW	Reserved	0x0
SrcRightX	[12:0]	RW	Right bottom X coordinate of source image Range = 0 to 8000	0x0

46.6.5.7 SRC_REPEAT_MODE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x031C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	RW	Reserved	0x0
SrcRepeatMode	[2:0]	RW	Repeat mode 3'b000 = Repeat 3'b001 = Pad 3'b010 = Reflect 3'b011 = Clamp 3'b100 = None	0x0

46.6.5.8 SRC_PAD_VALUE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0320, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SrcPadValue	[31:0]	RW	Color for repeat pad mode The format and the channel order of this value are same as SrcColorFormat and SrcChannelOrder of SRC_COLOR_MODE_REG register.	0x0

46.6.5.9 SRC_A8_RGB_EXT_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0324, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	RW	Reserved	0x0
SrcA8RGB	[23:0]	RW	RGB value for A8 format RGB value has 888 format and RGB order.	0x0

46.6.5.10 SRC_SCALE_CTRL_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0328, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	RW	Reserved	0x0
SrcScaleCtrl	[1:0]	RW	Source scale control 2'b00 = Disables scaling 2'b01 = Enables scaling: nearest sampling 2'b10 = Enables scaling: bilinear sampling 2'b11 = Reserved	0x0

46.6.5.11 SRC_XSCALE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x032C, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
SrcXScale	[28:0]	RW	X Scale ratio-unsigned fixed point value Integer part = [28:16] Fraction part = [15:0] Default value = 1.0 (0x10000)	0x10000

46.6.5.12 SRC_YSCALE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0330, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
SrcYScale	[28:0]	RW	Y Scale ratio-unsigned fixed point value Integer part = [28:16] Fraction part = [15:0] Default value = 1.0 (0x10000)	0x10000

46.6.6 Destination

46.6.6.1 DST_SELECT_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0400, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	RW	Reserved	0x0
DstSelect	[1:0]	RW	Select destination 2'b00 = Normal mode (using destination image in the external memory) 2'b01 = Uses foreground color as destination image 2'b10 = Uses background color as destination image 2'b11 = Reserved	0x1

46.6.6.2 DST_BASE_ADDR_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0404, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DstAddr	[31:0]	RW	Base address of the destination image	0x0

46.6.6.3 DST_STRIDE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0408, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	RW	Reserved	0x0
DstStride	[15:0]	RW	Destination stride (2's complement value).	0x0

46.6.6.4 DST _COLOR_MODE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x040C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	RW	Reserved	0x0
DstChannelOrder	[5:4]	RW	2'b00 = {A, X}RGB 2'b01 = RGB{A, X} 2'b10 = {A, X}BGR 2'b11 = BGR{A, X}	0x0
DstColorFormat	[3:0]	RW	4'b0000 = XRGB_8888 4'b0001 = ARGB_8888 4'b0010 = RGB_565 4'b0011 = XRGB_1555 4'b0100 = ARGB_1555 4'b0101 = XRGB_4444 4'b0110 = ARGB_4444 4'b0111 = PACKED_RGB_888 4'b1011 = A8 (Alpha 8 bits) 4'b1100 = L8 (Luminance 8 bits: gray color)	0x0

46.6.6.5 DST _LEFT_TOP_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0410, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
DstTopY	[28:16]	RW	Left Top Y coordinate of destination image Range = 0 to 8000 (Requirement: DstTopY < DstBottomY)	0x0
RSVD	[15:13]	RW	Reserved	0x0
DstLeftX	[12:0]	RW	Left Top X coordinate of destination image Range = 0 to 8000 (Requirement: DstLeftX < DstRightX)	0x0

46.6.6.6 DST _RIGHT_BOTTOM_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0414, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
DstBottomY	[28:16]	RW	Right bottom Y coordinate of destination image Range = 0 to 8000 (Requirement: DstTopY < DstBottomY)	0x0
RSVD	[15:13]	RW	Reserved	0x0
DstRightX	[12:0]	RW	Right bottom X coordinate of destination image Range = 0 to 8000 (Requirement: DstLeftX < DstRightX)	0x0

46.6.6.7 DST_A8_RGB_EXT_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x041C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	RW	Reserved	0x0
DstA8RGB	[23:0]	RW	RGB value for A8 format RGB value has 888 format and RGB order.	0x0

46.6.7 Pattern

46.6.7.1 PAT_BASE_ADDR_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0500, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PatAddr	[31:0]	RW	Base address of the pattern image	0x0

46.6.7.2 PAT_SIZE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0504, Reset Value = 0x0001_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
PatHeight	[28:16]	RW	Height of pattern image. Range = 1 to 8000	0x1
RSVD	[15:13]	RW	Reserved	0x0
PatWidth	[12:0]	RW	Width of pattern image. Range = 1 to 8000.	0x1

46.6.7.3 PAT_COLOR_MODE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0508, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Reserved	[31:6]	RW	Reserved	0x0
PatChannelOrder	[5:4]	RW	2'b00 = {A, X}RGB 2'b01 = RGB{A, X} 2'b10 = {A, X}BGR 2'b11 = BGR{A, X}	0x0
Reserved	[3]	RW	Reserved	0x0
PatColorFormat	[2:0]	RW	3'b000 = XRGB_8888 3'b001 = ARGB_8888 3'b010 = RGB_565 3'b011 = XRGB_1555 3'b100 = ARGB_1555 3'b101 = XRGB_4444 3'b110 = ARGB_4444 3'b111 = PACKED_RGB_888	0x0

46.6.7.4 PAT_OFFSET_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x050C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
PatOffsetY	[28:16]	RW	Y value of pattern offset. Range = 0 to 7999	0x0
RSVD	[15:13]	RW	Reserved	0x0
PatOffsetX	[12:0]	RW	X value of pattern offset. Range = 0 to 7999.	0x0

46.6.7.5 PAT_STRIDE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0510, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	RW	Reserved	0x0
PatStride	[15:0]	RW	Pattern stride (2's complement value)	0x0

46.6.8 Mask

It uses this mask for ROP4 operation or masking the image data. The data format of that can be 1-bit, 4 bits, 8 bits, 16 bits, or 32 bits.

46.6.8.1 MSK_BASE_ADDR_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0520, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MskAddr	[31:0]	RW	Base address of the mask image	0x0

46.6.8.2 MSK_STRIDE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0524, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	RW	Reserved	0x0
MskStride	[15:0]	RW	Mask stride (2's complement value).	0x0

46.6.8.3 MSK_LEFT_TOP_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0528, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
MskTopY	[28:16]	RW	Left top Y coordinate of mask image Range = 0 to 8000	0x0
RSVD	[15:13]	RW	Reserved	0x0
MskLeftX	[12:0]	RW	Left Top X coordinate of mask image Range = 0 to 8000	0x0

46.6.8.4 MSK_RIGHT_BOTTOM_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x052C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
MskBottomY	[28:16]	RW	Right bottom Y coordinate of mask image Range = 0 to 8000	0x0
RSVD	[15:13]	RW	Reserved	0x0
MskRightX	[12:0]	RW	Right bottom X coordinate of mask image Range = 0 to 8000	0x0

46.6.8.5 MSK_MODE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0530, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	RW	Reserved	0x0
MskOpType	[9:8]	RW	Defines this type in only 32-bit/16-bit mask mode 2'b00 = Uses only alpha channel as mask 2'b01 = Uses component blend with mask 2'b10 = Uses channel mask 2'b11 = Reserved	0x0
RSVD	[7:6]	RW	Reserved	0x0
MskChannelOrder	[5:4]	RW	2'b00 = {A, X}RGB 2'b01 = RGB{A, X} 2'b10 = {A, X}BGR 2'b11 = BGR{A, X}	0x0
MskMode	[3:0]	RW	Mask mode 4'b0000 = 1-bit mask mode 4'b0001 = 4-bit mask mode 4'b0010 = 8-bit mask mode 4'b0011 = 1-6bit mask mode (565) 4'b0100 = 1-6bit mask mode (1555) 4'b0101 = 1-6bit mask mode (4444) 4'b0110 = 32-bit mask mode (8888) 4'b0111 = 4-bit mask mode for WinCE antialiased font	0x0

46.6.8.6 MSK_REPEAT_MODE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0534, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	RW	Reserved	0x0
MskRepeatMode	[1:0]	RW	Repeat mode 2'b00 = Repeat 2'b01 = Pad 2'b10 = Reflect 2'b11 = Clamp	0x0

46.6.8.7 MSK_PAD_VALUE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0538, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MskPadValue	[31:0]	RW	Color for repeat pad mode The mode and the channel order of this value are same as MskMode and MskChannelOrder of MSK_MODE_REG register.	0x0

46.6.8.8 MSK_SCALE_CTRL_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x053C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	RW	Reserved	0x0
MskScaleCtrl	[1:0]	RW	Mask scale control 2'b00 = Disables scale 2'b01 = Nearest sampling 2'b10 = Bilinear sampling 2'b11 = Reserved	0x0

46.6.8.9 MSK_XSCALE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0540, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
MskXScale	[28:0]	RW	X Scale ratio Unsigned fixed point value Integer part = [28:16] Fraction part = [15:0] Default value = 1.0 (0x10000)	0x10000

46.6.8.10 MSK_YSCALE_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0544, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
MskYScale	[28:0]	RW	Y Scale ratio Unsigned fixed point value Integer part = [28:16] Fraction part = [15:0] Default value = 1.0 (0x10000)	0x10000

46.6.9 Clipping Window

46.6.9.1 CW_LT_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0600, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
CWTopY	[28:16]	RW	Top Y clipping window Requirement = DstTopY ≤ CWTopY < CWBottomY	0x0
RSVD	[15:13]	RW	Reserved	0x0
CWLeftX	[12:0]	RW	Left X coordinate of clipping window. Requirement = DstLeftX ≤ CWLeftX < CWRightX	0x0

46.6.9.2 CW_RB_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0604, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	RW	Reserved	0x0
CWBottomY	[28:16]	RW	Bottom Y clipping window Requirement: CWBottomY ≤ DstBottomY	0x0
RSVD	[15:13]	RW	Reserved	0x0
CWRightX	[12:0]	RW	Right X clipping window Requirement: CWRightX ≤ DstRightX	0x0

46.6.10 ROP & Alpha Setting

46.6.10.1 THIRD_OPERAND_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0610, Reset Value = 0x0000_0011

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	RW	Reserved	0x0
MaskedSelect	[5:4]	RW	2'00 = Pattern 2'01 = Foreground color 2'10 = Background color Others = Reserved	0x1
RSVD	[3:2]	RW	Reserved	0x0
UnmaskedSelect	[1:0]	RW	2'00 = Pattern 2'01 = Foreground color 2'10 = Background color Others = Reserved	0x1

46.6.10.2 ROP4_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0614, Reset Value = 0x0000_CCCC

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	RW	Reserved	0x0
MaskedROP3	[15:8]	RW	Raster operation value	0xCC
UnmaskedROP3	[7:0]	RW	Raster operation value	0xCC

46.6.10.3 ALPHA_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0618, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
ColorValue	[31:8]	RW	Global color value (RGB order)	0x0
AlphaValue	[7:0]	RW	Global alpha value	0xFF

46.6.11 Color

46.6.11.1 FG_COLOR_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0700, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ForegroundColor	[31:0]	RW	Foreground color value.	0x0

The color format of the foreground color is the ARGB_8888 format and the channel order is ARGB.

46.6.11.2 BG_COLOR_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0704, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
BackgroundColor	[31:0]	RW	Background color value.	0x0

The color format of the background color is the ARGB_8888 format and the channel order is ARGB.

46.6.11.3 BS_COLOR_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0708, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
BlueScreenColor	[31:0]	RW	BlueScreen color value. Discards the alpha field of the blue screen color.	0x0

The color format of the bluescreen color is generally the same as the source color format.

46.6.11.4 SF_COLOR_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x070C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SolidFillColor	[31:0]	RW	Solid Fill color value.	0x0

The color format of the solid fill color is the same as the destination color format (DST_COLOR_MODE_REG). The channel order of solid fill color is set as ARGB.

When Solid Fill Color is ARGB1555 format, SF_COLOR_REG that is set to:

- SolidFillColor[15] is A
- SolidFillColor[14:10] is R
- SolidFillColor[9:5] is G
- SolidFillColor[4:0] is B

46.6.12 Color Key

46.6.12.1 SRC_COLORKEY_CTRL_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0710, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	RW	Reserved	0x0
SrcStencilInv	[16]	RW	0 = Normal stencil test 1 = Inversed stencil test	0x0
RSVD	[15:13]	RW	Reserved	–
SrcStencilOnA	[12]	RW	0 = Stencil test off for A value 1 = Stencil test on for A value	0x0
RSVD	[11:9]	RW	Reserved	–
SrcStencilOnR	[8]	RW	0 = Stencil test off for R value 1 = Stencil test on for R value	0x0
RSVD	[7:5]	RW	Reserved	–
SrcStencilOnG	[4]	RW	0 = Stencil test off for G value 1 = Stencil test on for G value	0x0
RSVD	[3:1]	RW	Reserved	–
SrcStencilOnB	[0]	RW	0 = Stencil test off for B value 1 = Stencil test on for B value	0x0

46.6.12.2 SRC_COLORKEY_DR_MIN_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0714, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SrcDRMinA	[31:24]	RW	Minimum value for alpha DR	0x0
SrcDRMinR	[23:16]	RW	Minimum value for RED DR	0x0
SrcDRMinG	[15:8]	RW	Minimum value for GREEN DR	0x0
SrcDRMinB	[7:0]	RW	Minimum value for BLUE DR	0x0

The color format of source colorkey decision reference register is generally the same as the source color format.

But when you select the source color as the foreground color or the background color, it does not activate the source colorkey operation because the colorkeying for the foreground color and the background color set by user is meaningless.

46.6.12.3 SRC_COLORKEY_DR_MAX_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0718, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
SrcDRMaxA	[31:24]	RW	Maximum value for alpha DR	0xFF
SrcDRMaxR	[23:16]	RW	Maximum value for RED DR	0xFF
SrcDRMaxG	[15:8]	RW	Maximum value for BLUE DR	0xFF
SrcDRMaxB	[7:0]	RW	Maximum value for BLUE DR	0xFF

The color format of source colorkey decision reference register is generally the same as the source color format.

But when you select the source color as the foreground color or the background color, it does not activate the source colorkey operation because the colorkeying for the foreground color and the background color set by user is meaningless.

46.6.12.4 DST_COLORKEY_CTRL_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x071C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	RW	Reserved	0x0
DstStencilInv	[16]	RW	0 = Normal stencil test 1 = Inversed stencil test	0x0
RSVD	[15:13]	RW	Reserved	–
DstStencilOnA	[12]	RW	0 = Stencil test off for A value 1 = Stencil test on for A value	0x0
RSVD	[11:9]	RW	Reserved	–
DstStencilOnR	[8]	RW	0 = Stencil test off for R value 1 = Stencil test on for R value	0x0
RSVD	[7:5]	RW	Reserved	–
DstStencilOnG	[4]	RW	0 = Stencil test off for G value 1 = Stencil test on for G value	0x0
RSVD	[3:1]	RW	Reserved	–
DstStencilOnB	[0]	RW	0 = Stencil test off for B value 1 = Stencil test on for B value	0x0

46.6.12.5 DST_COLORKEY_DR_MIN_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0720, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DstDRMinA	[31:24]	RW	Minimum value for alpha DR	0x0
DstDRMinR	[23:16]	RW	Minimum value for RED DR	0x0
DstDRMinG	[15:8]	RW	Minimum value for GREEN DR	0x0
DstDRMinB	[7:0]	RW	Minimum value for BLUE DR	0x0

The color format of destination colorkey decision reference register is the same as the destination color format.

But when you select the source color as the foreground color or the background color, it does not activate the source colorkey operation because the colorkeying for the foreground color and the background color set by user is meaningless.

46.6.12.6 DST_COLORKEY_DR_MAX_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0724, Reset Value = 0xFFFF_FFFF

Name	Bit	Type	Description	Reset Value
DstDRMaxA	[31:24]	RW	Maximum value for alpha DR	0xFF
DstDRMaxR	[23:16]	RW	Maximum value for RED DR	0xFF
DstDRMaxG	[15:8]	RW	Maximum value for GREEN DR	0xFF
DstDRMaxB	[7:0]	RW	Maximum value for BLUE DR	0xFF

The color format of destination colorkey decision reference register is the same as the destination color format.

But when you select the source color as the foreground color or the background color, it does not activate the source colorkey operation because the colorkeying for the foreground color and the background color set by user is invalid.

46.6.13 Gamma Table

The gamma table consists of two sets of 16-bit integer numbers. You can use this table for 4-bit masking (WinCE Antialiased Font).

46.6.13.1 GAMMA_TABLE0_0_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0800, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_0	[31:0]	RW	Gamma table = Entry 0 of set 0	0x0

46.6.13.2 GAMMA_TABLE0_1_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0804, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_1	[31:0]	RW	Gamma table = Entry 1 of set 0	0x0

46.6.13.3 GAMMA_TABLE0_2_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0808, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_2	[31:0]	RW	Gamma table = Entry 2 of set 0	0x0

46.6.13.4 GAMMA_TABLE0_3_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x080C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_3	[31:0]	RW	Gamma table = Entry 3 of set 0	0x0

46.6.13.5 GAMMA_TABLE0_4_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0810, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_4	[31:0]	RW	Gamma table = Entry 4 of set 0	0x0

46.6.13.6 GAMMA_TABLE0_5_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0814, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_5	[31:0]	RW	Gamma table = Entry 5 of set 0	0x0

46.6.13.7 GAMMA_TABLE0_6_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0818, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_6	[31:0]	RW	Gamma table = entry 6 of set 0	0x0

46.6.13.8 GAMMA_TABLE0_7_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x081C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_7	[31:0]	RW	Gamma table = Entry 7 of set 0	0x0

46.6.13.9 GAMMA_TABLE0_8_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0820, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_8	[31:0]	RW	Gamma table = Entry 8 of set 0	0x0

46.6.13.10 GAMMA_TABLE0_9_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0824, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_9	[31:0]	RW	Gamma table = Entry 9 of set 0	0x0

46.6.13.11 GAMMA_TABLE0_10_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0828, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_10	[31:0]	RW	Gamma table = Entry 10 of set 0	0x0

46.6.13.12 GAMMA_TABLE0_11_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x082C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_11	[31:0]	RW	Gamma table = Entry 11 of set 0	0x0

46.6.13.13 GAMMA_TABLE0_12_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0830, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_12	[31:0]	RW	Gamma table = Entry 12 of set 0	0x0

46.6.13.14 GAMMA_TABLE0_13_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0834, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_13	[31:0]	RW	Gamma table = Entry 13 of set 0	0x0

46.6.13.15 GAMMA_TABLE0_14_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0838, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_14	[31:0]	RW	Gamma table = Entry 14 of set 0	0x0

46.6.13.16 GAMMA_TABLE0_15_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x083C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable0_15	[31:0]	RW	Gamma table = Entry 15 of set 0	0x0

46.6.13.17 GAMMA_TABLE1_0_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0840, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_0	[31:0]	RW	Gamma table = Entry 0 of set 1	0x0

46.6.13.18 GAMMA_TABLE1_1_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0844, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_1	[31:0]	RW	Gamma table = Entry 1 of set 1	0x0

46.6.13.19 GAMMA_TABLE1_2_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0848, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_2	[31:0]	RW	Gamma table = Entry 2 of set 1	0x0

46.6.13.20 GAMMA_TABLE1_3_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x084C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_3	[31:0]	RW	Gamma table = Entry 3 of set 1	0x0

46.6.13.21 GAMMA_TABLE1_4_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0850, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_4	[31:0]	RW	Gamma table = Entry 4 of set 1	0x0

46.6.13.22 GAMMA_TABLE1_5_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0854, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_5	[31:0]	RW	Gamma table = Entry 5 of set 1	0x0

46.6.13.23 GAMMA_TABLE1_6_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0858, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_6	[31:0]	RW	Gamma table = Entry 6 of set 1	0x0

46.6.13.24 GAMMA_TABLE1_7_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x085C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_7	[31:0]	RW	Gamma table = Entry 7 of set 1	0x0

46.6.13.25 GAMMA_TABLE1_8_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0860, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_8	[31:0]	RW	Gamma table = Entry 8 of set 1	0x0

46.6.13.26 GAMMA_TABLE1_9_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0864, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_9	[31:0]	RW	Gamma table = Entry 9 of set 1	0x0

46.6.13.27 GAMMA_TABLE1_10_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0868, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_10	[31:0]	RW	Gamma table = Entry 10 of set 1	0x0

46.6.13.28 GAMMA_TABLE1_11_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x086C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_11	[31:0]	RW	Gamma table = Entry 11 of set 1	0x0

46.6.13.29 GAMMA_TABLE1_12_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0870, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_12	[31:0]	RW	Gamma table = Entry 12 of set 1	0x0

46.6.13.30 GAMMA_TABLE1_13_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0874, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_13	[31:0]	RW	Gamma table = Entry 13 of set 1	0x0

46.6.13.31 GAMMA_TABLE1_14_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0878, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_14	[31:0]	RW	Gamma table = Entry 14 of set 1	0x0

46.6.13.32 GAMMA_TABLE1_15_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x087C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaTable1_15	[31:0]	RW	Gamma table = Entry 15 of set 1	0x0

46.6.13.33 GAMMA_REF_COLOR_REG

- Base Address: 0x1080_0000
- Address = Base Address + 0x0880, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
GammaRefColor	[31:0]	RW	Gamma table reference color	0x0

You can use the reference color of gamma table to decide the values of gamma table. You can use this reference color for 4-bit masking (WinCE Antialiased Font).

The channel order of reference color is ARGB and each channel consist 8-bit data. The 8-bit data follows the destination color format.

For example, when destination color format is 565, then the reference colors have these values:

- Reference color Alpha channel is 0xFF
- Reference color Red channel has 5-bit red value
- Reference color Blue channel has 6bit Blue value
- Reference color Green channel has 5-bit green value.

47 3D Graphic Accelerator (G3D)

47.1 Overview of G3D

The G3D block is based on Mali-400 MP core that belongs to ARM®family. The ARM®Mali™ family of products combines to provide the complete graphics stack for all embedded graphics needs. This enables device manufacturers and content developers to deliver highest quality and advanced graphics solutions across a broad range of consumer devices.

The Mali technology addresses a wide range of performance points, starting from the smallest GPU to full multi-core scalability up to 1080p resolutions. Pre-integrated and tested Mali graphics software supports this technology.

47.2 Features of G3D

The features of G3D are:

- 3D graphics, vector graphics based on programmable processors
- Tile based pixel processing
- Scalable multi-core pixel processors
  - 128 hardware multi-threads efficiently hiding the memory latency
- Advanced shader feature set and industry stand API support
- OGL-ES 1.1 and 2.0, Open VG 1.1
- Full featured Memory Management Unit (MMU) for all processor cores
  - Fully virtualized memory addressing for OS operation in a unified memory architecture
- Programmable power management
- On Chip tile of 24-bit fixed point depth buffer, or 32-bit floating point depth buffer
- On Chip tile of 8-bit stencil buffer
- Four-level hierarchical Z-buffer and stencil operations
- Fast dynamic branching
- Texture support
  - Cube map texture, projected texture, non-square texture
- Texture format
  - RGB8888, RGB565, RGB1555
  - Packed 12-bit, 24-bit, 32-bit YCC
  - 64-bit ARGB texture
  - ETC1 compressed texture
  - Depth-Stencil Texture, 32-bit: 24-bit Z, 8-bit stencil, and so on.
- Resolution of 4k × 4kframe buffer and 4k × 4k texture
- Bi-linear and tri-linear texture filtering
- Anti-aliasing: penalty-free 4x multi sampling, up to 512x FSAA. limited to 16x FSAA by driver.
- Indexed and non-indexed geometry input
- Configurable L2 level cache of 128 KB
- 4-way set-associative cache
- Up to 32 outstanding AXI transactions
- 8 to 64 bytes of uncached read bursts and write bursts

47.3 Architecture Brief of G3D

The architecture of G3D includes:

- Hybrid Architecture
  The hybrid architecture allows immediate mode rendering onto on-chip tile buffers.
  - Supports low power and cost by reducing memory bandwidth and processing complexity.
- On-chip Z-buffer, stencil and color buffers
  - Supports lower required memory bandwidth and footprint without affecting the functionality.
  - Supports single buffered rendering with the usage of Z-buffer and Stencil-buffers.
- Independent geometry and pixel processor operation
  - Less dependency on ARM CPU
  - Less IRQ overhead
- Per-frame autonomous bus-masters
  - AMBA AXI bus masters
  - Easy software set-up and operation
  - Burst optimized memory accesses
- Integrated MMU
  - Efficient and Flexible Mali memory management
  - Less constraints on how it maps Mali memory
- 16/32-bit frame buffer with many pixel color format options
- MSAA (4xAA) or SSAA (16xAA)
  - The architecture natively supports 4x with no performance hit
  - It can enable 4x per triangle, per texture
  - It can enable 16x on a per frame basis
- Maximum resolution
  - Supports a maximum resolution of 4096 × 4096
- 16 × 16 pixel tiles
  - Research shows 16 × 16 pixel tiles are optimal for mobile applications. The 32 × 32 pixel tiles may be beneficial for desktop-type applications.

47.4 G3D Structure

G3D structure section describes the structure of the GPU.

The graphics system includes:

- The pixel processor. It uses a list of primitives that geometry processor generates to produce a final image that is displayed on the screen. Add an additional three pixel processors to increase the rendering performance of the system.
- A programmable geometry processor that generates lists of primitives for a pixel processor to draw.
- A full-featured Memory Management Unit (MMU) for every processor. All memory accesses from the Pixel processor and Geometry processor use MMU for access checking and translation.
- A Level 2 cache controller
- An AMBA AXI Interconnect that can target high performance for high clock frequency system designs.A Power Management Unit (PMU) with an APB interface. The PMU powers down the Level 2 cache controller, Geometry Processor (GP), and the Pixel Processors (PP) individually. You can program the PMU to assert its interrupt line when power is stable for all requested devices.

Figure 40-1 illustrates a graphics system.

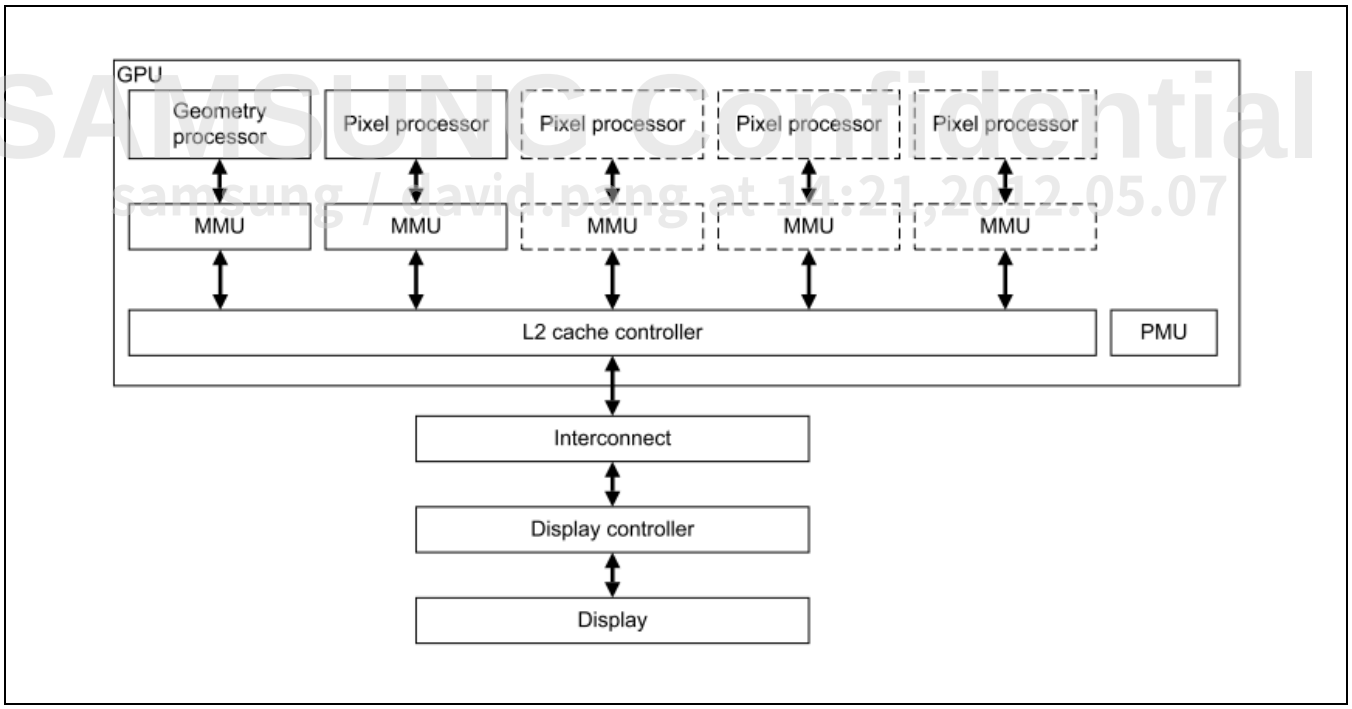


Figure 47-1 Graphics System

47.5 GPU Hardware Architecture

This section describes the GPU hardware architecture.

It includes:

- Top-level system
- GPU hardware architecture
- Level 2 cache controller hardware architecture

47.5.1 Top-level System

Figure 47-2 illustrates the Mali-400 MP GPU top-level system with interconnecting bus and interfaces.

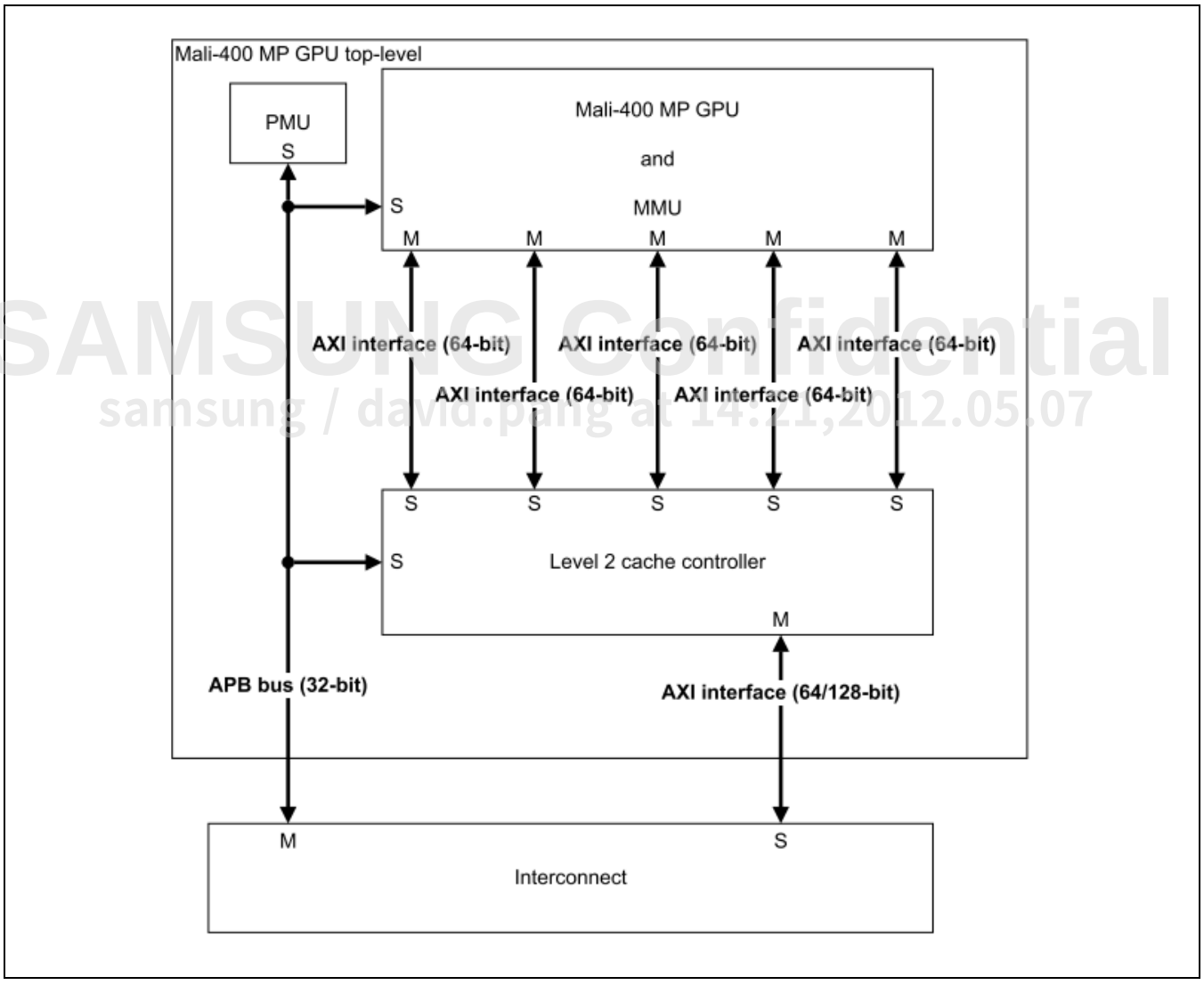


Figure 47-2 Mali-400 MP GPU Top-level System

47.5.2 Functional Block Diagram of GPU Hardware Architecture

Figure 47-3 illustrates the main functional blocks of the GPU hardware architecture.

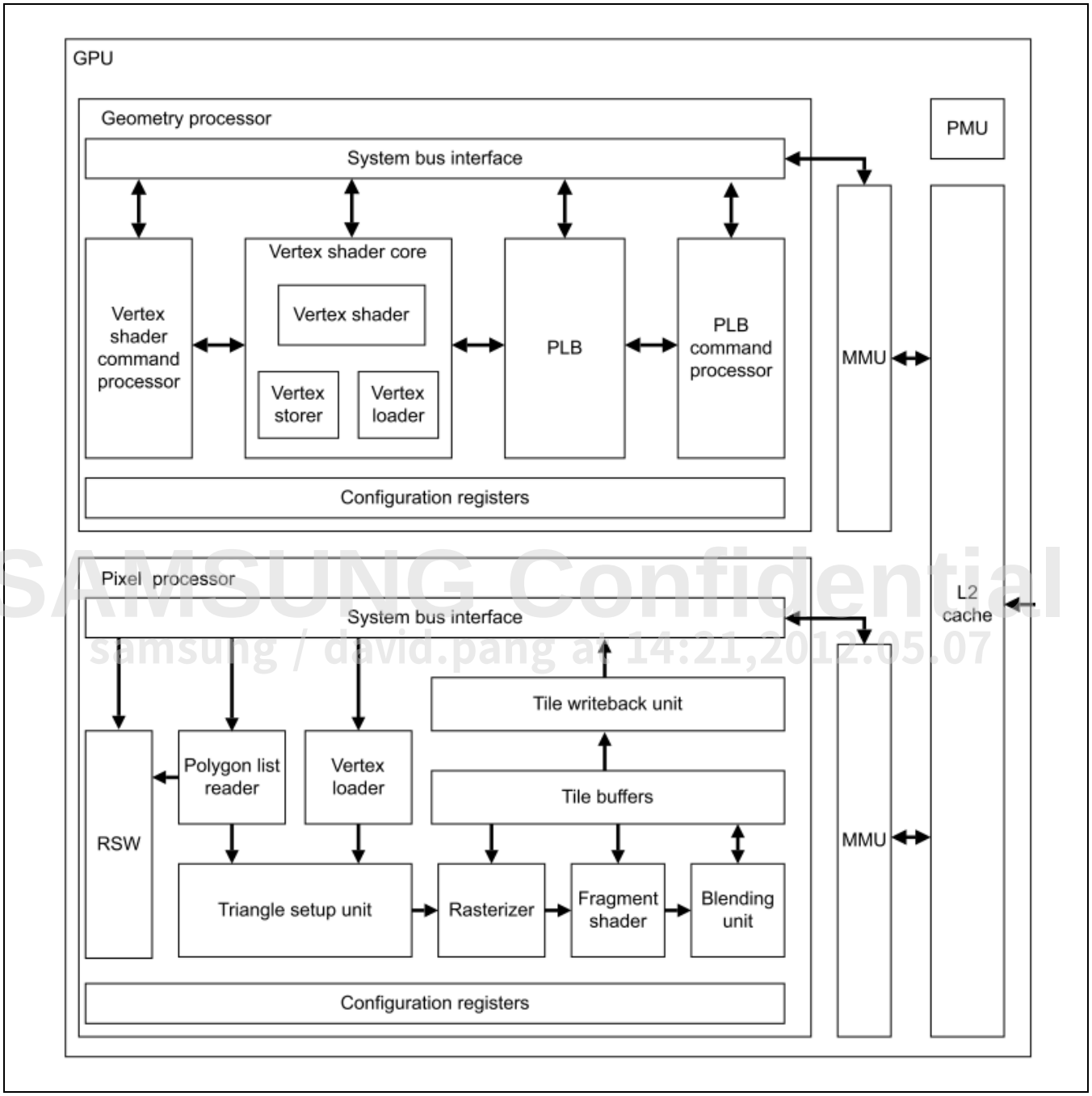


Figure 47-3 Mali-400 MP GPU Architecture

The Geometry processor consists of:

- A vertex shader command processor that reads and executes commands from a command list stored in memory.
- A vertex shader core that loads data for processing, performs the required calculations for each vertex, stores data from output registers in memory, and then exports data to integer or floating point numbers of different sizes.
- A polygon list builder unit that creates lists of polygons that the pixel processor should draw.
- The Polygon List Builder (PLB) command processor reads and executes the commands from the command list stored in memory.

The Pixel processor consists of:

- A polygon list reader reads the polygon lists from main memory and executes commands from the lists.
- The Render State Word (RSW) component is a data structure in main memory. This data structure contains the render state of polygons. The different pipeline stages in the renderer references each RSW that determines how to process the primitives.
- The vertex loader fetches the required vertices from memory for each primitive in the polygon list.
- The triangle setup unit takes data from the vertex loader and polygon list reader. Then it uses vertex data to compute coefficients for edge equations and varying interpolation equations.
- The rasterizer takes coefficients of appropriate equations from the triangle setup unit and uses them to divide polygons into fragments.
- The fragment shader is a programmable unit that calculates how each fragment of a primitive looks.
- The blending unit blends the calculated fragment value into the current frame buffer value in that position.
- The tile buffers take inputs from the fragment shader. The buffers perform various tests on the fragments, for example, Z-tests and stencil tests. When the tile is fully rendered, it is written to the frame buffer.
- The writeback unit writes the content of the tile buffer to system memory after the tile has been completely rendered.

47.5.3 PMU Hardware Architecture

The Mali PMU is an APB controlled power management unit. The Mali-400 MP GPU consists of the geometry processor, the Level 2 cache controller, and one to four pixel processors. These separate items can be referred to collectively as devices. The PMU is capable of individually powering up and powering down each device separately using the APB interface. The PMU has three to six Power Test Reset Controllers (PTRCs). The number of PTRCs depends on the number of devices. The PMU also contains a PTRC APB slave.

Figure 47-4 illustrates PMU with four devices.

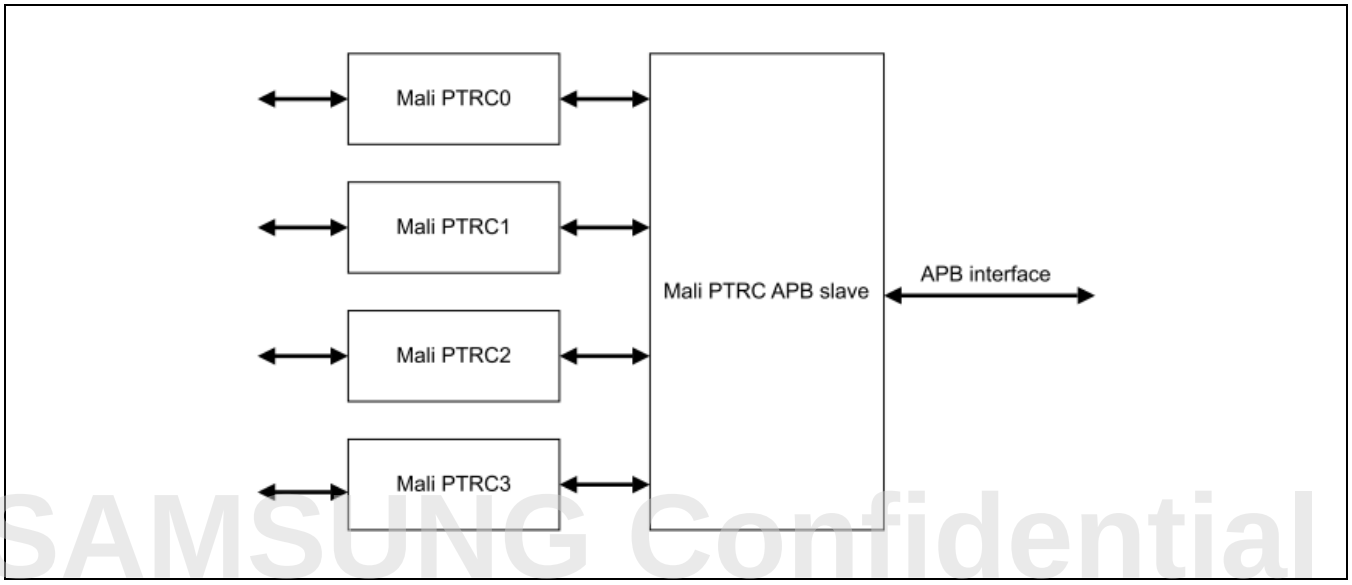


Figure 47-4 PMU with Four Devices

47.5.4 Level 2 Cache Controller Hardware Architecture

Figure 47-5 illustrates the hardware architecture of Level 2 cache controller.

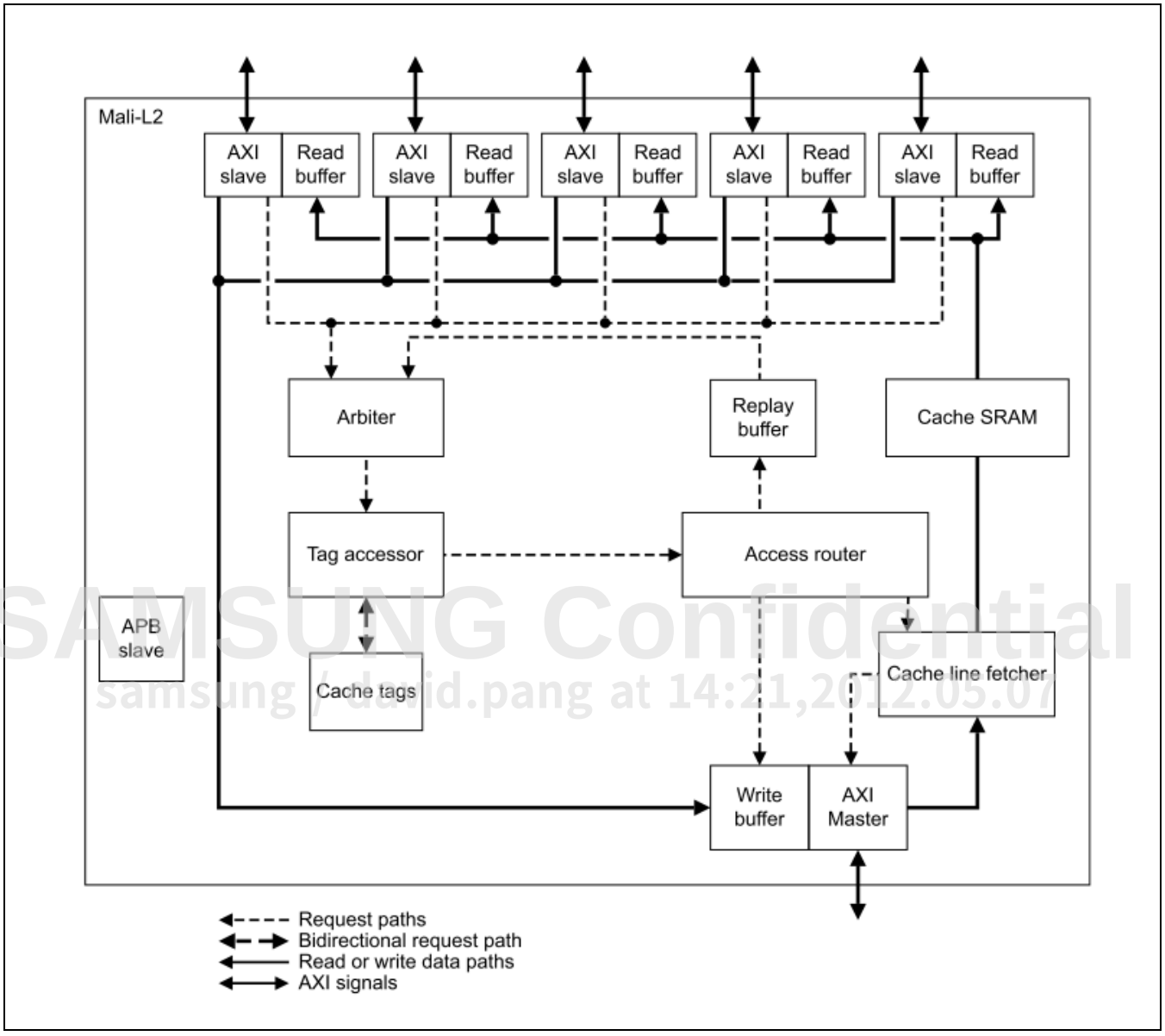


Figure 47-5 Level 2 Cache Controller Hardware Architecture

The Level 2 cache controller consists of:

- An APB slave that provides an interface to enable bus masters to control the Level 2 cache.
- An Arbiter that accepts memory access requests into a circulating loop. They then keep circulating in the loop until the Access router determines that they can be taken out of the loop.
- A Tag accessor that performs a cache lookup, to check if data is in the cache.
- An Access router for each Read or Write request that matches the AXI ID and the timestamp of the current request against all other requests in the loop.
- A Replay buffer that handles all request collisions of data.
- The Cache tag unit that holds a pipelined SRAM for the cache tags.
- The Cache line fetcher that draws the external data from the AXI master interface.
- The Cache SRAM that is the actual data store of the cache.

48 Image Rotator

48.1 Overview of Image Rotator

Image Rotator chapter describes image rotator that performs rotating and flipping image data.

It includes:

- Rotate FSM
- Rotate Buffer
- AMBA 3.0 AXI master and APB slave interface
- Register files

Describes overall features in the further section.

48.2 Features of Image Rotator

The features of image rotator are:

- Image format = YCbCr 4:2:2 (interleave), YCbCr 4:2:0 (non-interleave, 2-plane and 3-plane), RGB565 and RGB888 (unpacked)
- Rotate degree = 0, 90, 180, and 270 with flip vertical and flip horizontal
- Windows offset function
- Image size = Up to 32 K by 32 K (the maximum sizes are different from the types of image format)
- Image size restriction = Memory size should not exceed 16-bit address size. For example, RGB888 has a size limitation up to 14-bit image size, only[13:0] is valid and ignores[15:14].

48.3 Block Diagram

Figure 48-1 illustrates the block diagram of image rotator.

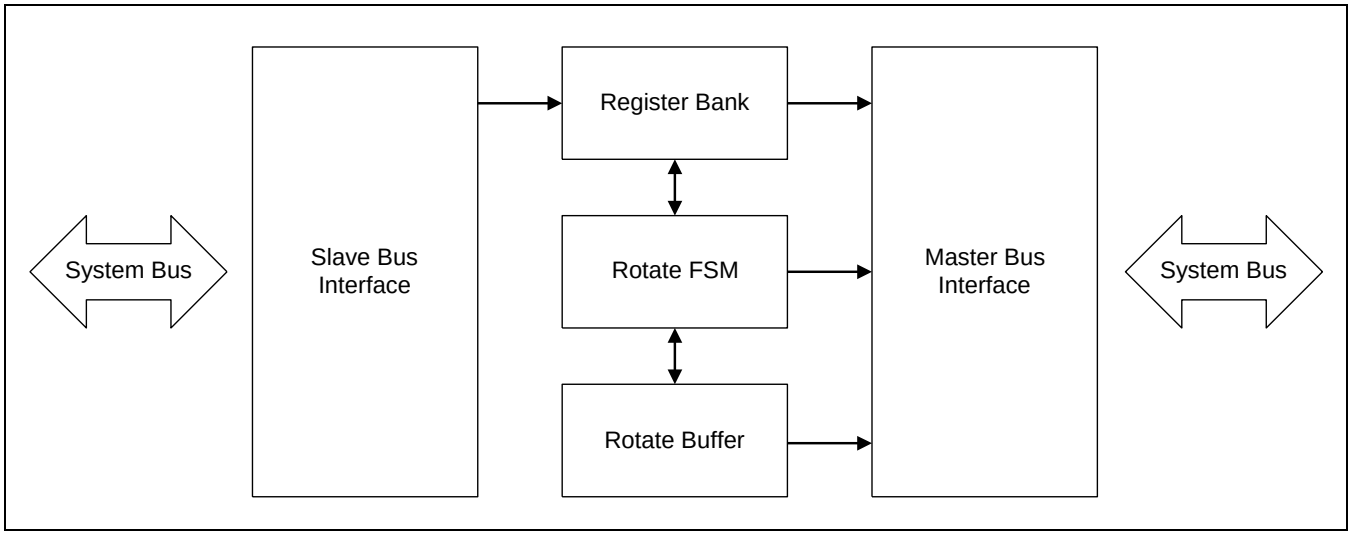


Figure 48-1 Image Rotator Block Diagram

48.4 Supported Image Rotation Functions

Figure 48-2 illustrates the rotation functions supported by image rotator.

Figure 48-2 Ported Image Rotation Functions

48.5 Image Rotation with Windows Offset

Image rotator supports image rotation with window offset function. This function is useful to move smaller portion of a large image into a portion of a large image. *Figure 48-3* illustrates the source image rotation example (with window offset function).

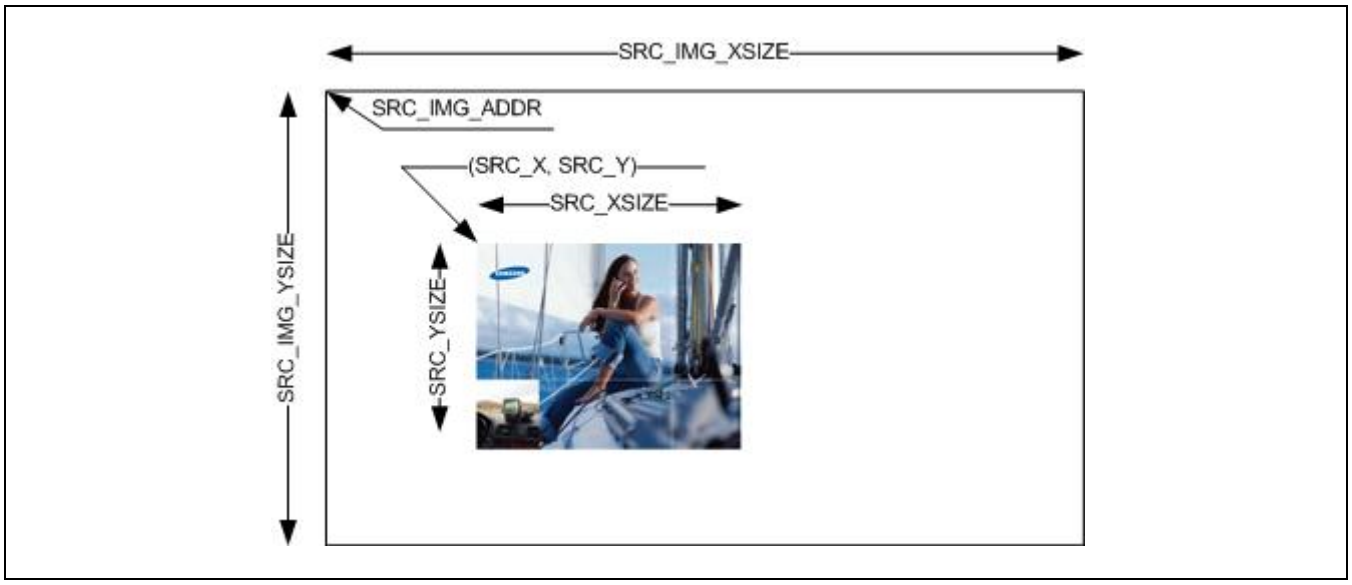


Figure 48-3 Source Image Example (With Window Offset Function)

Figure 48-4 illustrates the destination image example (90 degree rotated with window offset function).

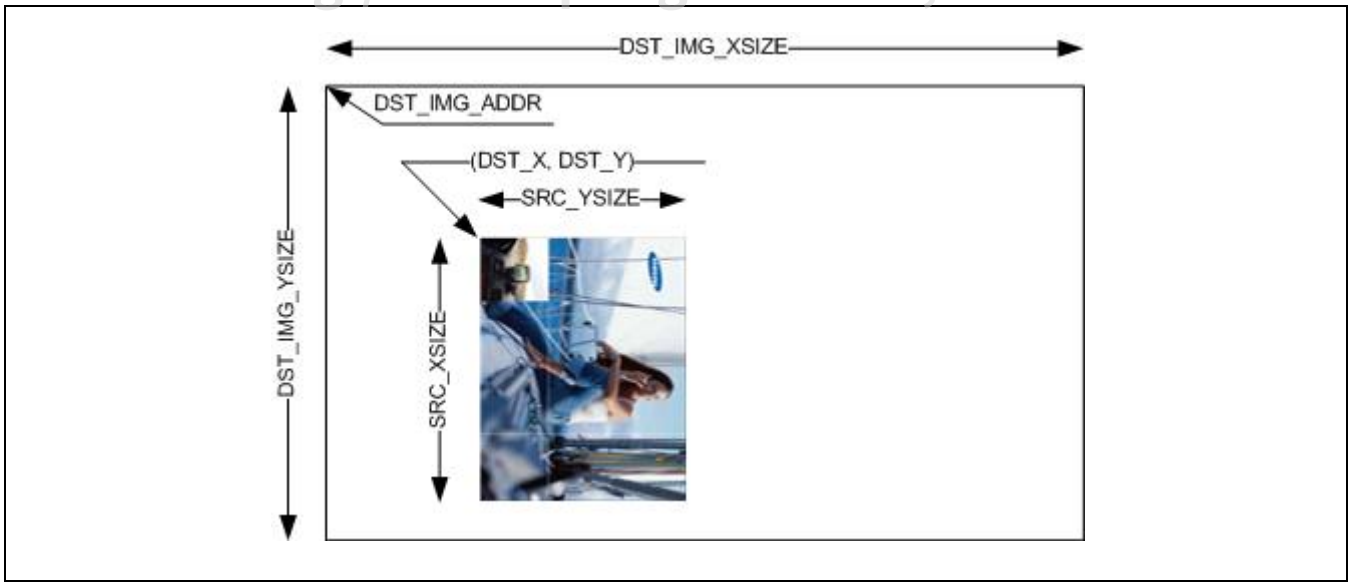


Figure 48-4 Destination Image Example (90 Degree Rotated With Window Offset Function)

48.6 Programming Guide

Programming Guide section includes:

- Resister Setting
- Restrictions on the Image Size

48.6.1 Resister Setting

You should set CONTROL[0] bit to start image rotator and all registers with CONTROL register except CONTROL[0] should previously set for proper operation.

48.6.2 Restrictions on the Image Size

Image rotator has some restrictions on image size. You should not violate these restrictions. The restrictions are:

- Image base address:
  The bit[2:0] should be zero for the SRCADDRREG0/1/2 and DSTADDRREG0/1/2 registers.
- Image size: You should set SRCIMGSIZE and DSTIMGSIZE image size as in the table here,

Image Format	Minimum Size	Maximum Size
RGB888	8 × 8	8 K × 8 K
RGB565	16 × 16	16 K × 16 K
YCbCr422	16 × 16	16 K × 16 K
YCbCr420 2-Plane	32 × 32	32 K × 32 K (in case of Y components)
YCbCr420 3-Plane	64 × 32	32 K × 32 K (in case of Y components)

- Rotate the image coordinates:
  You should set SRC_XY and DST_XY co-ordinates as in the table here,

Image Formats	Image Size Restrictions
RGB888	X and Y pixel size should be multiple of 4.
RGB565	X and Y pixel size should be multiple of 4.
YCbCr422	X and Y pixel size should be multiple of 4.
YCbCr420 2-Plane	X and Y pixel size should be multiple of 8.
YCbCr420 3-Plane	X and Y pixel size should be multiple of 16.

48.7 Register Description

48.7.1 Register Map Summary

- Base Address: 0x1281_0000

Register	Offset	Description	Reset Value
CONFIG	0x0000	Rotator configuration	0x0000_0000
CONTROL	0x0010	Rotator image0 control	0x0000_0000
STATUS	0x0020	Rotator status	0x0000_0000
SRCBASEADDR 0	0x0030	Rotator source image base address 0	0x0000_0000
SRCBASEADDR 1	0x0034	Rotator source image base address 1	0x0000_0000
SRCBASEADDR 2	0x0038	Rotator source image base address 2	0x0000_0000
SRCIMGSIZE	0x003C	Rotator source image X, Y size	0x0000_0000
SRC_XY	0x0040	Rotator source image X, Y coordinates	0x0000_0000
SRCROTSIZE	0x0044	Rotator source image rotation Size	0x0000_0000
DSTBASEADDR 0	0x0050	Rotator destination image base address 0	0x0000_0000
DSTBASEADDR 1	0x0054	Rotator destination image base address 1	0x0000_0000
DSTBASEADDR 2	0x0058	Rotator destination image base address 2	0x0000_0000
DSTIMGSIZE	0x005C	Rotator destination image X, Y size	0x0000_0000
DST_XY	0x0060	Rotator destination image X, Y coordinates	0x0000_0000

48.7.1.1 CONFIG

- Base Address: 0x1281_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	000_0000b
Enable Interrupt	[9]	RW	Interrupt enable indicates that configurations are set illegally. 0 = Disables interrupt 1 = Enables interrupt NOTE: In this case, rotator does not work.	1b
Enable Interrupt	[8]	RW	Interrupt enable indicates that an image rotation is complete. 0 = Disables interrupt 1 = Enables interrupt	0b
RSVD	[7:0]	–	Reserved	0x00

48.7.1.2 CONTROL

- Base Address: 0x1281_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved	0x0000
Pattern Writing	[16]	RW	Specifies to designate the pattern to write pattern to fill a destination image. 0 = Disables pattern writing 1 = Enables pattern writing NOTE: If this bit is set, then it ignores the information of a source image.	0b
RSVD	[15:11]	–	Reserved	00000b
Input Image Format	[10:8]	RW	Specifies to rotate the input image format. 000 = YCbCr 4:2:0 (3-plane) 001 = YCbCr 4:2:0 (2-plane) 010 = Reserved 011 = YCbCr 4:2:2 (interleave) 100 = RGB565 110 = RGB888 (Unpacked) 111 = Reserved	000b
Flip Direction	[7:6]	RW	Flip direction 0x = No flip 10 = Flips vertical 11 = Flips horizontal	00b
Rotation Degree	[5:4]	RW	Rotation degree 00 = 0 degree 01 = 90 degree 10 = 180 degree 11 = 270 degree	00b
RSVD	[3:1]	–	Reserved	000b
Start Rotate	[0]	RW	Rotate enable signal. Rotator starts the operation when this bit is set. Clears this bit if rotator starts to move an image. 1 = Starts rotate operation	0b

48.7.1.3 STATAUS

- Base Address: 0x1281_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	0x00
Interrupt Pending	[9]	R	This bit is set if the SFR is set illegally. Write of 1 clears this bit.	1b
Interrupt Pending	[8]	R	This bit is set if an image rotation is complete. Write of 1 clears this bit.	0b
RSVD	[3:1]	–	Reserved	0x00
Rotator status	[1:0]	R	These bits show the rotator operation status. 00 = IDLE status 01 = Reserved 10 = Rotating an image (BUSY) 11 = Rotating an image, and has one more job to rotate (BUSY)	00b

48.7.1.4 SRCBASEADDR 0

- Base Address: 0x1281_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SRC_IMG_ ADDR	[31:0]	RW	Base address of source image for RGB or Y component	0x0000_0000

48.7.1.5 SRCBASEADDR 1

- Base Address: 0x1281_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SRC_IMG_ ADDR	[31:0]	RW	Base Address of source image for Cb component.	0x0000_0000

48.7.1.6 SRCBASEADDR 2

- Base Address: 0x1281_0000
- Address = Base Address + 0x0038, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SRC_IMG_ ADDR	[31:0]	RW	Base Address of source image for Cr component.	0x0000_0000

48.7.1.7 SRCIMGSIZE

- Base Address: 0x1281_0000
- Address = Base Address + 0x003C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SRC_YSIZE	[31:16]	RW	Vertical pixel size of a source image	0x0000
SRC_XSIZE	[15:0]	RW	Horizontal pixel size of a source image	0x0000

48.7.1.8 SRC_XY

- Base Address: 0x1281_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SRC_Y	[31:16]	RW	Specifies to rotate the pixel coordinates on Y-axis of an image.	0x0000
SRC_X	[15:0]	RW	Specifies to rotate the pixel coordinates on X-axis of an image.	0x0000

48.7.1.9 SRCROTSIZE

- Base Address: 0x1281_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SRC_YSIZE	[31:16]	RW	Specifies to rotate the vertical pixel size of an image.	0x0000
SRC_XSIZE	[15:0]	RW	Specifies to rotate the horizontal pixel size of an image	0x0000

48.7.1.10 DSTBASEADDR 0

- Base Address: 0x1281_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DST_IMG_ ADDR	[31:0]	RW	Address of destination image for RGB or Y component.	0x0000_0000

48.7.1.11 DSTBASEADDR 1

- Base Address: 0x1281_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DST_IMG_ ADDR	[31:0]	RW	Address of destination image for CB component.	0x0000_0000

48.7.1.12 DSTBASEADDR 2

- Base Address: 0x1281_0000
- Address = Base Address + 0x0058, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DST_IMG_ ADDR	[31:0]	RW	Address of destination image for Cr component.	0x0000_0000

48.7.1.13 DSTIMGSIZE

- Base Address: 0x1281_0000
- Address = Base Address + 0x005C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DST_YSIZE	[31:16]	RW	Vertical pixel size of a target image	0x0000
DST_XSIZE	[15:0]	RW	Horizontal pixel size of a target image	0x0000

48.7.1.14 DST_XY

- Base Address: 0x1281_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DST_Y	[31:16]	RW	The pixel coordinates on Y-axis of a destination image	0x0000
DST_X	[15:0]	RW	The pixel coordinates on X-axis of a destination image	0x0000

49 JPEG Codec

49.1 Overview

This chapter describes the usage of JPEG codec.

The JPEG compression standard for still image is a general-purpose photographic image compression algorithm. JPEG accommodates:

- A wide variety of pixel resolution
- Color spaces
- Transmission bandwidth

JPEG emphasizes the importance of the compression capability of standard algorithm and its efficiency of implementation for both software and hardware.

Figure 49-1 illustrates the codec structure. This figure also illustrates the different processing blocks involved in JPEG.

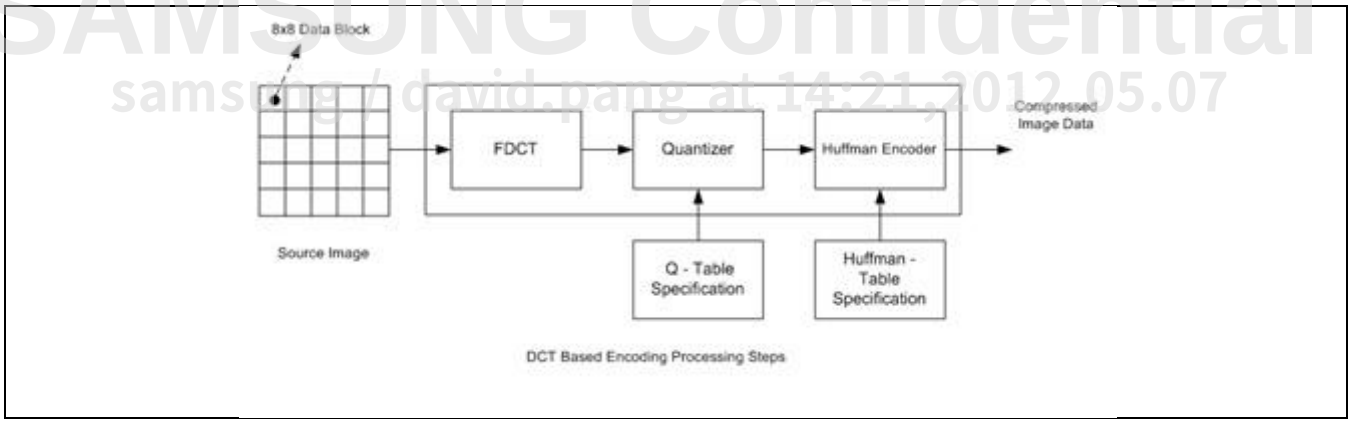


Figure 49-1 Codec Structure

49.2 Features

Features of JPEG are:

- Supports 8-bit grayscale or RGB-888, RGB-565 Color components or YCbCr 4:4:4, YCbCr 4:2:2, and YCbCr 4:2:0 color component input for encoding.
- Input pixel range/precision is 8-bit.
- Supports baseline encoding (Sequential DCT-based, lossy mode, interchange format syntax)
- Supports image size up to 8192 × 8192
- Supports transformation from RGB to YCbCr during encoding.
- Performs image padding (completion of incomplete image)

The features of JPEG compression are:

- Compresses either full color or gray-scale images.
- Handles only still images.
- Lossy compression scheme.
- Provides 10:1 to 20:1 compression of full color data without visible loss.

49.3 Color Component Data Ordering

Table 49-1 lists the ordering of color component data in respective memory plane with respect to different color formats.

Table 49-1 Color Component Data Ordering

Color Format	Data Ordering
YCbCr 4:4:4 2PLANE	Plane0 = Y Component Plane1 = {Cr, Cb} <= {MSB, LSB}
YCbCr 4:2:2 2PLANE	Plane0 = Y Component Plane1 = {Cr, Cb} <= {MSB, LSB}
YCbCr 4:2:2 1PLANE	Plane0 = {Cr, Y1, Cb, Y0} <= {MSB, LSB}
YCbCr 4:2:0 2PLANE	Plane0 = Y Component Plane1 = {Cr, Cb} <= {MSB, LSB}
RGB 5:6:6 1PLANE	Plane0 = {R[4 = 0] = G[5 = 0] = B[4 = 0]} <= {MSB, LSB}
RGB 8:8:8 1PLANE	Plane0 = {8'b0 = R[7 = 0] = G[7 = 0] = B[7 = 0]} <= {MSB, LSB}
YCbCr 4:4:4 3PLANE	Plane0 = Y Component Plane1 = Cb Component Plane2 = Cr Component
YCbCr 4:2:2 3PLANE	Plane0 = Y Component Plane1 = Cb Component Plane2 = Cr Component
YCbCr 4:2:0 3PLANE	Plane0 = Y Component Plane1 = Cb Component Plane2 = Cr Component
8-bit Gray	Plane0 = Y Component

49.4 Register Description

49.4.1 Register Map Summary

- Base Address: 0x1184_0000

Register	Offset	Description	Reset Value
JPEG_CNTL	0x0000	JPEG Control Register	0x3000_0000
INT_EN	0x0004	Interrupt Enable Register	0x0000_001F
RSVD	0x0008	Reserved	Undefined
INT_STATUS	0x000C	Interrupt Status Register	0x0000_0000
IMAGE_OP_MEM_BA	0x0010	Output Image Base Address	0x0000_0000
JPEG_IMAGE_SIZE	0x0014	JPEG Image Size	0x0000_0000
IMAGE_BA_Plane1	0x0018	Input Image Base Address for Plane1	0x0000_0000
IMAGE_SO_Plane1	0x001C	Input Image Scanline Offset for Plane1	0x0000_0000
IMAGE_PO_Plane1	0x0020	Input Image Picture Offset for Plane1	0x0000_0000
IMAGE_BA_Plane2	0x0024	Input Image Base Address for Plane2	0x0000_0000
IMAGE_SO_Plane2	0x0028	Input Image Scanline Offset for Plane2	0x0000_0000
IMAGE_PO_Plane2	0x002C	Input Image Picture Offset for Plane2	0x0000_0000
IMAGE_BA_Plane3	0x0030	Input Image Base Address for Plane3	0x0000_0000
IMAGE_SO_Plane3	0x0034	Input Image Scanline Offset for Plane3	0x0000_0000
IMAGE_PO_Plane3	0x0038	Input Image Picture Offset for Plane3	0x0000_0000
TBL_SEL	0x003C	Table select for quantization and Huffman table	0x0000_0000
IMAGE_FORMAT	0x0040	Refer the required bit information in the section, *49.4.1.16 IMAGE_FORMAT* register description, for more information.	0x0000_0000
BITSTREAM_SIZE	0x0044	Input JPEG Stream Size (in number of 32 Bytes)	0x1FFF_FFFF
PADDING	0x0048	Padding Register	0x0000_0000
HUFF_CNT	0x004C	Huffman Info Count Register	0x0000_0000
FIFO_STATUS (1)	0x0050	Y0, Y1, Cb, Cr, OP FIFO Status	0x0000_0067
RSVD	0x0054 to 0x0058	Reserved	0x0000_0000
QUAN_TBL_ENT (2)	0x0100 to 0x01FF	Single entry point to store the quantization table information	0x0000_0000
HUFF_TBL_ENT (2)	0x0200 to 0x03BF	Single entry point to store the Huffman table information	0x0000_0000
RSVD	0x005C to 0x00FF	Reserved	Undefined

NOTE: This table summarizes the Special Function Registers. JPEG codec uses these registers

1. After RESET Release, the value of FIFO_STATUS Register is 0x0000_0000 until it configures IP. Default values of FIFO Empty Status signals[0x0000_0067] is valid only after configuring IP.
2. Write Only Register: User/Host can perform only Write operation to the Register.

49.4.1.1 JPEG_CNTL

- Base Address: 0x1184_0000
- Address = Base Address + 0x0000, Reset Value = 0x3000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	R	Reserved	2'd0
Soft reset	[29]	RW	Soft reset 1'b0 = Reset pulled low (It asserts soft reset to all It asserts soft reset to all registers of JPEG and APB domain except soft reset bit (itself) 1'b1 = Releases reset (Releases soft reset)	1'b1
SYS_INT_EN	[28]	RW	System Interrupt enable 1'b0 = Disables interrupt 1'b1 = Enables interrupt	1'b1
Image_padding	[27]	RW	Use internal padding 1'b0 = Does not use padding register 1'b1 = Uses padding register for padding the column	1'b0
RSVD	[26:20]	R	Reserved	7'd0
Update_Huf_Tbl (NOTE)	[19]	RW	Update Huffman table 1'b0 = Does not generate Huffman tables internally 1'b1 = Generates the Huffman tables by using the information that is programmed in SFR "HUFF_TBL_ENT"	1'b0
Rest_int	[18:3]	RW	Restart interval	16'd0
Aut_rst_marker	[2]	RW	Adds the restart marker automatically 1'b0 = Does not add restart marker 1'b1 = Adds restart marker automatically	1'b0
JPEG_CODEC_ON	[1]	RW	Codec Disable/Enable 1'b0 = Disable encoder 1'b1 = Enable encoder	1'b0
RSVD	[0]	R	Reserved. Should be 0.	1'b0

NOTE: User/host must program this bit as "1" for every encoder operation. It should program SFR "HUFF_TBL_ENT" and SFR "UFF_CNT" accordingly for every encoder operation.

49.4.1.2 INT_EN

- Base Address: 0x1184_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_001F

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	R	Reserved	30'h0000_0007
Image_Completion_en	[1]	RW	Image completion interrupt enable 1'b0 = Disables interrupt 1'b1 = Enables interrupt	1'b1
RSVD	[0]	R	Reserved	1'b1

49.4.1.3 INT_STATUS

- Base Address: 0x1184_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	R	Reserved	30'd0
Image_Completion	[1]	COR	Image completion 1'b0 = Does not complete image encode 1'b1 = Completes image encode	1'b0
RSVD	[0]	R	Reserved	1'b0

NOTE:

1. Upon receiving the interrupt, on reading, the INT_STATUS register, and if any one of the error flags is set to HIGH, the user/host must issue a SOFT RESET post servicing the interrupt. Refer to Section *49.4.1.1 JPEG_CNTL* register for more information on SOFT Reset.
2. When any one of the error flags is set to HIGH, it updates the INT_STATUS accordingly and the value of INT_STATUS is retained until user/host services the interrupt.

49.4.1.4 IMAGE_OP_MEM_BA

- Base Address: 0x1184_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Image_op_mem_ba	[31:0]	RW	This is the destination address for the encoded stream	32'd0

49.4.1.5 JPEG_IMAGE_SIZE

- Base Address: 0x1184_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Image_y	[31:16]	RW	Number of lines (Y) in encoder	16'd0
Image_x	[15:0]	RW	Number of samples per line (X) in encoder	16'd0

NOTE:

1. If input image is 422, then image_x must be of EVEN length.
2. If input image is 420, then image_x and image_y must be of EVEN length.
3. Minimum value of the image_x and image_y should be 8 (1 Macro Block Boundary).

49.4.1.6 IMAGE_BA_Plane1

- Base Address: 0x1184_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Image_ba_plane1	[31:0]	RW	Base address for input image for plane1	32'd0

49.4.1.7 IMAGE_SO_Plane1

- Base Address: 0x1184_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Image_so_plane1	[31:0]	RW	Scanline offset for input image for plane1 (User programs this register)	32'd0

49.4.1.8 IMAGE_PO_Plane1

- Base Address: 0x1184_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Image_po_plane1	[31:0]	RW	Picture offset for input image for plane1	32'd0

49.4.1.9 IMAGE_BA_Plane2

- Base Address: 0x1184_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Image_ba_plane2	[31:0]	RW	Base address for input image for plane2	32'd0

49.4.1.10 IMAGE_SO_Plane2

- Base Address: 0x1184_0000
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Image_so_plane2	[31:0]	RW	Scanline offset for input image for plane2 (User programs this register)	32'd0

49.4.1.11 IMAGE_PO_Plane2

- Base Address: 0x1184_0000
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Image_po_plane2	[31:0]	RW	Picture offset for input image for plane2	32'd0

49.4.1.12 IMAGE_BA_Plane3

- Base Address: 0x1184_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Image_ba_plane3	[31:0]	RW	Base address for input image for plane3	32'd0

49.4.1.13 IMAGE_SO_Plane3

- Base Address: 0x1184_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Image_so_plane3	[31:0]	RW	Scanline offset for input image for plane3 (User programs this register)	32'd0

49.4.1.14 IMAGE_PO_Plane3

- Base Address: 0x1184_0000
- Address = Base Address + 0x0038, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Image_po_plane3	[31:0]	RW	Picture offset for input image for plane3	32'd0

49.4.1.15 TBL_SEL

- Base Address: 0x1184_0000
- Address = Base Address + 0x003C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	R	Reserved	20'd0
HUFF_TBL_comp3	[11:10]	RW	Huffman table selector for component3 (It places this value in the encoded bit-stream for this component) 2'b00 = 0 AC-0 DC-0 2'b01 = 1 AC-0 DC-1 2'b10 = 2 AC-1 DC-0 2'b11 = 3 AC-1 DC-1	2'd0
HUFF_TBL_comp2	[9:8]	RW	Huffman table selector for component2 (It places this value in the encoded bit-stream for this component) 2'b00 = 0 AC-0 DC-0 2'b01 = 1 AC-0 DC-1 2'b10 = 2 AC-1 DC-0 2'b11 = 3 AC-1 DC-1	2'd0
HUFF_TBL_comp1	[7:6]	RW	Huffman table selector for component1 (It places this value in the encoded bit-stream for this component) 2'b00 = 0 AC-0 DC-0 2'b01 = 1 C-0 DC-1 2'b10 = 2 AC-1 DC-0 2'b11 = 3 AC-1 DC-1	2'd0
Q_TBL_comp3	[5:4]	RW	Quantization table selector for component3 (It places this value in the encoded bit-stream for this component) 2'b00 = 0 2'b01 = 1 2'b10 = 2 2'b11 = 3	2'd0
Q_TBL_comp2	[3:2]	RW	Quantization table selector for component2 (It places this value in the encoded bit-stream for this component) 2'b00 = 0 2'b01 = 1 2'b10 = 2 2'b11 = 3	2'd0
Q_TBL_comp1	[1:0]	RW	Quantization table selector for component1 (It places this value in the encoded bit-stream for this component) 2'b00 = 0 2'b01 = 1 2'b10 = 2 2'b11 = 3	2'd0

49.4.1.16 IMAGE_FORMAT

- Base Address: 0x1184_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	R	Reserved	6'd0
Image_ENC	[25:24]	RW	Formats in which it encodes the image 2'b00 = Gray image 2'b01 = YCbCr 444 2'b10 = YCbCr 422 2'b11 = YCbCr 420	2'd0
RSVD	[23:18]	R	Reserved	6'd0
YCbCr_420_ Image_IP	[17:15]	RW	YCbCr 420 image definition in memory 3'b0xx = Non-YCbCr 420 image 3'b100 = YCbCr 420 image 2 plane 3'b101 = YCbCr 420 image 3 plane 3'b110, 3'b111 = Reserved	3'd0
YCbCr_422_ Image_IP	[14:12]	RW	YCbCr 422 image definition in memory 3'b0xx = Non-YCbCr 422 Image 3'b100 = YCbCr 422 image 1 plane 3'b101 = YCbCr 422 image 2 plane 3'b110 = YCbCr 422 image 3 plane 3'b111 = Reserved	3'd0
YCbCr_444_ Image_IP	[11:9]	RW	YCbCr 444 image definition in memory 3'b0xx = Non-YCbCr 444 image 3'b100 = YCbCr 444 image 2 plane 3'b101 = YCbCr 444 image 3 plane 3'b110, 3'b111 = Reserved	3'd0
RGB_Image_IP	[8:6]	RW	RGB image definition in memory 3'b0xx = Non-RGB image 3'b100 = 16-bit RGB image (565) 3'b101 = 32-bit RGB 3'b110, 3'b111 = Reserved	3'd0
Gray_Image_IP	[5:3]	RW	Gray level image definition in memory 3'b0xx = Non-gray Image 3'b100 = 8-bit gray IMAGE 3'b101, 3'b110, 3'b111 = Reserved	3'd0
Image_sel	[2:0]	RW	Selects the type of image stored in memory for encoder 3'b000 = Gray 3'b001 = RGB 3'b010 = YCbCr 444 3'b011 = YCbCr 422 3'b100 = YCbCr 420 3'b101, 3'b110, 3'b111 = Reserved	3'd0

49.4.1.17 BITSTREAM_SIZE

- Base Address: 0x1184_0000
- Address = Base Address + 0x0044, Reset Value = 0x1FFF_FFFF

Name	Bit	Type	Description	Reset Value
BITSTREAM_SIZE	[31:0]	RW	This register indicates the encoded bit-stream size in bytes. User/Host should read this register after they receive the interrupt to identify the compressed bit-stream size. The value of this register is always Even. NOTE: 1. When the encoded bit-stream size is N (N being odd value), this register indicates the value as N + 1. When the encoded bit-stream size is N (N being even value), this register indicates the value as N.	32'h1FFF_FFFF

NOTE: After encoding, this register indicates the encoded bit-stream size in bytes. User or Host can read this register after they receive the interrupt.

49.4.1.18 PADDING

- Base Address: 0x1184_0000
- Address = Base Address + 0x0048, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	R	Reserved	8'd0
PAD_CR	[23:16]	RW	Padding the Cr-sample	8'd0
PAD_CB	[15:8]	RW	Padding the Cb-sample	8'd0
PAD_Y	[7:0]	RW	Padding the Y-sample	8'd0

49.4.1.19 HUFF_CNT

- Base Address: 0x1184_0000
- Address = Base Address + 0x004C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	R	Reserved	16'd0
Huff_info_cnt	[15:0]	RW	Huffman table byte count, including the 2-bytes for count. (Huff_info_cnt shall be put after DHT marker in encoder mode.)	16'd0

49.4.1.20 FIFO_STATUS (NOTE)

- Base Address: 0x1184_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000_0067

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	R	Reserved	18'd0
i_fifo_full_y0	[13]	R	HIGH indicates Y0 FIFO full	1'b0
i_fifo_full_y1	[12]	R	HIGH indicates Y1 FIFO full	1'b0
RSVD	[11]	R	Reserved	1'b0
RSVD	[10]	R	Reserved	1'b0
i_fifo_full_cb	[9]	R	HIGH indicates Cb FIFO full	1'b0
i_fifo_full_cr	[8]	R	HIGH indicates Cr FIFO full	1'b0
i_fifo_full_op	[7]	R	HIGH indicates OP FIFO full	1'b0
i_y0_fifo_em_i	[6]	R	HIGH indicates Y0 FIFO empty	1'b1
i_y1_fifo_em_i	[5]	R	HIGH indicates Y1 FIFO empty	1'b1
RSVD	[4]	R	Reserved	1'b0
RSVD	[3]	R	Reserved	1'b0
i_cb_fifo_em_i	[2]	R	HIGH indicates Cb FIFO empty	1'b1
i_cr_fifo_em_i	[1]	R	HIGH indicates Cr FIFO empty	1'b1
i_rempty_op	[0]	R	HIGH indicates OP FIFO empty	1'b1

NOTE: After RESET Release, value of FIFO_STATUS Register is 0x0000_0000 until H/W is configured. Default values of FIFO Empty status signals[0x0000_0067] are valid only after configuring IP.

49.4.1.21 QUAN_TBL_ENT

- Base Address: 0x1184_0000
- Address = Base Address + 0x0100 to 0x01FF, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Quan_tbl_entry	[31:0]	WO	Quantization table entry for each component in encoder (This APB write cycle will be internally converted to memory write cycle.)	32'd0

49.4.1.22 HUFF_TBL_ENT

- Base Address: 0x1184_0000
- Address = Base Address + 0x200 to 0x3BF, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
Huff_tbl_entry	[31:0]	WO	Huffman table entry for each component in encoder (This APB write cycle will be internally converted to memory write cycle.)	32'd0

49.5 Programmer's Model

This section includes:

- Encoder Flow Chart

49.5.1 Encoder Flow Chart

illustrates the flow chart of JPEG-Encoder.

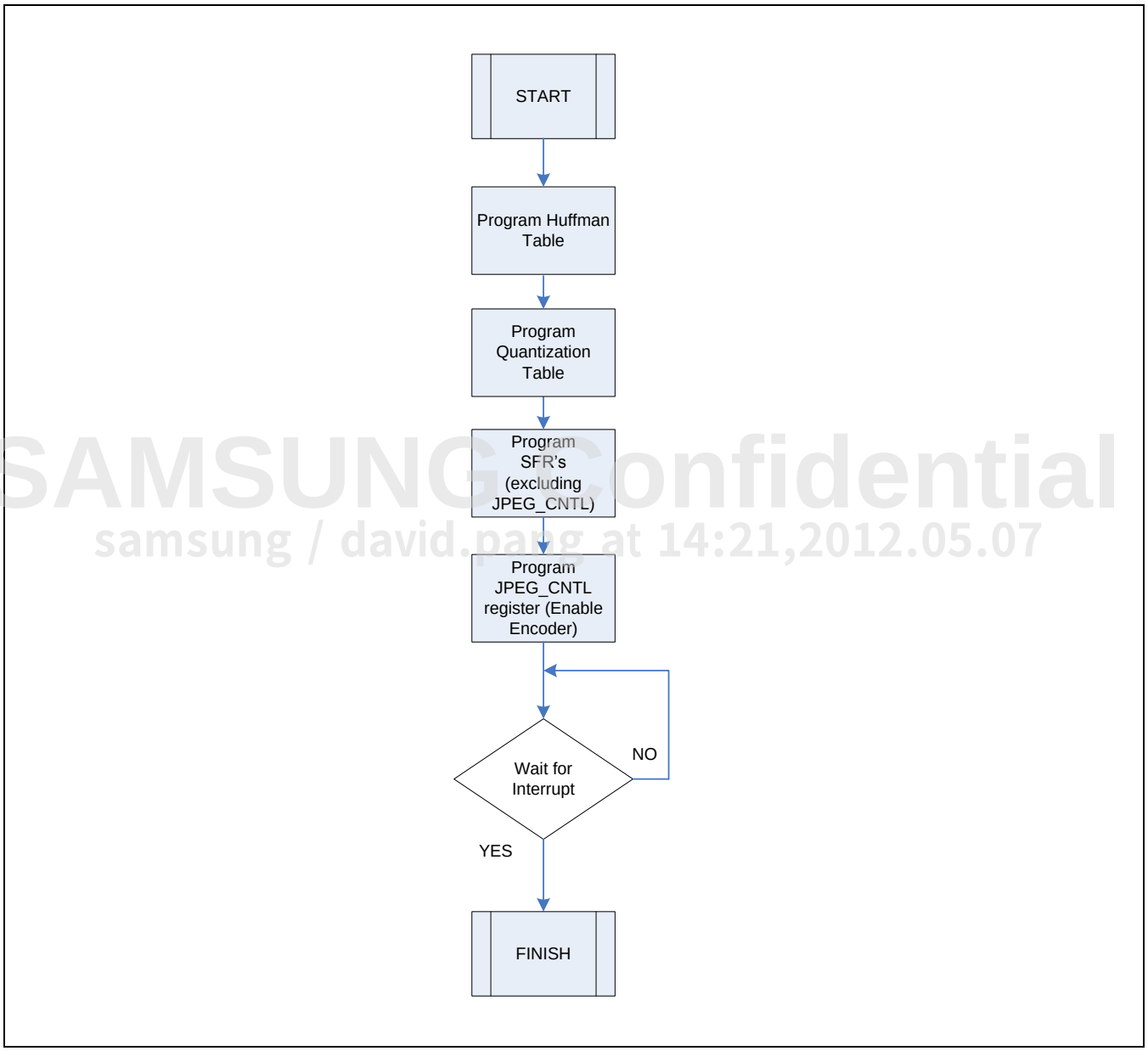


Figure 49-2 JPEG-Encoder Flowchart

49.5.2 Programming QUAN_TBL_ENT and HUFF_TBL_ENT

In encoder mode, user/host should program Quantization and Huffman table through the QUAN_TBL_ENT and HUFF_TBL_ENT SFR by using APB interface as listed in Table 1-2.

For example, when the user/host wants to program the Quantization table 0, then the user/host must generate 16 APB write cycles onto the register specified in the Register interface with addresses range from 0x100 to 0x13C.

Table 49-2 Quantization/Huffman Table

APB Address	Quantization/Huffman Value				Remarks
0x100	Q3	Q2	Q1	Q0	Quantization Table 0
0x104	Q7	Q6	Q5	Q4	
0x13C	Q63	Q62	Q61	Q60	
0x140	Q3	Q2	Q1	Q0	Quantization Table 1
0x144	Q7	Q6	Q5	Q4	
0x17C	Q63	Q62	Q61	Q60	
0x180	Q3	Q2	Q1	Q0	Quantization Table 2
0x184	Q7	Q6	Q5	Q4	
0x1BC	Q63	Q62	Q61	Q60	
0x1C0	Q3	Q2	Q1	Q0	Quantization Table 3
0x1C4	Q7	Q6	Q5	Q4	
0x1FC	Q63	Q62	Q61	Q60	
0x200	code_len_4	code_len_3	code_len_2	code_len_1	DC Luminance (Code length)
0x204	code_len_8	code_len_7	code_len_6	code_len_5	
0x20C	code_len_16	code_len_15	code_len_14	code_len_13	
0x210	Value_4	Value_3	Value_2	Value_1	DC Luminance (Values)
0x214	Value_8	Value_7	Value_6	Value_5	
0x218	Value_12	Value_11	Value_10	Value_9	
0x21C	–	–	–	–	
0x220	code_len_4	code_len_3	code_len_2	code_len_1	DC Chrominance (Code length)
0x224	code_len_8	code_len_7	code_len_6	code_len_5	
0x22C	code_len_16	code_len_15	code_len_14	code_len_13	
0x230	Value_4	Value_3	Value_2	Value_1	DC Chrominance (Values)
0x234	Value_8	Value_7	Value_6	Value_5	
0x238	Value_12	Value_11	Value_10	Value_9	
0x23C	–	–	–	–	
0x240	code_len_4	code_len_3	code_len_2	code_len_1	AC Luminance (Code length)
0x244	code_len_8	code_len_7	code_len_6	code_len_5	
0x24C	code_len_16	code_len_15	code_len_14	code_len_13	
0x250	Value_4	Value_3	Value_2	Value_1	AC Luminance

APB Address	Quantization/Huffman Value				Remarks
0x254	Value_8	Value_7	Value_6	Value_5	(Values)
0x2F0	–	–	Value_162	Value_161	
0x2F4 to 0x2FF	–	–	–	–	
0x300	code_len_4	code_len_3	code_len_2	code_len_1	AC Chrominance (Code length)
0x304	code_len_8	code_len_7	code_len_6	code_len_5	
0x30C	code_len_16	code_len_15	code_len_14	code_len_13	

49.6 Example Codes

This is an example code for encoder (444-3PLANE-422).

Initialization Routine: Encoder 444-3PLANE-422 (Image Size 64 × 64)

444 3PLANE 422 correspond to these conditions from user configuration perspective:

1. Format in which Raw Data is available in External Memory: 444
2. Number of planes in which Raw Data is available in External Memory: 3-PLANE
3. Format in which image needs to be encoded: 422

```
void  activate_fimp_jpeg_enc (void) {
                                                    //Program Huffman Table information
      apbif_multiple_write (HUFF_TBL_ENT, <random_data>);  //Program Quantization Table information
      apbif_multiple_write (QUANT_TBL_ENT, <random_data>); //Program SFR (except JPEG_CNTL)
      apbif_single_write (JPEG_IMAGE_SIZE, 0x0040_0040);
      apbif_single_write (IMAGE_FORMAT, 0x0200_0B02);
      apbif_single_write (IMAGE_BA_PLANE1, 0x0200_0000);   //Plane-1 raw sample address
      apbif_single_write (IMAGE_BA_PLANE2, 0x0300_0000);   //Plane-1 raw sample address
      apbif_single_write (IMAGE_BA_PLANE3, 0x0400_0000);   //Plane-1 raw sample address
      apbif_single_write (Image_OP_MEM_BA, 0x0500_0000);   //Encoded Bit-stream address
      apbif_single_write (INT_EN, 0x0000_0002);            //Program SFR (JPEG_CNTL)
      apbif_single_write (JPEG_CNTL, 0x3008_0002);         //[JPEG_CNTL register, enable encoder mode]
}
```


Interrupt Service Routine

```
Interrupt_service_routine () is {
apbif_single_read (INT_STATUS);
}
```

49.7 References

1. Information technology-digital compression and coding of continuous-tone still images: Requirements and guidelines [ISO/IEC 10918-1]

2. Information technology-digital compression and coding of continuous-tone still images: Compliance testing [ISO/IEC 10918-2]

3. AMBA 3 APB protocol specification [version 1.0]

4. AMBA AXI protocol specification [version 1.0]

50 Multi Format Codec (MFC)

50.1 Introduction

The multi-format video codec is a synthesizable core that performs encoding and decoding of multiple streams. The multiple stream encode and decode occurs at 30 fps up to 1080p with resolution image of (1920 × 1080).

The Multi Format Codec (MFC) handles real-time coding up to 16 multi-channels that supports H.264, MPEG4, MPEG2, VC-1, and H.263. The MFC supports up to 16 channels. The actual number of channels depends on use-cases of applications. It also gets limited due to many system specific factors, such as, system memory, clock speed provided to MFC, and so on. The detailed features are described in the latter sections.

50.1.1 Supported Standards

The supported standards of multi format Codec are:

- ITU-T H.264, ISO/IEC 14496-10
  - Decoding: High Profile Level 4.0, 1920 × 1080 at 30 fps at 20 Mbps
    - Supports Baseline Profile Level 4.0,except Flexible Macroblock Ordering (FMO), Arbitrary Slice Ordering (ASO) and Redundant Slice (RS)
    - Supports Main Profile Level 4.0
    - Supports High Profile Level 4.0
  - Encoding: High Profile Level 4.0 1920 × 1080 at 30 fps at 20 Mbps
    - Supports Baseline ,Main, and High Profile, except FMO, ASO and RS
    - Supports 8 × 8 transform in High Profile
    - Supports cyclic intra macroblock refresh

- ITU-T H.263
  - Decoding: Profile 3 Level 70 D1 at 30fps at 8 Mbps
    - Profile 3 restricts up to SD resolution 30 fps
    - Supports H.263 Annexes
      Annex I: Advanced Intra Coding
      Annex J: De-blocking in-loop filter
      Annex K: Slice Structured Mode without RS and ASO
      Annex T: Modified Quantization
      Annex D: Unrestricted Motion Vector Mode
      Annex F: Advanced Prediction Mode except overlapped motion compensation for luminance
  - Encoding: Base Profile at 30 fps at 8 Mbps
    - Baseline Profile
    - Supports custom size up to 1920 × 1088

- ISO/IEC 14496-2 MPEG4
  - Decoding: Advanced Simple Profile Level 5 D1 at 30 fps at 8 Mbps
    - Supports MPEG4 Simple Profile
    - Supports MPEG4 Advanced Simple Profile Level 5
    - Supports Xvid
    - Supports de-blocking filter for post-processing
      Error resilience tools: re-sync marker, data-partitioning with reversible VLC.
    - Data-partitioning supports up to SD resolution
    - Supports quarter pixel motion compensation
    - Restricts Global Motion Compensation (GMC) 1 warp point. It allows 2 and 3 warping points that support quality degradation. This enables continuous decode of corrupted image.
    - Supports one rectangular visual object
    - Supports forward reversible VLC (RVLC)
    - Supports error resilience tool
    - Supports re-use of H.263 in-loop filter for post-processing.
  - Encoding: Advanced Simple Profile Level 5 D1 at 30 fps at 8 Mbps
    - Supports MPEG4 Simple Profile/Advanced Simple Profile, except data partitioning, and RVLC
    - Supports one rectangular visual object
    - Supports DC prediction

- ISO/IEC 13818-2 MPEG2
  - Decoding: Main Profile High Level, 1920 × 1080 at 30 fps at 40 Mbps
    - Supports Main Profile High Level
    - Supports MPEG1 except D-picture

- SMPTE 421M VC-1
  - Decoding: Advanced Profile Level 3, 1920 × 1080 at 30 fps at 45 Mbps
    - Supports Simple Profile Medium Level
    - Supports Main Profile High Level
    - Supports Advanced Profile Level 3
    - Does not process multi-resolution inside video decoder Provides range mapping information for post-processing

50.1.2 Features

The features of multi format Codec are:

- Image features
  - Maximum size: 1920 × 1088 at progressive mode
    1920 × 544 at interlaced mode
  - Minimum size: 32 × 32 at progressive mode
    32 × 16 at interlaced mode
  - Input image size for encoding
    Arbitrary input image size for encoding. There are no constraints such as multiple of 8 or 16.
  - Chrominance interleaved in external memory
  - 4:2:0 format for encoding
  - 4:2:0 format for decoding
  - Supports monochrome for H.264 decoding
  - 8 bits per sample
  - Does not support non-paired field mode
- Slice
  - Minimum size: 16 × 16
- Picture coding structure for encoding
  - Progressive mode
  - Field mode (only H.264)
- Inter prediction for encoding
  - Number of reference frames: Maximum 2 (P frame: 1 or 2, B frame: 2)
  - Search range: Horizontal range of ± 64, Vertical range of ± 32
  - Motion estimation resolution: 1/4 pel for H264, 1/2 pel for MPEG4
  - Number of B frames: 1 or 2
  - Supported modes are:
    H264: 16 × 16, 16 × 8, 8 × 16, 8 × 8, spatial direct mode
    MPEG4: 4MV and UMV
- Intra macroblock for encoding
  - Supports cyclic intra macroblock refresh
  - Intra prediction: Supports 4 × 4 (9 modes), 16 × 16 (4 modes), and 8 × 8 (9 modes) at H.264
- Rate control for encoding
  - Constant Bit-Rate (CBR) and Variable Bit-Rate (VBR)
  - Frame level (H.264/MPEG4/H.263) and macroblock level (H.264) rate control enables or disables selectively
- Stream
  - Time-multiplexed multi-stream encodes/decodes up to 16 channels
  - It should include start code at every position of frame or slice in the stream for decoding.

50.1.3 Target Performance and Functions

Target Performance and Functions section includes:

- Video Decoding Capability
- Video Encoding Capability
- Error Detection

50.1.3.1 Video Decoding Capability

The video decoding capabilities are:

- Decodes up to 1080p at 30 fps at 200 MHz core clock frequency and 200 MHz bus clock frequency.
- Some standards have different maximum resolution, even though MFC has the capability to handle up to 1080p.

50.1.3.2 Video Encoding Capability

The video encoding capabilities are:

- Encodes up to 1080p at 30 fps at 200 MHz core clock frequency and 200 MHz bus clock frequency.
- Verifies the performance numbers under the condition that the number of reference frames for a P frame is 1.
- Some standards have different maximum resolution, even though MFC has the capability to handle up to 1080p.

50.1.3.3 Error Detection

- H264
  - Header error detection
    - This error detection checks if firmware parsing falls out of range.
    - Hardware error detection Variable Length Decoding (VLD) error detection: Detects this error when there is no stream to decode before stream decoder completes decoding.
    - Macroblock error detection: Detects this error when MB type goes out of range. The values should not go beyond 0 to 25 at I slice, 0 to 30 at P slice, and 0 to 48 at B slice.)
    - Sub MB type error detection: Detects this error when sub_mb_type or intra prediction mode goes out of range. The ranges are Intra prediction mode 0 to 7, sub_mb_type 0 to 2 at P slice/0 to 11 at B slice.)
    - IDF error detection: Detects this error when ref_idx value is greater than num_ref_idx_lx_active_minus1. This technique uses Picture Parameter Sets (PPS).
    - Delta Q error detection: Detects this error when mb_qp_delta goes out of range. The error sets when the range goes beyond (– 26 to 25).
    - Timeout interrupt: This interrupt sets when decode time is greater than firmware setting time.

- MPEG2/MPEG4/H.263/VC-1
  - Header error detection: This error detection checks if the result of the firmware parsing falls out of range.
    - Hardware error detection VLD error: Detects this error when the VLD result goes out of table.
    - Coefficient error: Detects this error when AC/DC coefficient of a block exceeds 64
    - Time out interrupt: This interrupt sets when decode time is greater than firmware setting time.

50.2 Hardware Overview

Hardware Overview section includes:

- Block Diagram
- Frame Memory

50.2.1 Block Diagram

The top level of MFC contains hardware modules that include an OpenRISC with 8 KB I-cache and 4 KB D-cache. The optimum partition of the codec functions into software and hardware ensures that small sized hardware supports multiple standards. The hardware operates encoding and decoding at slice level. The firmware on RISC performs slice header parsing and generation. You can change the host processor settings at the frame boundary through host interface.

The block diagram of MFC comprises of RISC, MFC core, RG, bus interface, host interface, and stream interface. MFC core includes many codec accelerators. The RISC and HOST accesses the Register Group (RG). Host and RISC communicates through registers in RG. The register in RG generates a risc2host interrupt. If RISC gets some interrupt or information from hardware, then the RISC sets the registers to let host know the status of MFC. The host clears the interrupt signal when MFC_RISC_HOST_INT Register resets.

There are two AXI master interfaces in which it can use both Port_A and Port_B for full performance. As per the AXI standard, MFC masters handle the Read/Write hazard issues. The internal masters in MFC always checks for response for write access before MFC provides read access for written data.The search SRAM contains reference image for motion estimation and motion compensation.

The shared SRAM shares the current image for encoding.

The pixel cache in MFC core reduces bandwidth for reference image loading. The memory size is 2 KB for Luma and 1 KB for Chroma. For encode of image, the pixel cache loads only the Chroma reference as Luma reference gets loaded prior in the search SRAM for motion estimation. To use the pixel cache for decode of image the reconstruction image should be placed in the Port_A memory area.

Figure 50-1 illustrates the MFC block diagram.

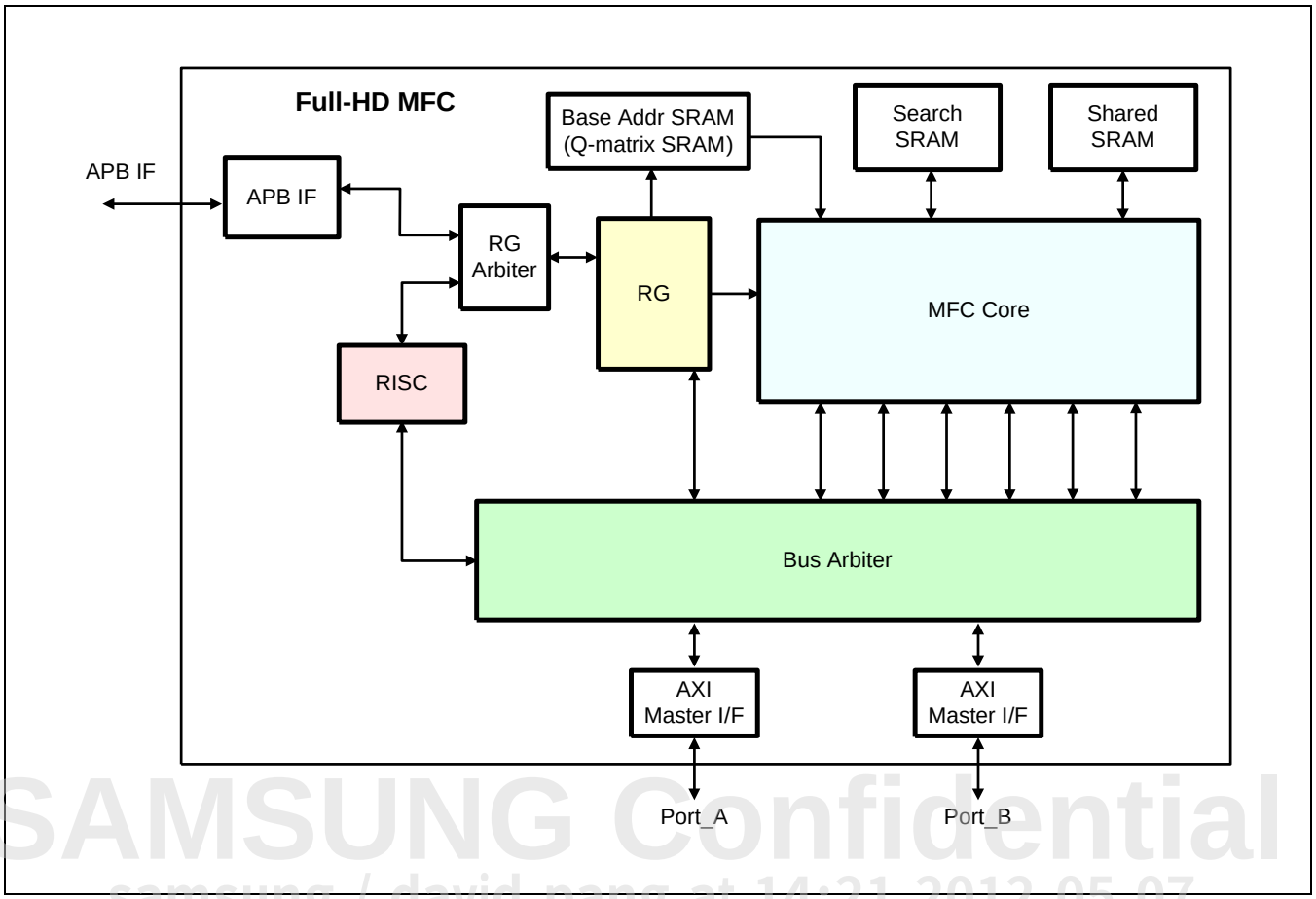


Figure 50-1 MFC Block Diagram

There are several internal masters in MFC core. The bus arbiter controls the interfaces of the internal masters. The firmware controls the internal masters through the index[6:0] register. The index[6:0] register generates an address for the output of AXI interface.

The bus arbiter decides the port to be used based on the index return value. In this case the index is index[6]. If index[6] = 0, it implies that you should use Port_A. If index[6] = 1, it implies that you should use Port_B. The bus arbiter uses index[5:0] to retrieve base address from Q-matrix SRAM. The base address and DRAM_BASE_ADDR in AXI_MASTER calculates the actual address. Therefore, the host should set a base address before codec starts.

50.2.2 Frame Memory

The base address, horizontal and vertical image size specifies the frame memory area. A complete image consists of Y, Cb, and Cr components. It stores the Cb and Cr pixels in a byte interleaved way. Therefore, an image requires two frame buffers. One frame buffer is for Y component and the second frame buffer is for Cb and Cr components. The sum of image horizontal size and image horizontal offset should be a multiple of 16. The horizontal offset forces the horizontal size to be a multiple of 16. The vertical image size for Cb/Cr frame buffer should be half of the Y frame buffer.

Figure 50-2 illustrates the Luma and Chroma pixel that are 8 bytes aligned.

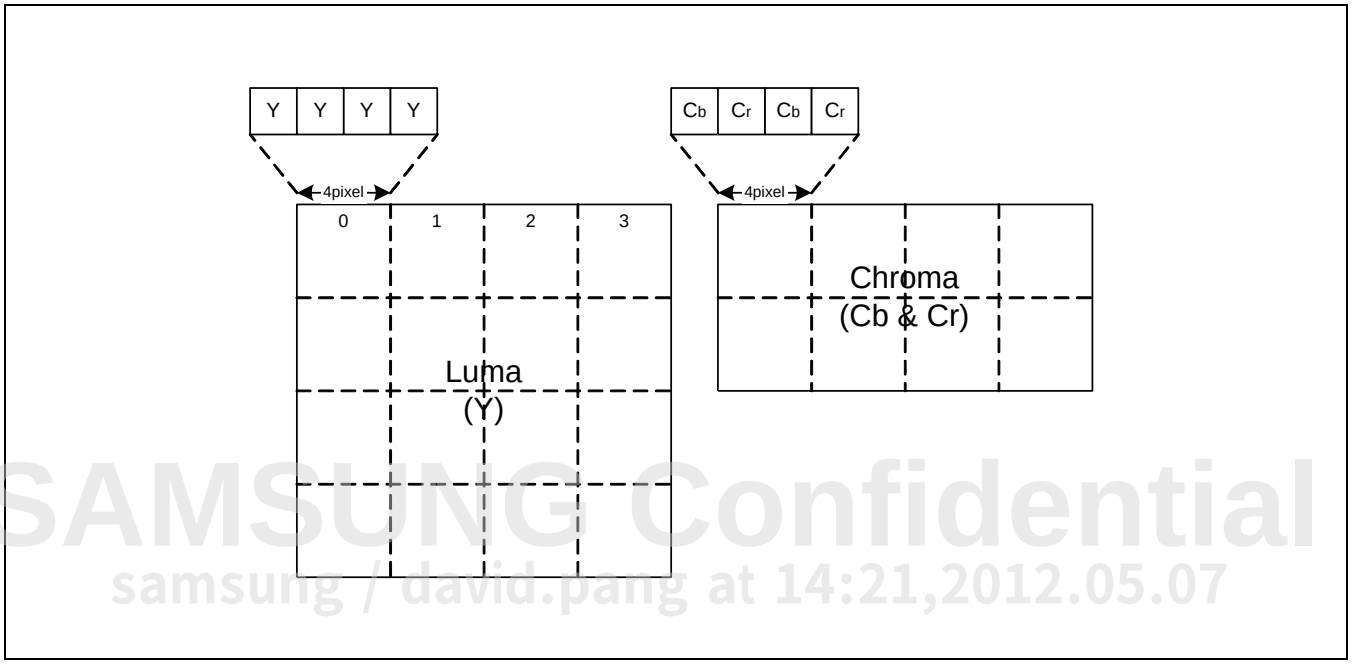


Figure 50-2 Luma and Chroma Pixel (8 bytes-aligned)

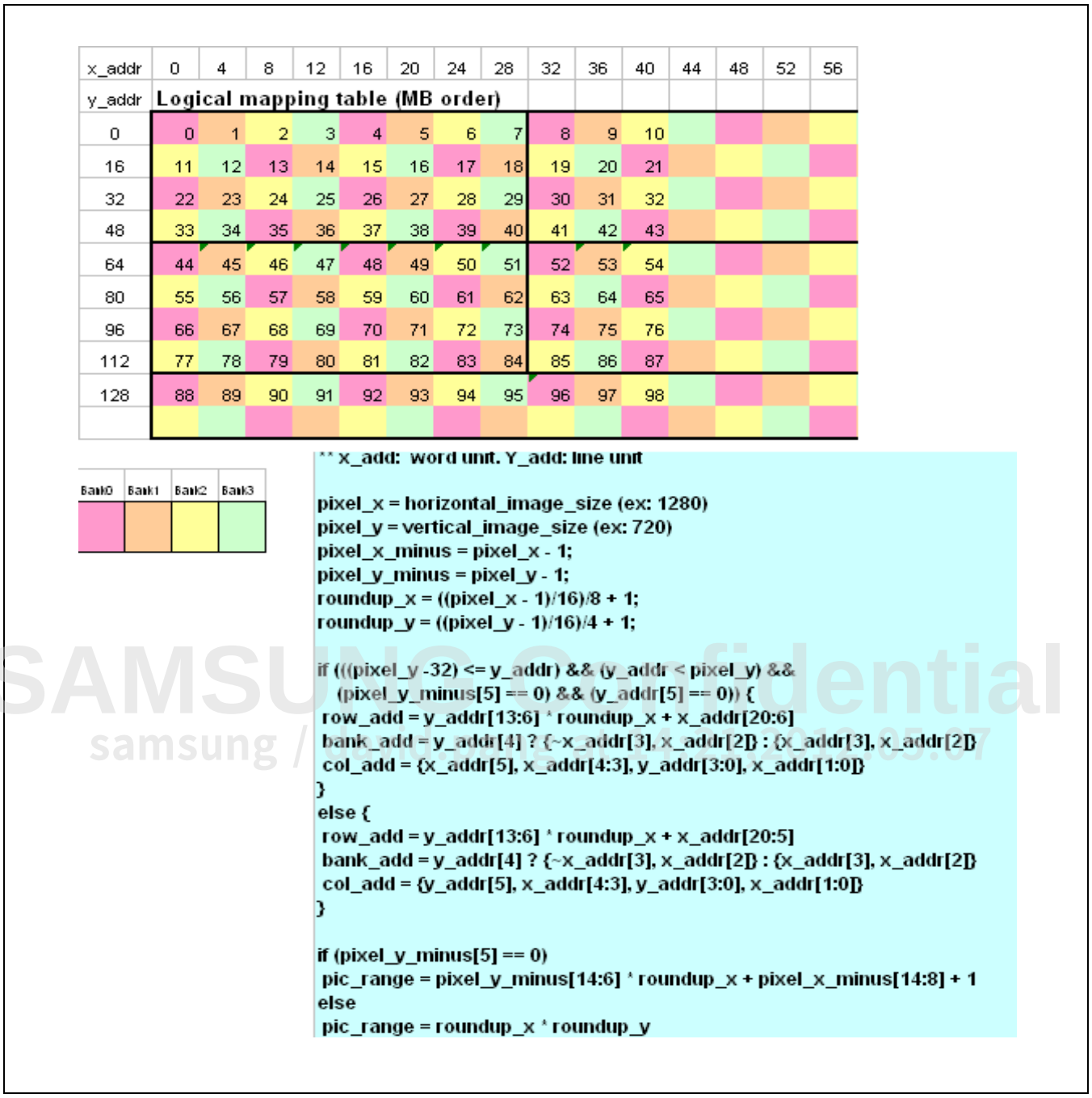
Reference picture is always made in the tile mode memory structure. Decoding reconstruction image is made in 64 pixels × 32 lines tiled mode. Encoding reconstruction image is made in 16 pixels × 16 lines tiled mode.

The two memory structures that can store current picture for encoding depending on the ENC_MAP_FOR_CUR register (0xC51C) are:

- Linear memory structure
- Tile mode memory structure

The host sets external memory parameters with the memory structure configuration. The memory structure, base addresses, and coordinates of pixel in the frame determine the physical memory address of each pixel data.

Figure 50-3 and *Figure 50-4* illustrates the QCIF Image in 16 pixel × 16 lines (1 × 1) Tiled Mode and 64 pixel × 32 lines (4 × 2) Tiled Mode configuration respectively.



x_addr	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56
y_addr	Logical mapping table (MB order)														
0	0	1	2	3	4	5	6	7	8	9	10				
16	11	12	13	14	15	16	17	18	19	20	21				
32	22	23	24	25	26	27	28	29	30	31	32				
48	33	34	35	36	37	38	39	40	41	42	43				
64	44	45	46	47	48	49	50	51	52	53	54				
80	55	56	57	58	59	60	61	62	63	64	65				
96	66	67	68	69	70	71	72	73	74	75	76				
112	77	78	79	80	81	82	83	84	85	86	87				
128	88	89	90	91	92	93	94	95	96	97	98				


```
** x_add:  word unit. Y_add: line unit

pixel_x = horizontal_image_size (ex: 1280)
pixel_y = vertical_image_size (ex: 720)
pixel_x_minus = pixel_x - 1;
pixel_y_minus = pixel_y - 1;
roundup_x = ((pixel_x - 1)/16)/8 + 1;
roundup_y = ((pixel_y - 1)/16)/4 + 1;

if (((pixel_y -32) <= y_addr) && (y_addr < pixel_y) &&
    (pixel_y_minus[5] == 0) && (y_addr[5] == 0)) {
 row_add = y_addr[13:6] * roundup_x + x_addr[20:6]
 bank_add = y_addr[4] ? {~x_addr[3], x_addr[2]} : {x_addr[3], x_addr[2]}
 col_add = {x_addr[5], x_addr[4:3], y_addr[3:0], x_addr[1:0]}
}
else {
 row_add = y_addr[13:6] * roundup_x + x_addr[20:5]
 bank_add = y_addr[4] ? {~x_addr[3], x_addr[2]} : {x_addr[3], x_addr[2]}
 col_add = {y_addr[5], x_addr[4:3], y_addr[3:0], x_addr[1:0]}
}

if (pixel_y_minus[5] == 0)
 pic_range = pixel_y_minus[14:6] * roundup_x + pixel_x_minus[14:8] + 1
else
 pic_range = roundup_x * roundup_y
```


Figure 50-3 QCIF Image in 16 pixel × 16 lines (1 × 1) Tiled Mode

x_addr	0	4	8	12	16	20	24	28	32	36	40	44	48
y_addr	Logical mapping table (MB order)												
0	0	1	2	3	4	5	6	7	8	9	10		
16	11	12	13	14	15	16	17	18	19	20	21		
32	22	23	24	25	26	27	28	29	30	31	32		
48	33	34	35	36	37	38	39	40	41	42	43		
64	44	45	46	47	48	49	50	51	52	53	54		
80	55	56	57	58	59	60	61	62	63	64	65		
96	66	67	68	69	70	71	72	73	74	75	76		
112	77	78	79	80	81	82	83	84	85	86	87		
128	88	89	90	91	92	93	94	95	96	97	98		
144													

Bank0	Bank1	Bank2	Bank3

```
** x_add:  word unit. Y_add: line unit

pixel_x = horizontal_image_size (ex: 1280)
pixel_y = vertical_image_size (ex: 720)
pixel_x_minus = pixel_x - 1;
pixel_y_minus = pixel_y - 1;
roundup_x = ((pixel_x - 1)/16)/8 + 1;
roundup_y = ((pixel_y - 1)/16)/4 + 1;

if (((pixel_y -32) <= y_addr) && (y_addr < pixel_y) &&
   (pixel_y_minus[5] == 0) && (y_addr[5] == 0)) {
 row_add = y_addr[11:6] * roundup_x + x_addr[20:6]
 bank_add = {x_addr[5], x_addr[4]}
 col_add = {y_addr[4:0], x_addr[3:0]}
}
else {
 row_add = y_addr[13:6] * roundup_x + x_addr[20:5]
 bank_add = x_addr[5] ? {~y_addr[5], x_addr[4]} : {y_addr[5], x_addr[4]}
 col_add = {y_addr[4:0], x_addr[3:0]}
}

if (pixel_y_minus[5] == 0)
 pic_range = pixel_y_minus[14:6] * roundup_x + pixel_x_minus[14:8] + 1
else
 pic_range = roundup_x * roundup_y
```


Figure 50-4 QCIF Image in 64 pixel × 32 lines (4 × 2) Tiled Mode

50.3 Register Description

50.3.1 Register Map Summary

- Base Address: 0x1340_0000

Register	Offset	Description	Reset Value
Control Registers			
MFC_SW_RESET	0x0000	Soft reset in MFC.	0x0000_03FE
MFC_RISC _HOST_INT	0x0008	MFC to host interrupt register An interrupt raise occurs when MFC enables the interrupt bit. The Host CPU checks this register, processes the Interrupt Service Routine (ISR) and clears the interrupt bit.	0x0000_0000
MFC_HOST2RISC _COMMAND	0x0030	Host to MFC command register.	0x0000_0000
MFC_HOST2RISC _ARG1	0x0034	HOST2RISC argument register Specifies first argument of the host command	0x0000_0000
MFC_HOST2RISC _ARG2	0x0038	HOST2RISC argument register Specifies second argument of the host command	0x0000_0000
MFC_HOST2RISC _ARG3	0x003C	HOST2RISC argument register Specifies third argument of the host command	0x0000_0000
MFC_HOST2RISC _ARG4	0x0040	HOST2RISC argument register Specifies fourth argument of the host command	0x0000_0000
MFC_RISC2HOST _COMMAND	0x0044	MFC to host command register MFC responds to the host that uses MFC_RISC2HOST_COMMAND register.	0x0000_0000
MFC_RISC2HOST _ARG1	0x0048	MFC to host argument register The MFC_RISC2HOST_COMMAND register uses MFC_RISC2HOST_ARG register data to pass the first argument.	0x0000_0000
MFC_RISC2HOST _ARG2	0x004C	MFC to host argument register Specifies second argument of the host command	0x0000_0000
MFC_RISC2HOST _ARG3	0x0050	MFC to host argument register Specifies third argument of the host command	0x0000_0000
MFC_RISC2HOST _ARG4	0x0054	MFC to host argument register Specifies fourth argument of the host command	0x0000_0000
MFC_FIRMWARE_ VERSION	0x0058	Firmware version information register	0x0000_0000
DBG_INFO _OUTPUT1	0x0064	Debug information output register1	0x0000_0000
DBG_INFO _OUTPUT2	0x0068	Debug information output register2	0x0000_0000
MFC_FIRMWARE _STATUS	0x0080	Firmware status register	0x0000_0000

Register	Offset	Description	Reset Value
MFC_MC_DRAMBASE_ADDR_A	0x0508	DRAM base address that indicates base address for memory map of port A	0xD300_0000
MFC_MC_DRAMBASE_ADDR_B	0x050C	DRAM base address that indicates base address for memory map of port B	0x2300_0000
MFC_MC_STATUS	0x0510	Returns status of bus arbiter This register checks whether the bus is busy or not before MFC resets.	0x0000_000X
MFC_COMMON_BASE_ADDR_0 to 63	0x0600 to 0x06FC	Base address of codec common memory region	X
MFC_COMMON_BASE_ADDR_64 to 127	0x0700 to 0x07FC	Base address of codec common memory region	X
Codec Registers			
MFC_HSIZE_PX	0x0818	Picture width in pixel register at encoder	0x0000_0000
MFC_VSIZE_PX	0x081C	Picture height in pixel register at encoder In interlaced-field coding mode, it specifies the coded height of a field. In the progressive or interlaced-frame coding mode, it specifies the coded height of a frame at encoder.	0x0000_0000
MFC_PROFILE	0x0830	Profile and level control register at encoder	0x0000_0000
MFC_PICTURE_STRUCT	0x083C	Picture structure register Field picture/frame picture flag register at encoder	0x0000_0000
MFC_LF_CONTROL	0x0848	Loop filter control register	0x0000_0000
MFC_LF_ALPHA_OFF	0x084C	H.264 loop filter alpha offset register	0x0000_0000
MFC_LF_BETA_OFF	0x0850	H.264 loop filter beta offset register	0x0000_0000
MFC_QP_OFFSET	0x0C30	Information offset register Specifies QP information offset from DPB start address	0x0000_0000
MFC_QP_OUT_EN	0x0C34	QP information enable register Specifies QP information enable at decoder	0x0000_0000
MFC_SI_RTN_CHID	0x2000	Returns channel instance ID register	0x0000_0000
MFC_COMMON_SI_RG_1 to 15	0x2004 to 0x203C	Returns status of MFC after process	0x0000_0000
MFC_SI_CH0_INST_ID	0x2040	CH0 instance ID and control register	0x0000_0000
MFC_SI_CH1_INST_ID	0x2080	CH1 instance ID and control register	0x0000_0000
MFC_COMMON	0x2044	Host and MFC interface registers through CH0	0x0000_0000

Register	Offset	Description	Reset Value
_CH0_RG_1 to 15	to 0x207C		
MFC_COMMON _CH1_RG_1 to 15	0x2084 to 0x20BC	Host and MFC interface registers through CH1	0x0000_0000
MFC_COMMON _SI_RG_1	0x2004	Vertical resolution register	0x0000_0000
MFC_COMMON _SI_RG_2	0x2008	Horizontal resolution register	0x0000_0000
MFC_COMMON _SI_RG_3	0x200C	Required buffer number register	0x0000_0000
MFC_COMMON _SI_RG_4	0x2010	Luminance address register for display	0x0000_0000
MFC_COMMON _SI_RG_5	0x2014	Chrominance address register for display	0x0000_0000
MFC_COMMON _SI_RG_6	0x2018	Decoded frame size for a frame	0x0000_0000
MFC_COMMON _SI_RG_7	0x201C	Display status register	0x0000_0000
MFC_COMMON _SI_RG_8	0x2020	Frame type register	0x0000_0000
MFC_COMMON _SI_RG_9	0x2024	Luminance address register in decoding order	0x0000_0000
MFC_COMMON _SI_RG_10	0x2028	Chrominance address setting register in decoding order	0x0000_0000
MFC_COMMON _SI_RG_11	0x202C	Decoding status register	0x0000_0000
MFC_COMMON _CHx_RG_1	0x2044 or 0x2084	Start address of the coded picture buffer (CPB) in the external stream buffer	0x0000_0000
MFC_COMMON _CHx_RG_2	0x2048 or 0x2088	Decoding unit size register	0x0000_0000
MFC_COMMON _CHx_RG_3	0x204C or 0x208C	Channel descriptor buffer address	0x0000_0000
MFC_COMMON _CHx_RG_4	0x2050 or 0x2090	Reserved	0x0000_0000
MFC_COMMON _CHx_RG_5	0x2054 or 0x2094	Reserved	0x0000_0000
MFC_COMMON _CHx_RG_6	0x2058 or	CPB Size register	0x0000_0000

Register	Offset	Description	Reset Value
	0x2098		
MFC_COMMON_CHx_RG_7	0x205C or 0x209C	Descriptor buffer size register	0x0000_0000
MFC_COMMON_CHx_RG_8	0x2060 or 0x20A0	Release buffer register that specifies individual DPB availability	0x0000_0000
MFC_COMMON_CHx_RG_9	0x2064 or 0x20A4	Shared memory address	0x0000_0000
MFC_COMMON_CHx_RG_10	0x2068 or 0x20A8	DPB configuration that host prepares for decoding	0x0000_0000
MFC_COMMON_CHx_RG_11	0x206C or 0x20AC	Specifies command sequence number from the host	0x0000_0000
MFC_COMMON_SI_RG_1	0x2004	Encoded stream size register	0x0000_0000
MFC_COMMON_SI_RG_2	0x2008	Encoded picture count register	0x0000_0000
MFC_COMMON_SI_RG_3	0x200C	Stream buffer write pointer	0x0000_0000
MFC_COMMON_SI_RG_4	0x2010	Slice type of current frame to be encoded	0x0000_0000
MFC_COMMON_SI_RG_5	0x2014	Encoded luma address	0x0000_0000
MFC_COMMON_SI_RG_6	0x2018	Encoded chroma address	0x0000_0000
MFC_COMMON_CHx_RG_1	0x2044 or 0x2084	Stream buffer start address at encoder	0x0000_0000
MFC_COMMON_CHx_RG_3	0x204C or 0x208C	Stream buffer size register	0x0000_0000
MFC_COMMON_CHx_RG_4	0x2050 or 0x2090	Current luma address	0x0000_0000
MFC_COMMON_CHx_RG_5	0x2054 or 0x2094	Current chroma address	0x0000_0000
MFC_COMMON_CHx_RG_6	0x2058 or 0x2098	Frame insertion control register	0x0000_0000
MFC_COMMON	0x2064	Shared memory address	0x0000_0000

Register	Offset	Description	Reset Value
_CHx_RG_9	or 0x20A4		
MFC_COMMON _CHx_RG_10	0x2068 or 0x20A8	Flushing input buffer	0x0000_0000
MFC_COMMON _CHx_RG_11	0x206C or 0x20AC	Specifies command sequence number from the host	0x0000_0000
Encoding Registers			
ENC_PIC_TYPE _CTRL	0xC504	Picture type control register	0x0000_0000
ENC_B_RECON _WRITE_ON	0xC508	B-frame reconstructed data write control register	0x0000_0000
ENC_MSLICE _CTRL	0xC50C	Multi-slice control register	0x0000_0000
ENC_MSLICE_MB	0xC510	Slice size register when multi-slice enables. It uses fixed number of macro-blocks that determines the size of one slice.	0x0000_0000
ENC_MSLICE_BIT	0xC514	Slice size register when multi-slice enables. The bit count determines the size of one slice.	0x0000_0000
ENC_CIR_CTRL	0xC518	Intra refresh macro-block setting register	0x0000_0000
ENC_MAP _FOR_CUR	0xC51C	Memory structure setting register of current frame	0x0000_0000
ENC_PADDING _CTRL	0xC520	Padding control register	0x0000_0000
ENC_COMMON _INTRA_BIAS	0xC588	Intra mode bias register for macro-block mode decision	0x0000_0000
ENC_COMMON _BI_DIRECT_BIAS	0xC58C	Bi-directional mode bias register for Macro-block mode decision	0x0000_0000
RC_CONFIG	0xC5A0	Configuration of rate control	0x0000_0000
RC_FRAME_RATE	0xD0D0	Frame rate for frame level RC	0x0000_0000
RC_BIT_RATE	0xC5A8	Target bit-rate for frame level RC	0x0000_0000
RC_QBOUND	0xC5AC	Maximum and minimum value of quantization parameter	0x0000_0000
RC_RPARA	0xC5B0	Rate control reaction coefficient	0x0000_0000
RC_MB_CTRL	0xC5B4	Controls macro-block adaptive scaling features	0x0000_0000
H264_ENC _ENTRP_MODE	0xD004	Entropy coding mode	0x0000_0000
H264_ENC _NUM_OF_REF	0xD010	Maximum number of Reference pictures	0x0000_0000
H264_ENC_TRANS_ 8X8_FLAG	0xD034	8 × 8 transform enable flag in PPS at high profile	0x0000_0000

Register	Offset	Description	Reset Value
H264_ENC_MB_ INFO_ENABLE	0xD140	MB information dump enable	0x0000_0000
MPEG4_ENC _QUART_PXL	0xE008	Quarter pel interpolation control register	0x0000_0000
Shared Memory Structure			
EXTENDED_DECOD E_STATUS	0x0000	Extended decode status	Undefined
SET_FRAME_TAG	0x0004	Sets frame tag of an output frame	Undefined
GET_FRAME _TAG_TOP	0x0008	Gets first frame tag of an output frame	Undefined
GET_FRAME _TAG_BOTTOM	0x000C	Gets second frame tag of an output frame	Undefined
PIC_TIME_TOP	0x0010	Presentation time of an output frame or top field	Undefined
PIC_TIME _BOTTOM	0x0014	Presentation time of bottom field	Undefined
START_BYTE _NUM	0x0018	Specifies an offset of start position in the stream when the start position is not aligned	Undefined
CROP_INFO1	0x0020	Frame cropping information It is valid for H.264 only.	Undefined
CROP_INFO2	0x0024	Frame cropping information It is valid for H.264 only.	Undefined
EXT_ENC _CONTROL	0x0028	Encoder control	Undefined
ENC_PARAM _CHANGE	0x002C	Encoding parameter change that signals change of bit-rate, frame rate, or GOP size	Undefined
VOP_TIMING	0x0030	VOP timing	Undefined
HEC_PERIOD	0x0034	Specifies number of consecutive video packet between header extension codes	Undefined
METADATA _ENABLE	0x0038	Enables storing of metadata information to shared memory	Undefined
METADATA _STATUS	0x003C	Returns the status of metadata	Undefined
METADATA _DISPLAY_INDEX	0x0040	Specifies decoded picture buffer (DPB) number when it conceals the macroblock or enables the QP	Undefined
EXT_METADATA _START_ADDR	0x0044	Start address of metadata memory	Undefined
PUT_EXTRADATA	0x0048	Signals existence of extra metadata	Undefined
EXTRADATA _ADDR	0x004C	Address of extra metadata	Undefined
ALLOCATED _LUMA_DPB_SIZE	0x0064	Size of luma DPB that host allocates for decoding	Undefined

Register	Offset	Description	Reset Value
ALLOCATED_CHROMA_DPB_SIZE	0x0068	Size of chroma DPB that host allocates for decoding	Undefined
ALLOCATED_MV_SIZE	0x006C	Size of motion vector buffers that host allocates for decoding	Undefined
P_B_FRAME_QP	0x0070	P frame QP and B frame QP	Undefined
ASPECT_RATIO_IDC	0x0074	VUI aspect ratio IDC for H.264 encoding	Undefined
EXTENDED_SAR	0x0078	Extended sample aspect ratio for H.264 VUI encoding	Undefined
DISP_PIC_PROFILE	0x007C	Profile Info for displayed picture	Undefined
FLUSH_CMD_TYPE	0x0080	Type of a flushed command	Undefined
FLUSH_CMD_INBUF1	0x0084	Input buffer pointer of a flushed command	Undefined
FLUSH_CMD_INBUF2	0x0088	Input buffer pointer of a flushed command	Undefined
FLUSH_CMD_OUTBUF	0x008C	Output buffer pointer of a flushed command	Undefined
NEW_RC_BIT_RATE	0x0090	Updated target bit rate	Undefined
NEW_RC_FRAME_RATE	0x0094	Updated target frame rate	Undefined
NEW_I_PERIOD	0x0098	Updated intra period	Undefined
H264_I_PERIOD	0x009C	Intra picture period to generate open GOP It is valid for H.264 encoder only.	Undefined
RC_CONTROL_CONFIG	0x00A0	Rate control configuration register for fixed target bit control	Undefined
BATCH_INPUT_ADDR	0x00A4	Start address of Input structure for batch encoding	Undefined
BATCH_OUTPUT_ADDR	0x00A8	Start address of output structure for batch encoding	Undefined
BATCH_OUTPUT_SIZE	0x00AC	Size of output structure for batch encoding	Undefined
MIN_LUMA_DPB_SIZE	0x00B0	Minimum size of luma DPB that host needs to allocate for decoding	Undefined
H264_POC_TYPE	0x00B8	POC_TYPE of H.264 decoder	Undefined
MIN_CHROMA_DPB_SIZE	0x00BC	Minimum size of chroma DPB that host needs to allocate for decoding	Undefined
DISP_PIC_FRAME_TYPE	0x00C0	Frame type of a displayed picture	Undefined
FREE_LUMA_DPB	0x00C4	Available luma DPB address	Undefined
ASPECT	0x00C8	Aspect ratio Information for MPEG4 decoding	Undefined

Register	Offset	Description	Reset Value
_RATIO_INFO			
EXTENDED_PAR	0x00CC	Extended pixel aspect ratio information for MPEG4 decoding	Undefined
DBG_HISTORY _INPUT0	0x00D0	Debug history input register 0 for stage counter settings	Undefined
DBG_HISTORY _INPUT1	0x00D4	Debug history input register1 for stage counter settings	Undefined
DBG_HISTORY _OUTPUT	0x00D8	Output size register for stage counter History	Undefined
HIERARCHICAL _P_QP	0x00E0	QP for hierarchical p frames	Undefined
H264_ENC_MB _INFO_ADDR	0x0720	Base address for MB info dump	Undefined

50.3.2 Control Registers

50.3.2.1 MFC Core Control Register

50.3.2.1.1 MFC_SW_RESET

- Base Address: 0x1340_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_03FE

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	0
RSTN_RG_MPEG2	[9]	RW	Soft reset for RG_MPEG2	1
RSTN_RG_MPEG4	[8]	RW	Soft reset for RG_MPEG4	1
RSTN_RG_VC1	[7]	RW	Soft reset for RG_VC1	1
RSTN_RG_H264	[6]	RW	Soft reset for RG_H264	1
RSTN_RG_COMMON	[5]	RW	Soft reset for RG_COMMON and RG_DECCOM	1
RSTN_DMX	[4]	RW	Soft reset for DMX 0	1
RSTN_VI	[3]	RW	Soft reset for VI	1
RSTN_MFCCORE	[2]	RW	Soft reset for MFC core	1
RSTN_MC	[1]	RW	Soft reset for MC	1
RSTN_RISC	[0]	RW	Soft reset for RISC core	0

50.3.2.1.2 MFC_RISC_HOST_INT

- Base Address: 0x1340_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
INTERRUPT	[0]	RW	Interrupt 0 = Clears interrupt 1 = MFC raises interrupt	0

50.3.2.1.3 MFC_HOST2RISC_COMMAND

- Base Address: 0x1340_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
HOST2RISC _COMMAND	[31:0]	RW	Specifies host to RISC commands 0 = No operation 1 = OPEN_CH (open instance) 2 = CLOSE_CH (close instance) 3 = SYS_INIT (system initialization) 4 = FLUSH_COMMAND (flush commands in ch0, ch1) 5 = SLEEP 6 = WAKEUP 7 = CONTINUE_ENC (continue encoding) 8 = ABORT_ENC (abort encoding)	0

50.3.2.1.4 MFC_HOST2RISC_ARG1

- Base Address: 0x1340_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
HOST2RISC_ARG1	[31:0]	RW	HOST2RISC Argument2 As per HOST2RISC_COMMAND, HOST2RISC_ARG has different meanings such as: OPEN_CH: Specify the codec type as: 0 = H.264 Decoding 1 = VC1 Advanced Profile Decoding 2 = MPEG4/XVid Decoding 3 = MPEG1/MPEG2 Decoding 4 = H.263 Decoding 5 = VC1 Simple/Main Profile Decoding 6 = Reserved 7 = Reserved 8 = Reserved 9 = Reserved 16 = H.264 Encoding 17 = MPEG4 Encoding 18 = H.263 Encoding 19 = VC9 Decoding 20 = VC1 Decoding without start code CLOSE_CH: Specify an instance ID to close. SYS_INIT: Specify the size of the memory for the firmware. In this case the memory size is 400 KB.	0

50.3.2.1.5 MFC_HOST2RISC_ARG2

- Base Address: 0x1340_0000
- Address = Base Address + 0x0038, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
HOST2RISC_ARG2	[30:0]	RW	HOST2RISC Argument2 When HOST2RISC_COMMAND is OPEN_CH, it enables/disables the pixel cache. At encoder, 0 = Enables pixel cache 3 = Disables pixel cache At decoder, 0 = Enables pixel cache for P picture only 1 = Enables pixel cache for B picture only 2 = Enables pixel cache for both P and B pictures 3 = Disables pixel cache	0

50.3.2.1.6 MFC_HOST2RISC_ARG3

- Base Address: 0x1340_0000
- Address = Base Address + 0x003C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CONTEXT_ADDR	[31:0]	RW	Context Memory Address	0

50.3.2.1.7 MFC_HOST2RISC_ARG4

- Base Address = 0x1340_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CONTEXT_SIZE	[31:0]	RW	Context Memory Size H.264 decoder requires a memory size of 600 KB and others require a memory size of 10 KB.	0

50.3.2.1.8 MFC_RISC2HOST_COMMAND

- Base Address: 0x1340_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RISC2HOST_COMMAND	[31:0]	RW	Specifies RISC to Host Commands 0 = RISC2HOST_CMD_EMPTY 1 = RISC2HOST_CMD_OPEN_CH_RET 2 = RISC2HOST_CMD_CLOSE_CH_RET 3 = Reserved 4 = RISC2HOST_CMD_SEQ_DONE_RET 5 = RISC2HOST_CMD_FRAME_DONE_RET 6 = RISC2HOST_CMD_SLICE_DONE_RET 7 = RISC2HOST_CMD_ENC_COMPLETE_RET 8 = RISC2HOST_CMD_SYS_INIT_RET 9 = RISC2HOST_CMD_FIRMWARE_STATUS_RET 10 = RISC2HOST_CMD_SLEEP_RET 11 = RISC2HOST_CMD_WAKEUP_RET 12 = RISC2HOST_CMD_FLUSH_COMMAND_RET 13 = RISC2HOST_CMD_ABORT_RET 14 = RISC2HOST_CMD_BATCH_ENC_RET 15 = RISC2HOST_CMD_INIT_BUFFERS_RET 16 = RISC2HOST_CMD_EDFU_INT_RET 17 to 31 = Reserved 32 = RISC2HOST_CMD_ERROR_RET	0

50.3.2.1.9 MFC_RISC2HOST_ARG1

- Base Address: 0x1340_0000
- Address = Base Address + 0x0048, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MFC_RISC2HOST_ARG1	[31:0]	RW	RISC2HOST Argument Register OPEN_CH_RET: Returns an instance ID. SYS_INIT_RET: Returns firmware memory size. SEQ_START_RET: Returns a channel ID FRAME_START_RET: Returns a channel ID LAST_SEQ_RET: Returns a channel ID INIT_BUFFERS_RET: Returns a channel ID FRAME_START_REALLOC_RET: Returns a channel ID FLUSH_COMMAND_RET: [31:16]: Returns Instance ID of CH1 [15:0]: Returns Instance ID of CH0	0

When host receives FLUSH_COMMAND_RET, it checks the shared memory at 0x80 and 0x8C to find the input and output pointers in each command channel. If [31:16] is not equal to 0xFFFF, then it flushes the command in CH1. If [15:0] is not equal to 0xFFFF, then it flushes the command in CH0.

50.3.2.1.10 MFC_RISC2HOST_ARG2

- Base Address: 0x1340_0000
- Address = Base Address + 0x004C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DISP_ERROR_STATUS	[31:16]	RW	Specifies error status for displayed frame Section 50.3.2.2 Error Codes defines the error codes	0
DEC_ERROR_STATUS	[15:0]	RW	Specifies error status for decoded/encoded frame Section 50.3.2.2 Error Codes defines the error codes	0

50.3.2.1.11 MFC_RISC2HOST_ARG3

- Base Address: 0x1340_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MFC_RISC2HOST_ARG3	[31:0]	RW	RISC2HOST Argument RegisterFRAME_DONE_RET: Returns the specified size of encoded stream. SLICE_DONE_RET: Returns the specified size of encoded stream.	0

50.3.2.1.12 MFC_RISC2HOST_ARG4

- Base Address: 0x1340_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MFC_RISC2HOST_ARG4	[31:0]	RW	Reserved	0

50.3.2.1.13 MFC_FIRMWARE_VERSION

- Base Address: 0x1340_0000
- Address = Base Address + 0x0058, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0
YEAR	[23:16]	R	Specifies Year: 00 to 99 NOTE: 00 means year 2000.	0
MONTH	[15:8]	R	Specifies Month: 1 to 12	0
DAY	[7:0]	R	Specifies Day: 1 to 31	0

50.3.2.1.14 DBG_INFO_OUTPUT1

- Base Address: 0x1340_0000
- Address = Base Address + 0x0064, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
INTERMEDIATE_ STAGE_COUNTER	[31:0]	R	Intermediate Stage Counter for different stages of MFC firmware The counter values have different interpretation for each codec.	0

50.3.2.1.15 DBG_INFO_OUTPUT2

- Base Address: 0x1340_0000
- Address = Base Address + 0x0068, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
EXCEPTION_STATUS	[31:0]	R	Status of Exception Handler 0x01 = Bus error handler 0x02 = Illegal instruction handler 0x04 = Tick handler 0x10 = Trap handler 0x20 = Align handler 0x40 = Range handler 0x80 = DTLB miss exception handler 0x100 = ITLB miss exception handler 0x200 = Data page fault exception handler 0x400 = Instruction page fault exception handler	0

50.3.2.1.16 MFC_FIRMWARE_STATUS

- Base Address: 0x1340_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
FIRMWARE_STATUS	[0]	R	Specifies Firmware Status 0 = Not ready 1 = Ready	0

50.3.2.2 Error Codes

Error Code	Error Name	Description
Command Control Errors		
1	INVALID_CHANNEL _NUMBER	Specifies the use of an invalid channel number. The allowable limit falls in the range from 0 to 15.
2	INVALID_COMMAND_ID	Specifies the use of an illegal command. You should use the commands in CHx_INST_ID register specification.
3	CHANNEL_ALREADY _IN_USE	Specifies an error if channel status is open and host tries to re-open the channel before closing it
4	CHANNEL_NOT_OPEN _BEFORE_CHANNEL _CLOSE	Specifies an error when CLOSE_CH completes before OPEN_CH when OPEN_CH = 0.
5	OPEN_CH_ERROR _SEQ_START	Specifies an error if channel is not open. It specifies ERROR in SEQ_START.
6	SEQ_START _ALREADY_CALLED	Specifies an error if it issues SEQ_START for the same channel after it completes its operation.
7	OPEN_CH_ERROR _INIT_BUFFERS	Specifies an error if it does not open the channel in INIT_BUFFERS.
8	SEQ_START_ERROR _INIT_BUFFERS	Specifies an error if it issues SEQ_START for the same channel after it completes its operation.
9	INIT_BUFFER _ALREADY_CALLED	Specifies an error if INIT_BUFFERS is already complete when HOST tries INIT_BUFFERS.
10	OPEN_CH_ERROR _FRAME_START	Specifies an error if channel is not open. It specifies ERROR in FRAME_START.
11	SEQ_START_ERROR _FRAME_START	"not complete" does not mean "fail" Specifies an error if SEQ_START is not complete before FRAME_START.
12	INIT_BUFFERS_ERROR _FRAME_START	Specifies an error if INIT_BUFFERS is not complete to complete before FRAME_START.
13	CODEC_LIMIT _EXCEEDED	Specifies an error if number of codecs exceed 16. Currently, this feature is not applicable.)
20	MEM_ALLOCATION _FAILED	Specifies an error when memory allocation fails in firmware
25	INSUFFICIENT _CONTEXT_SIZE	Specifies an error for insufficient context buffer size
SEQ_START Errors		
27	UNSUPPORTED _FEATURE_ IN_PROFILE	It does not support features like CABAC/Interlace in baseline profile.
28	RESOLUTION_NOT _SUPPORTED	It does not support resolution.
Decoder Fatal Errors on SEQ_START		

Error Code	Error Name	Description
52	HEADER_NOT_FOUND	Header Not Found
Encoder Fatal Errors on SEQ_START		
61	RESERVED	Reserved
62	FRAME_RATE_NOT _SUPPORED	When it enables Rate Control, the Frame Rate cannot have a zero value.
63	INVALID_QP_VALUE	Specifies an error for invalid QP value.
64	INVALID_RC_REACTION _COEFFICIENT	Specifies an error for invalid value of rate control reaction parameter. It does not allow Zero value.
65	INVALID_CPB_SIZE_AT _GIVEN_LEVEL	Specifies an error for invalid value of CPB/VBV size at a given level The invalid value violates Annex A in H.264.
INIT_BUFFERS Errors		
71	ALLOC_DPB_SIZE_NOT _SUFFICIENT	Specifies an error for insufficient DPB SIZE allocation.
72	RESERVED	Reserved
73	RESERVED	Reserved
74	NUM_DPB_OUT _OF_RANGE	Specifies an error if NUM_DPB goes out of range NUM_DPB value should be equal or greater than MIN_NUM_DPB and equal to or smaller than 32.
77	NULL_METADATA _INPUT_POINTER	Specifies an error for NULL value of external metadata input structure address.
78	NULL_DPB_POINTER	Specifies an error for NULL value of allocated DPB address.
79	NULL_OTH_EXT_BUF _ADDR	Specifies an error for NULL value of other external buffers for decoder.
80	NULL_MV_POINTER	Specifies an error for NULL value of MV address.
Common Hardware Errors		
81	DIVIDE_BY_ZERO	Specifies divide By Zero Error.
82	BIT_STREAM_BU F_EXHAUST	Specifies an error when bit stream buffer exhausts.
83	DESCRIPTOR_BUFFER _EMPTY	Specifies an error for empty descriptor buffer. It holds valid for H264 and VC1 decoders only.
84	DMA_TX_NOT _COMPLETE	Specifies DMA Transmission Not Complete Error.
Decoder Hardware Errors		
85	MB_HEADER_NOT _DONE	Specifies MB Header Decode Not Done Error.
86	MB_COEFF_NOT_DONE	Specifies MB Co-efficient Not Done Error. This error occurs at entropy decoding.
87	CODEC_SLICE_NOT _DONE	Specifies Codec Slice Done Error.

Error Code	Error Name	Description
88	MFC_CORE_TIME_OUT	Specifies timeout error for hardware processing.
89	VC1_BITPLANE _DECODE_ERR	Specifies VC1 Bit Plane Decode Error.
Encoder Hardware Errors		
90	VSP_NOT_READY	Specifies VSP Not Ready Error.
91	BUFFER_FULL_STATE	Specifies an error if buffer gets full.
Decoder Errors on FRAME_RUN		
112	RESOLUTION _MISMATCH	Specifies an error if the resolution in GOV exceeds the values in sequence header.
113	NV_QUANT_ERR	Specifies errors in quantization parameters.
114	SYNC_MARKER_ERR	Specifies Sync Marker Error.
115	FEATURE_NOT _SUPPORTED	Specifies an error for unsupported feature in the profile.
116	MEM_CORRUPTION	Specifies an error if it corrupts the MFC core memory.
117	INVALID_REFERENCE _FRAME	Specifies an error for invalid reference frames.
118	PICTURE_CODING _TYPE_ERR	Specifies PICTURE_CODING_TYPE error in MPEG2.
119	MV_RANGE_ERR	Specifies invalid Fcode in MV_Range).
120	PICTURE_STRUCTURE _ERR	Specifies picture structure error in FRAME/TOP/BOTTOM field.
121	SLICE_ADDR_INVALID	Specifies error for invalid slice address.
122	NON_PAIRED_FIELD _NOT_SUPPORTED	Specifies an error for unsupported non-paired fields.
123	NON_FRAME_DATA _RECEIVED	Specifies an error if received data contains only the header but not the frame data. For example, an error occurs, if seq headers, namely, SPS/PPS, SEI get received but not its associated frame data.
124	INCOMPLETE_FRAME	Specifies an error for incomplete frame data received (For example, this error occurs when it receives slices that are in incomplete frames.)
125	NO_BUFFER_RELEASED _FROM_HOST	Specifies an error for unavailability of free buffers. This error occurs when host locks all buffers or they are anchor frames and you cannot use.
128	NALU_HEADER_ERROR	Specifies an error for invalid NALU Header
129	SPS_PARSE_ERROR	Specifies an error for invalid syntax element in SPS
130	PPS_PARSE_ERROR	Specifies an error for invalid syntax element in PPS
131	SLICE_PARSE_ERROR	Specifies an error for invalid syntax element in slice header
Common Warnings		
145	COMMAND_FLUSHED	Flushes FRAME_START command from channel 0 and channel1

Error Code	Error Name	Description
Decoder Warnings		
150	METADATA_NO_SPACE _NUM_CONCEAL_MB	Specifies an error if there is no space for number of concealed MB metadata output.
151	METADATA_NO _SPACE_QP	Specifies an error if there is no space for QP metadata output.
152	METADATA_NO_SAPCE _CONCEAL_MB	Specifies an error if there exists no space for concealed MB output.
153	METADATA_NO_SPACE _VC1_PARAM	Specifies an error if there exists no space for VC1 parameter .output
154	METADATA_NO_SPACE _SEI	Specifies an error if there is no space for SEI information output.
155	METADATA_NO _SPACE_VUI	Specifies an error if there is no space for VUI information output.
156	METADATA_NO _SPACE_EXTRA	Specifies an error if there is no space for extra data output.
157	METADATA_NO _SPACE_DATA_NONE	Specifies an error if there is no space for DataNone.
158	FRAME_RATE _UNKNOWN	Specifies an error for unknown frame rate.
159	ASPECT_RATIO _UNKOWN	Specifies an error for unknown aspect ratio.
160	COLOR_PRIMARIES _UNKNOWN	Specifies an error for invalid color primaries.
161	TRASNFER_CHAR _UNKWON	Specifies an error for invalid transfer characteristics.
162	MATRIX_COEFF _UNKNOWN	Specifies an error for invalid matrix coefficients.
163	NON_SEQ_SLICE_ADDR	Specifies an error if the new slice address is not sequential with respect to the old address.
164	BROKEN_LINK	Specifies an error if current GOV contain B pictures whose anchor frame still exists in previous GOV.
165	FRAME_CONCEALED	Specifies an error if error concealment is done by MFC.
166	PROFILE_UNKOWN	Specifies as error for an unknown profile.
167	LEVEL_UNKOWN	Specifies as error for an unknown level.
168	BIT_RATE_NOT _SUPPORTED	Specifies an error if it does not support bit rate.
169	COLOR_DIFF_FORMAT _NOT_SUPPORTED	Specifies an error if it does not support color format.
170	NULL_EXTRA _METADATA_POINTER	Specifies an error for NULL value of allocated memory address of extra metadata.
171	SYNC_POINT_NOT	Specifies an error for an empty DPB with a non I/IDR frame

Error Code	Error Name	Description
	_RECEIVED_STARTED _DECODING	MFC starts decoding from the non I/IDR frame.
172	NULL_FW_DEBUG _INFO_POINTER	Specifies an error for NULL value of FW debug information address.
173	ALLOC_DEBUG_INFO _SIZE_INSUFFICIENT	Specifies an error for insufficient allocated size for debug information.
Encoder Warnings		
180	METADATA_NO _SPACE_ MB_INFO	Specifies an error if there is no space for macroblock information output. It holds valid only for H.264 encoder.
181	METADATA_NO _SPACE_SLICE_SIZE	Specifies an error if there is no space for slice size output.
182	RESOLUTION_WARNING	Specifies an error if given H.263 encoder profile does not support the resolution setting.

50.3.2.3 Memory Controller Registers

50.3.2.3.1 MFC_MC_DRAMBASE_ADDR_A

- Base Address: 0x1340_0000
- Address = Base Address + 0x0508, Reset Value = 0XD300_0000

Name	Bit	Type	Description	Reset Value
MC_DRAMBASE_ADDR_A	[31:17]	RW	Channel A DRAM Base Address Register The DRAM base address should be aligned at 128 KB. The access range of MFC through port A varies from DRAMBASE_ADDR_A to (DRAMBASE_ADDR_A + 256 MB).	0x6980
RSVD	[16:0]	–	Reserved	0

50.3.2.3.2 MFC_MC_DRAMBASE_ADDR_B

- Base Address: 0x1340_0000
- Address = Base Address + 0x050C, Reset Value = 0x2300_0000

Name	Bit	Type	Description	Reset Value
MC_DRAMBASE_ADDR_B	[31:17]	RW	Channel B DRAM Base Address Register The DRAM base address should be aligned at 128 KB. The access range of MFC through port B varies from DRAMBASE_ADDR_B to (DRAMBASE_ADDR_B + 256 MB).	0x1180
RSVD	[16:0]	–	Reserved	0

50.3.2.3.3 MFC_MC_STATUS

- Base Address: 0x1340_0000
- Address = Base Address + 0x0510, Reset Value = 0x0000_000X

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved Do not use this bit.	0
MC_BUSY_B	[1]	R	Specifies if port B status is Busy 0 = Idle 1 = Busy	X
MC_BUSY_A	[0]	R	Specifies if port A status is Busy 0 = Idle 1 = Busy	X

NOTE: X stands for undetermined value.

50.3.2.4 Common Address Control

Each codec requires a common base address. The interpretation of common base address varies. Refer to the sections *50.3.2.5 Buffer Address of Decoder* and *50.3.2.6 Buffer Address of Encoder* for more information. You can define the common base addresses (0 to 63) for AXI_MEMORY_A and you can define the common base addresses (64 to 147) for AXI_MASTER_B.

NOTE: You can determine the base address as:
Base address= (MC_DRAMBASE_ADDR) + (MFC_COMMON_BASE_ADDR << 11)

50.3.2.4.1 MFC_COMMON_BASE_ADDR_0 to 63

- Base Address: 0x1340_0000
- Address = Base Address + 0x0600 to 0x06FC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MFC_COMMON_BASE_ADDR_0 to MFC_COMMON_BASE_ADDR_63	[16:0]	RW	Common Base RAM Register 0 to 63 for Port_A Specifies start address for codec common memory region.	X

50.3.2.4.2 MFC_COMMON_BASE_ADDR_64 to 127

- Base Address: 0x1340_0000
- Address = Base Address + 0x0700 to 0x07FC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MFC_COMMON_BASE_ADDR_64 to MFC_COMMON_BASE_ADDR_127	[16:0]	RW	Common Base RAM Register 64 to 147 for Port_B Specifies start address for codec common memory region.	X

50.3.2.5 Buffer Address of Decoder

This section describes the base address settings for different standards.

- H264 Decoder

Memory Region	Register Name	Description
H264DEC_VERT_NB_MV	MFC_COMMON_BASE_ADDR_35	Vertical Neighbor Motion Vector Buffer
H264DEC_NB_IP	MFC_COMMON_BASE_ADDR_36	Neighbor Pixels for Intra Prediction Buffer
H264DEC_LUMA_x	MFC_COMMON_BASE_ADDR_64 to 95	Luma DPB for Master Channel A
H264DEC_CHROMA_x	MFC_COMMON_BASE_ADDR_0 to 31	Chroma DPB for Master Channel B
H264DEC_MV_x	MFC_COMMON_BASE_ADDR_96 to 127	MV Buffer for H.264

- MPEG4 Decoder

Memory Region	Register Name	Description
DEC_NB_DCAC	MFC_COMMON_BASE_ADDR_35	Neighbor information of stream parser
DEC_UPNB_MV	MFC_COMMON_BASE_ADDR_36	Neighbor information of stream parser
DEC_SUB_ANCHOR_MV	MFC_COMMON_BASE_ADDR_37	Neighbor information of stream parser
OVERLAP_TRANSFORM	MFC_COMMON_BASE_ADDR_38	Information for Motion Compensation
DEC_STX_PARSER	MFC_COMMON_BASE_ADDR_42	Syntax Parser Buffer
DEC_LUMA_x	MFC_COMMON_BASE_ADDR_64 to 95	Reconstructed Luma plane
DEC_CHROMA_x	MFC_COMMON_BASE_ADDR_0 to 31	Reconstructed Chroma plane

- MPEG2 Decoder

Memory Region	Register Name	Description
DEC_LUMA_x	MFC_COMMON_BASE_ADDR_64 to 95	Reconstructed Luma plane
DEC_CHROMA_x	MFC_COMMON_BASE_ADDR_0 to 31	Reconstructed Chroma plane

- H.263 Decoder

Memory Region	Register Name	Description
DEC_NB_DCAC	MFC_COMMON_BASE_ADDR_35	Neighbor information of stream parser
DEC_UPNB_MV	MFC_COMMON_BASE_ADDR_36	Neighbor information of stream parser
DEC_SUB_ANCHOR_MV	MFC_COMMON_BASE_ADDR_37	Neighbor information of stream parser
OVERLAP_TRANSFORM	MFC_COMMON_BASE_ADDR_38	Information for Motion Compensation
DEC_LUMA_x	MFC_COMMON_BASE_ADDR_64 to 95	Reconstructed Luma plane
DEC_CHROMA_x	MFC_COMMON_BASE_ADDR_0 to 31	Reconstructed Chroma plane

- VC1 Decoder

Memory Region	Register Name	Description
DEC_NB_DCAC	MFC_COMMON_BASE_ADDR_35	Neighbor information of stream parser
DEC_UPNB_MV	MFC_COMMON_BASE_ADDR_36	Neighbor information of stream parser
DEC_SUB_ANCHOR_MV	MFC_COMMON_BASE_ADDR_37	Neighbor information of stream parser
OVERLAP_TRANSFORM	MFC_COMMON_BASE_ADDR_38	Information for Motion Compensation
BITPLANE3	MFC_COMMON_BASE_ADDR_39	Bit Plane
BITPLANE2	MFC_COMMON_BASE_ADDR_40	
BITPLANE1	MFC_COMMON_BASE_ADDR_41	
DEC_LUMA_x	MFC_COMMON_BASE_ADDR_64 to 95	Reconstructed Luma plane
DEC_CHROMA_x	MFC_COMMON_BASE_ADDR_0 to 31	Reconstructed Chroma plane

- Buffer Memory Size for Decoder

Memory Region	Size			
	H.264	MPEG4/H.263	VC1	MPEG2
CH_ES_ADDR	Up to 4 MB			
CH_DESC_ADDR	Up to 128 KB			
DEC_NB_DCAC	–	16 KB	16 KB	–
DEC_UPNB_MV	–	68 KB	68 KB	–
DEC_SUB_ANCHOR_MV	–	136 KB	136 KB	–
DEC_OVERLAP_TRANSFORM	–	32 KB	32 KB	–
DEC_BITPLANE3	–	–	2 KB	–
DEC_BITPLANE2	–	–	2 KB	–
DEC_BITPLANE1	–	–	2 KB	–
DEC_STX_PARSER	–	68 KB (does not require for H263)	–	–
DEC_LUMA_x	–	align (align (x_size, 128) × align(y_size, 32), 8192)		
DEC_CHROMA_x	–	align (align (x_size, 128) × align (y_size/2, 32), 8192)		
H264DEC_VERT_NB_MV	16 KB	–	–	–
H264DEC_NB_IP	32 KB	–	–	–
H264DEC_CHROMA_x	align (align (x_size, 128) × align (y_size/2, 32), 8192) + align (align (x_size, 128) × align (y_size/4, 32), 8192)	–	–	–
H264DEC_LUMA_x	align (align (x_size, 128) × align (y_size, 32), 8192)	–	–	–
H264DEC_MV_x	Quarter size of H.264 Luma DPB align (align (x_size, 128) × align (y_size/4, 32), 8192)	–	–	–

NOTE:

1. Align all linear information from this table at 2 KB and tile mode information (Luma/Chroma DPB and H264DEC_MV) at 8 KB.
2. DPB size does not take into account the QP save area.
3. "x" ranges from 0 to 31 for LUMA_x, CHROMA_x, MV_x.

50.3.2.6 Buffer Address of Encoder

- H.264 Encoder

Memory Region	Register Name in User's Manual	Description
ENC_DPB_Y0_ADDR	MFC_COMMON_BASE_ADDR_7	Reconstructed Y0 buffer
ENC_DPB_C0_ADDR	MFC_COMMON_BASE_ADDR_64	Reconstructed C0 buffer
ENC_DPB_Y1_ADDR	MFC_COMMON_BASE_ADDR_8	Reconstructed Y1 buffer
ENC_DPB_C1_ADDR	MFC_COMMON_BASE_ADDR_65	Reconstructed C1 buffer
ENC_DPB_Y2_ADDR	MFC_COMMON_BASE_ADDR_68	Reconstructed Y2 buffer
ENC_DPB_C2_ADDR	MFC_COMMON_BASE_ADDR_66	Reconstructed C2 buffer
ENC_DPB_Y3_ADDR	MFC_COMMON_BASE_ADDR_69	Reconstructed Y3 buffer
ENC_DPB_C3_ADDR	MFC_COMMON_BASE_ADDR_67	Reconstructed C3 buffer
UPPER_MV_ADDR	MFC_COMMON_BASE_ADDR_0	Upper row MV storage region
DIRECT_COLZERO_FLAG_ADDR	MFC_COMMON_BASE_ADDR_4	Direct Co-located Flag storage region
UPPER_INTRA_MD_ADDR	MFC_COMMON_BASE_ADDR_2	Upper row current pixel data storage region
UPPER_INTRA_PRED_ADDR	MFC_COMMON_BASE_ADDR_80	Upper row pre-filter reconstruction data storage region
NBOR_INFO_MPENC_ADDR	MFC_COMMON_BASE_ADDR_1	Neighbor MB information storage region

- H.263 Encoder

Memory Region	Register Name in User's Manual	Description
ENC_DPB_Y0_ADDR	MFC_COMMON_BASE_ADDR_7	Reconstructed Y0 buffer
ENC_DPB_C0_ADDR	MFC_COMMON_BASE_ADDR_64	Reconstructed C0 buffer
ENC_DPB_Y1_ADDR	MFC_COMMON_BASE_ADDR_8	Reconstructed Y1 buffer
ENC_DPB_C1_ADDR	MFC_COMMON_BASE_ADDR_65	Reconstructed C1 buffer
ENC_DPB_Y2_ADDR	MFC_COMMON_BASE_ADDR_68	Reconstructed Y2 buffer
ENC_DPB_C2_ADDR	MFC_COMMON_BASE_ADDR_66	Reconstructed C2 buffer
ENC_DPB_Y3_ADDR	MFC_COMMON_BASE_ADDR_69	Reconstructed Y3 buffer
ENC_DPB_C3_ADDR	MFC_COMMON_BASE_ADDR_67	Reconstructed C3 buffer
UPPER_MV_ADDR	MFC_COMMON_BASE_ADDR_0	Upper row MV storage region
ACDC_COEF_BASE_ADDR	MFC_COMMON_BASE_ADDR_1	Upper row inverse quantization coefficient storage region

- MPEG4 Encoder

Memory Region	Register Name in User's Manual	Description
ENC_DPB_Y0_ADDR	MFC_COMMON_BASE_ADDR_7	Reconstructed Y0 buffer
ENC_DPB_C0_ADDR	MFC_COMMON_BASE_ADDR_64	Reconstructed C0 buffer
ENC_DPB_Y1_ADDR	MFC_COMMON_BASE_ADDR_8	Reconstructed Y1 buffer
ENC_DPB_C1_ADDR	MFC_COMMON_BASE_ADDR_65	Reconstructed C1 buffer
ENC_DPB_Y2_ADDR	MFC_COMMON_BASE_ADDR_68	Reconstructed Y2 buffer
ENC_DPB_C2_ADDR	MFC_COMMON_BASE_ADDR_66	Reconstructed C2 buffer
ENC_DPB_Y3_ADDR	MFC_COMMON_BASE_ADDR_69	Reconstructed Y3 buffer
ENC_DPB_C3_ADDR	MFC_COMMON_BASE_ADDR_67	Reconstructed C3 buffer
UPPER_MV_ADDR	MFC_COMMON_BASE_ADDR_0	Upper row MV storage region
DIRECT_COLZERO_FLAG_ADDR	MFC_COMMON_BASE_ADDR_4	Skip flag storage region
ACDC_COEF_BASE_ADDR	MFC_COMMON_BASE_ADDR_1	Upper row inverse quantization coefficient storage region

- Buffer Memory Size for Encoder

Memory Region	H.264	MPEG4/H.263
ENC_DPB_Y0_ADDR	align (align (x_size, 128) × align (y_size, 32), 8192)	
ENC_DPB_C0_ADDR	align (align (x_size, 128) × align (y_size/2, 32), 8192)	
ENC_DPB_Y1_ADDR	align (align (x_size, 128) × align (y_size, 32), 8192)	
ENC_DPB_C1_ADDR	align (align (x_size, 128) × align (y_size/2, 32), 8192)	
ENC_DPB_Y2_ADDR	align (align (x_size, 128) × align (y_size, 32), 8192)	
ENC_DPB_C2_ADDR	align (align (x_size, 128) × align (y_size/2, 32), 8192)	
ENC_DPB_Y3_ADDR	align (align (x_size, 128) × align (y_size, 32), 8192)	
ENC_DPB_C3_ADDR	align (align (x_size, 128) × align (y_size/2, 32), 8192)	
CH_SB_ADDR	Configurable Limit1: Align at 2 KB Limit2: Multiple of 4 KB	Configurable Limit1: Align at 2 KB Limit2: Multiple of 4 KB
UPPER_MV_ADDR	align (xMB_size × 2 × 8, 2048) byte	align (xMB_size × 2 × 8, 2048) byte
DIRECT_COLZERO _FLAG_ADDR	align (((xMB_size × yMB_size+7)/8) × 8, 2048) byte	align (((xMB_size × yMB_size+7)/8) × 8, 2048) byte (does not require for H263 encoder)
UPPER_INTRA_MD_ADDR	align (xMB_size × 48, 2048) byte	–
UPPER_INTRA_PRED_ADDR	align (1024 × 2 × 8, 2048) byte	–
NBOR_INFO_MPENC_ADDR	CAVLC: align (xMB_size × 8 × 8, 2048) byte CABAC: align (xMB_size × 24 × 8, 2048) byte	–
ACDC_COEF_BASE_ADDR	–	align ((x_size/2) × 8, 2048) byte

NOTE:

1. Align all linear information from this table at 2 KB and tile mode information (Luma/Chroma DPB) at 8 KB.
2. The division operation in the table is an integer division.
3. When you use 1 reference P, the required DPBs are DPB_Y0, DPB_Y1, DPB_C0, and DPB_C1..
   The current Luma/Chroma buffers are placed in the opposite port.

50.3.3 Codec Registers

50.3.3.1 Codec Common Registers

50.3.3.1.1 MFC_HSIZE_PX

- Base Address: 0x1340_0000
- Address = Base Address + 0x0818, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	0
PICTURE_WIDTH	[12:0]	RW	Specifies coded width of a picture	0

50.3.3.1.2 MFC_VSIZE_PX

- Base Address: 0x1340_0000
- Address = Base Address + 0x081C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:13]	–	Reserved	0
PICTURE_HEIGHT	[12:0]	RW	Specifies coded height of a picture (field or frame)	0

50.3.3.1.3 MFC_PROFILE

- Base Address: 0x1340_0000
- Address = Base Address + 0x0830, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
LEVEL	[15:8]	RW	Specifies Level in MPEG4 and H.264 NOTE: In H.264, 31 stands for level 3.1 and 9 stands for level 1b. In MPEG4, 3 stands for level 3, 7 stands for level 3b, and 9 stands for level 0b.	0
RSVD	[7:6]	–	Reserved	0
PROFILE	[5:0]	RW	Specifies MPEG4 and H.264 Profile [0]: MPEG4_PROFILE 0 = Simple profile 1 = Advanced simple profile [1:0]: H.264_Profile 0 = Main profile 1 = High profile 2 = Baseline profile NOTE: You should set the unspecified bits to 0.	0

50.3.3.1.4 MFC_PICTURE_STRUCT

- Base Address: 0x1340_0000
- Address = Base Address + 0x083C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
FIELD	[0]	RW	Field value for H.264 and MPEG4 0 = Frame picture only 1 = Field picture	0

50.3.3.1.5 MFC_LF_CONTROL

- Base Address: 0x1340_0000
- Address = Base Address + 0x0848, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0
LF_CONTROL	[1:0]	RW	Loop Filter for H.264 [1:0]: Loop filter disable indicator corresponds to disable_deblocking_filter_idc. 0 = Enables deblocking filter 1 = Disables deblocking filter 2 = Disables at slice boundary Loop Filter for MPEG4 [1]: Reserved [0]: Deblocking filter enable (post filter) 0 = Disables deblocking filter 1 = Enables deblocking filter	0

NOTE:

1. Both encoders and decoders use the MFC_LF_CONTROL.
2. The MFC_LF_CONTROL register is not effective for MPEG4 encoder.
3. The MFC_LF_CONTROL register returns disable_deblocking_filter_idc from bit stream for H.264 decode.

50.3.3.1.6 MFC_LF_ALPHA_CONTROL

- Base Address: 0x1340_0000
- Address = Base Address + 0x084C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0
LF_ALPHA_OFF	[4:0]	RW	Specifies loop filter alpha offset for deblocking filter	0

50.3.3.1.7 MFC_LF_BETA_OFF

- Base Address: 0x1340_0000
- Address = Base Address + 0x0850, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0
LF_BETA_OFF	[4:0]	RW	Specifies loop filter beta offset for deblocking filter	0

50.3.3.1.8 QP MFC_QP_OFFSET

- Base Address: 0x1340_0000
- Address = Base Address + 0x0C30, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MFC_QP_OFFSET	[31:0]	RW	If MFC_QP_OUT_EN is set, then it stores the QP information from the offset value of Luma DPB area address. NOTE: The unit of the offset is a double word (64 bits).	0

50.3.3.1.9 MFC_QP_OUT_EN

- Base Address: 0x1340_0000
- Address = Base Address + 0x0C34, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
MFC_QP_OUT_EN	[0]	RW	If it enables the MFC_QP_OUT_EN, then it stores the quantization value for each macro-block in the Luma DPB area. 0 = Disables QP OUT 1 = Enables QP OUT	0

You can calculate the address as shown in the *Example 50-1*.

Example 50-1 Address Calculation

```
x_pos = [0 ... (img_hsize_mb-1)];
y_pos = frame ? [0 ... (img_vsize_mb - 1)]:
        top ? [0, 2, 4 ... (img_vsize_mb × 2 - 2)]:
              [1, 3, 5 ... (img_vsize_mb × 2 - 1)];

<I_XSIZE >
if (img_hsize_mb < 64) I_XSIZE = 64
else if ((img_hsize_mb & 0x3f)! = 0)        I_XSIZE = ((img_hsize_mb >> 6) <<6) + 64
else                                        I_XSIZE = img_hsize_mb

<I_XSIZE >
if (frame) begin
      if(img_vsize_mb < 32)                 I_YSIZE =  32 else if((img_vsize_mb & 0x1f) ! = 0)
                                            I_YSIZE = ((img_vsize_mb>>5) << 5) + 32 + 32
      else                                  I_YSIZE = img_vsize_mb + 32
end
else begin
      if(img_vsize_mb < 16)                 I_YSIZE =  32
      else if((img_vsize_mb & 0x1f) ! = 0)  I_YSIZE = ((img_vsize_mb >> 4) << 5) + 32 + 32
      else                                  I_YSIZE = (img_vsize_mb << 1) + 32
end
pixel_x_m1 = I_XSIZE - 1 ;
pixel_y_m1 = I_YSIZE - 1 ;
roundup_x = ((pixel_x_m1)/16/8 + 1) ;
roundup_y = ((pixel_x_m1)/16/4 + 1) ;

x_addr = x_pos/4;
linear_addr0 = (((y_pos  & 0x1f) <<4) |(x_addr & 0xf)) << 2 ;
linear_addr1 = (((y_pos >> 6) & 0xff) × roundup_x + ((x_addr >> 5) & 0x7f)) ;



if(((x_addr >> 5) & 0x1) ==  ((y_pos >> 5) & 0x1))
                    bank_addr = ((x_addr >> 4) & 0x1);
else
                    bank_addr = 0x2 | ((x_addr >> 4) & 0x1);


physical_addr = DRAM_BASE + DPB_OFFSET + QP_OFFSET + (linear_addr1 << 13) | (bank_addr << 11) |
linear_addr0 ;
qp_save_range = (pixel_y_minus[5] == 0) ? pixel_y_minus[14:6] × roundup_x + pixel_x_minus[14:8] + 1 :
                                roundup_x × roundup_y;

NOTE: QP values are set to zero for I_PCM macroblocks in H.264 and skipped macroblocks in VC1.
             Host should allocate a physical memory size as:
             Memory size = ALIGN (img_hsize_mb, 64) × (ALIGN(img_vsize_mb, 32) + 32)
      Note that qp_save_range above specifies a virtual address area.
```

Figure 50-5 illustrates how to allocate the Memory.

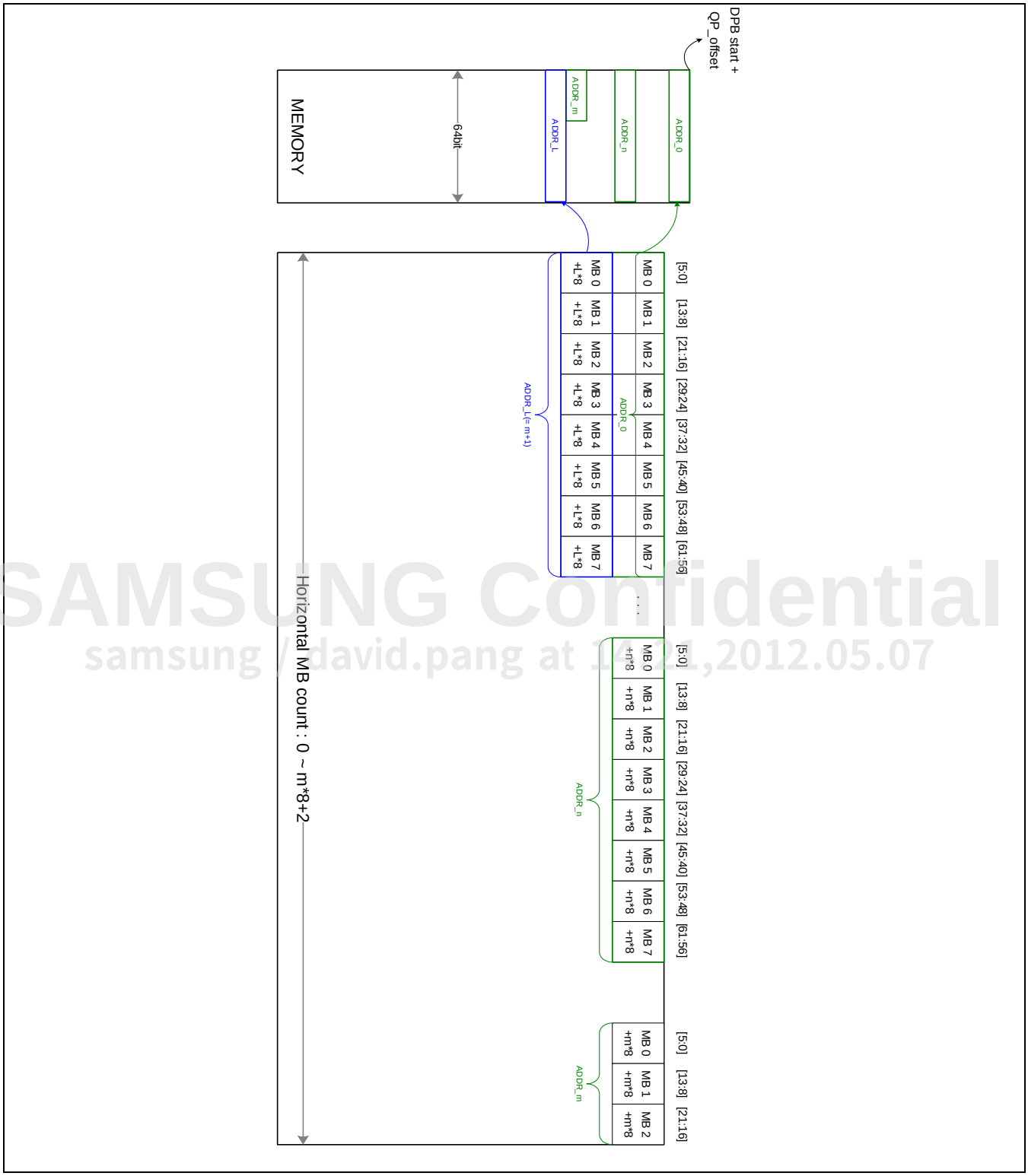


Figure 50-5 Memory Allocation

50.3.3.2 Channel and Stream Interface Registers

There are two sets of channels that communicate between host and MFC. Each channel consists of channel and stream interface registers. One channel waits for response from MFC that use MFC_SI_RTN_CHID and 15 MFC_COMMON_SI_RG registers. The other channel uses command from the host through MFC_SI_CH_INST_ID register and 15 MFC_COMMON_CHx_RG registers.

50.3.3.2.1 MFC_SI_RTN_CHID

- Base Address: 0x1340_0000
- Address = Base Address + 0x2000, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RTN_CHID	[31:0]	RW	Return Channel Instance ID Register The ID identifies the channel whose operation completes.	0

50.3.3.2.2 MFC_COMMON_SI_RG_n (n = 1 to 15)

- Base Address: 0x1340_0000
- Address = Base Address + 0x0204, Reset Value = 0x0000_0000
- Address = Base Address + 0x0208, Reset Value = 0x0000_0000
- Address = Base Address + 0x020C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0210, Reset Value = 0x0000_0000
- Address = Base Address + 0x0214, Reset Value = 0x0000_0000
- Address = Base Address + 0x0218, Reset Value = 0x0000_0000
- Address = Base Address + 0x021C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0220, Reset Value = 0x0000_0000
- Address = Base Address + 0x0224, Reset Value = 0x0000_0000
- Address = Base Address + 0x0228, Reset Value = 0x0000_0000
- Address = Base Address + 0x022C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0230, Reset Value = 0x0000_0000
- Address = Base Address + 0x0234, Reset Value = 0x0000_0000
- Address = Base Address + 0x0238, Reset Value = 0x0000_0000
- Address = Base Address + 0x023C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MFC_CH_COMMON_SI_RG_1 to 15	[31:0]	RW	Refer to the sections *50.3.3.3 Decoder Channel and Stream Interface Registers* and *50.3.3.4, Encoder Channel and Stream Interface Registers* for specific meaning of each register.	0

NOTE: The registers with offset values from 0x2040 to 0x207C share the same functionality as those registers whose offset values vary from 0x2080 to 0x20BC.

Channel 0 uses registers with offset values from 0x2040 to 0x207C. Channel 1 uses registers with offset values from 0x2080 to 0x20BC.

50.3.3.2.3 MFC_SI_CH0_INST_ID

- Base Address: 0x1340_0000
- Address = Base Address + 0x2040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:19]	–	Reserved	0
CH_DEC_TYPE	[18:16]	RW	Channel Decode Type [1:0]: CH0 control 1 = SEQ_START (sequence header processing) 2 = FRAME_START (frame decoding/encoding) 3 = LAST_SEQ (last frame decoding/encoding) 4 = INIT_BUFFERS (buffer initialization, Decoder only) 5 = FRAME_START_REALLOC (frame decoding for resolution change) 6 = FRAME_BATCH START (frame batch encoding)	
CH_INST_ID	[15:0]	RW	Specifies channel Instance ID for codec	

50.3.3.2.4 MFC_SI_CH1_INST_ID

- Base Address: 0x1340_0000
- Address = Base Address + 0x2080, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:19]	–	Reserved	0
CH_DEC_TYPE	[18:16]	RW	Channel Decode Type [1:0]: CH1 control 1 = SEQ_START (sequence header processing) 2 = FRAME_START (frame decoding/encoding) 3 = LAST_SEQ (last frame decoding/encoding) 4 = INIT_BUFFERS (buffer initialization. Decoder only) 5 = FRAME_START_REALLOC (frame decoding for resolution change) 6 = FRAME_BATCH START (frame batch encoding)	
CH_INST_ID	[15:0]	RW	Specifies channel Instance ID for codec	

50.3.3.2.5 MFC_COMMON_CH0_RG_n (n = 0 to 15)

- Base Address: 0x1340_0000
- Address = Base Address + 0x2044, Reset Value = 0x0000_0000
- Address = Base Address + 0x2048, Reset Value = 0x0000_0000
- Address = Base Address + 0x2048, Reset Value = 0x0000_0000
- Address = Base Address + 0x204C, Reset Value = 0x0000_0000
- Address = Base Address + 0x2050, Reset Value = 0x0000_0000
- Address = Base Address + 0x2054, Reset Value = 0x0000_0000
- Address = Base Address + 0x2058, Reset Value = 0x0000_0000
- Address = Base Address + 0x205C, Reset Value = 0x0000_0000
- Address = Base Address + 0x2060, Reset Value = 0x0000_0000
- Address = Base Address + 0x2064, Reset Value = 0x0000_0000
- Address = Base Address + 0x2068, Reset Value = 0x0000_0000
- Address = Base Address + 0x206V, Reset Value = 0x0000_0000
- Address = Base Address + 0x2070, Reset Value = 0x0000_0000
- Address = Base Address + 0x2074, Reset Value = 0x0000_0000
- Address = Base Address + 0x2078, Reset Value = 0x0000_0000
- Address = Base Address + 0x207C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MFC_COMMON_CH0_RG_1 to 15	[31:0]	RW	Common CH0 Register 1 to 15 Host sets parameters through these registers. Refer to the sections *50.3.3.3 Decoder Channel and Stream Interface Registers* and *50.3.3.4, Encoder Channel and Stream Interface Registers*, for specific meaning of each register.	0

50.3.3.2.6 MFC_COMMON_CH1_RG_n (n = 1 to 15)

- Base Address: 0x1340_0000
- Address = Base Address + 0x2084, Reset Value = 0x0000_0000
- Address = Base Address + 0x2088, Reset Value = 0x0000_0000
- Address = Base Address + 0x208C, Reset Value = 0x0000_0000
- Address = Base Address + 0x2090, Reset Value = 0x0000_0000
- Address = Base Address + 0x2094, Reset Value = 0x0000_0000
- Address = Base Address + 0x2098, Reset Value = 0x0000_0000
- Address = Base Address + 0x209C, Reset Value = 0x0000_0000
- Address = Base Address + 0x20A0, Reset Value = 0x0000_0000
- Address = Base Address + 0x20A4, Reset Value = 0x0000_0000
- Address = Base Address + 0x20A8, Reset Value = 0x0000_0000
- Address = Base Address + 0x20AC, Reset Value = 0x0000_0000
- Address = Base Address + 0x20B0, Reset Value = 0x0000_0000
- Address = Base Address + 0x20B4, Reset Value = 0x0000_0000
- Address = Base Address + 0x20B8, Reset Value = 0x0000_0000
- Address = Base Address + 0x20BC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MFC_COMMON_CH1_RG_1 to 15	[31:0]	RW	Common CH1 Register 1 to 15 When MFC is handling the request on CH0, the host communicates with MFC over CH1. MFC_COMMON_CH1_RG and MFC_COMMON_CH0_RG registers hold the same meaning.	0

50.3.3.3 Decoder Channel and Stream Interface Registers

50.3.3.3.1 MFC_COMMON_SI_RG_1

- Base Address: 0x1340_0000
- Address = Base Address + 0x2004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
VER_RESOL	[31:0]	R	Specifies vertical resolution of current sequence or picture for display It should read vertical resolution after decoding sequence header or a frame in case of resolution change.	0

50.3.3.3.2 MFC_COMMON_SI_RG_2

- Base Address: 0x1340_0000
- Address = Base Address + 0x2008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
HOR_RESOL	[31:0]	R	Specifies horizontal resolution of current sequence or picture for display It should read horizontal resolution after decoding sequence header or after decoding a frame in case of resolution change.	0

50.3.3.3.3 MFC_COMMON_SI_RG_3

- Base Address: 0x1340_0000
- Address = Base Address + 0x200C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MIN_NUM_DPB	[31:0]	R	Specifies required decoded picture buffer number After decoding sequence header, MFC sets the minimum number of required DPB buffers.	0

NOTE:

1. H.264: MaxDpbFrames + 2 (MaxDpbFrames is defined in the standard)
2. H.263: 3
3. Other codecs: 4
4. The number of reference buffers, one decoding buffer, and one display buffer determine the above values.

50.3.3.3.4 MFC_COMMON_SI_RG_4

- Base Address: 0x1340_0000
- Address = Base Address + 0x2010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DISPLAY_Y_ADR	[31:0]	R	Specifies display luminance address in display order	0

50.3.3.3.5 MFC_COMMON_SI_RG_5

- Base Address: 0x1340_0000
- Address = Base Address + 0x2014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DISPLAY_C_ADR	[31:0]	R	Specifies display chrominance address in display order	0

50.3.3.3.6 MFC_COMMON_SI_RG_6

- Base Address: 0x1340_0000
- Address = Base Address + 0x2018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MFC_DEC_FRM_SIZE	[31:0]	R	Specifies consumed number of bytes to decode a frame	0

50.3.3.3.7 MFC_COMMON_SI_RG_7

- Base Address: 0x1340_0000
- Address = Base Address + 0x201C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	0
DISPLAY_STATUS	[5:0]	R	Displays the status of decoded picture [5:4]: Resolution change 0 = No change 1 = Increases resolution 2 = Decreases resolution [3]: Progressive/interlace 0 = Progressive frame. 1 = Interlace frame [2:0]: Display status 0 = Decoding only (no display) 1 = Decoding and display. 2 = Display only. 3 = Empty DPB and decode completes	0

50.3.3.3.8 MFC_COMMON_SI_RG_8

- Base Address: 0x1340_0000
- Address = Base Address + 0x2020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0
FRAME_TYPE	[2:0]	R	Returns frame type of the decoded frame 0 = Not coded frame (skipped frame) 1 = I frame 2 = P frame 3 = B frame 4 = Others	0

50.3.3.3.9 MFC_COMMON_SI_RG_9

- Base Address: 0x1340_0000
- Address = Base Address + 0x2024, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DECODE_Y_ADR	[31:0]	R	Specifies luminance address in the order of decode	0

50.3.3.3.10 MFC_COMMON_SI_RG_10

- Base Address: 0x1340_0000
- Address = Base Address + 0x2028, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DECODE_C_ADR	[31:0]	R	Specifies chrominance address in the order of decode	0

50.3.3.3.11 MFC_COMMON_SI_RG_11

- Base Address: 0x1340_0000
- Address = Base Address + 0x202C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	0
DECODE_STATUS	[3:0]	R	Specifies status of decoded picture [3]: Progressive/interlace 0 = Progressive frame 1 = Interlace frame [2:0]: Decoding status 0 = Decoding only (no display) 1 = Decoding and display 2 = Display only 3 = Empty DPB and decode completes4 = No decoding and no display (equivalent to DISPLAY_STATUS=2)	0

50.3.3.3.12 MFC_COMMON_CHx_RG_1

- Base Address: 0x1340_0000
- Address = Base Address + 0x2044 or 2084, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CH_ES_ADDR	[31:0]	RW	Start address of the CPB of the elementary stream to be decoded	0

NOTE: The address should be in Port_A. It means that the address should lie between MC_DRAMBASE_ADDR_A and (MC_DRAMBASE_ADDR_A + 256MB).

50.3.3.3.13 MFC_COMMON_CHx_RG_2

- Base Address: 0x1340_0000
- Address = Base Address + 0x2048 or 2088, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CH_ES_DEC_UNIT_SIZE	[31:0]	RW	Decoding Unit Size in the CPB	0

50.3.3.3.14 MFC_COMMON_CHx_RG_3

- Base Address: 0x1340_0000
- Address = Base Address + 0x204C or 208C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CH_DESC_ADDR	[1:0]	RW	Channel Descriptor Buffer Address	0

NOTE: The address should be in Port_A (that is, the address should be between MC_DRAMBASE_ADDR_A and C_DRAMBASE_ADDR_A + 256MB).

50.3.3.3.15 MFC_COMMON_CHx_RG_4

- Base Address: 0x1340_0000
- Address = Base Address + 0x2050 or 2090, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:0]	–	Reserved	0

50.3.3.3.16 MFC_COMMON_CHx_RG_5

- Base Address: 0x1340_0000
- Address = Base Address + 0x2054 or 2094, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:0]	–	Reserved	0

50.3.3.3.17 MFC_COMMON_CHx_RG_6

- Base Address: 0x1340_0000
- Address = Base Address + 0x2058 or 2098, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CPB_SIZE	[31:0]	RW	CPB size register It should be set before SEQ_START and FRAME_START. The maximum CPB size is 4 MB.	0

NOTE: CPB_SIZE = align (CH_ES_DEC_UNIT_SIZE + 64, pow (2 KB)) for H.264 and VC1 decoders when it enables DMX, where pow (2 KB) = 1 KB, 2 KB, 4 KB, 8 KB, …, 4 MB.

50.3.3.3.18 MFC_COMMON_CHx_RG_7

- Base Address: 0x1340_0000
- Address = Base Address + 0x205C or 209C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
DESC_SIZE	[31:0]	RW	Descriptor Buffer Size Register This register should be set before SEQ_START and FRAME_START. The maximum descriptor buffer size is 128 KB.	0

50.3.3.3.19 MFC_COMMON_CHx_RG_8

- Base Address: 0x1340_0000
- Address = Base Address + 0x2060 or 20A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RELEASE_BUFFER	[31:0]	RW	Release Buffer Register 1 = Free 0 = Busy It specifies the availability of each DPB. The nth bit specifies the availability of the nth DPB.	0

50.3.3.3.20 MFC_COMMON_CHx_RG_9

- Base Address: 0x1340_0000
- Address = Base Address + 0x2064 or 20A4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
HOST_WR_ADR	[31:0]	RW	The address points to a space of shared memory that consists of multiple commands which host can Read/Write. Refer to the 1.4 Shared Memory for more information on Shared Memory.	0

50.3.3.3.21 MFC_COMMON_CHx_RG_10

- Base Address: 0x1340_0000
- Address = Base Address + 0x2068 or A8 Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SLICE_IF_ENABLE	[31]	RW	Enable Slice Interface for decode 0 = Disables slice interface 1 = Enables slice interface	0
CONFIG_DELAY_ENABLE	[30]	RW	Enable Configurable Display Delay for H.264 decode 0 = Disables configurable display delay 1 = Enables configurable display delay	0
DISPLAY_DELAY	[29:16]	RW	Specifies number of frames for Display Delay MFC returns frames for display even if DPB is not filled. It is valid for H.264 decoder only.	0
DMX_DISABLE	[15]	RW	Host generates descriptor information on behalf of MFC demux 0 = Enables demux so that MFC generates descriptor information 1 = Disables demux so that host generates descriptor information This register is valid for H.264 and VC1 decoders only.	–
DPB_FLUSH	[14]	RW	Flush DPB that discards all output buffers in DPB 0 = Normal operation 1 = Flushes DPB	–
NUM_DPB	[13:0]	RW	Specifies number of DPB that host prepares for decode	0

NOTE:

1. When it disables demux, the host will fill the descriptor buffer for each slice/NALU. For VC1, after constructing all the descriptors create one dummy descriptor. The dummy descriptor will contain a start code suffix byte 0x82 (ID [7:0] = 0x82). This byte follows a zero word that indicates the end of descriptors.
2. In case of H264 and VC1, all descriptor entries including VC1 dummy descriptor entry, a zero word written to the descriptor table indicates the end of descriptors.

ID[7:0]	Offset[2:0]	Start_addr[20:0]
9'd0		Unit_size[22:0]
32'd0		
32'd0		

ID: start code suffix byte (first byte after start code: NALU header byte for H.264 and 0x82 for VC1)
Offset: (nal start address) and 0x07
Start_addr: ((nal start address) >> 3) << 1
Unit_size: size of NALU

50.3.3.3.22 MFC_COMMON_CHx_RG_11

- Base Address: 0x1340_0000
- Address = Base Address + 0x206C or 20AC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CMD_SEQ_NUM	[31:0]	RW	Command Sequence Number from the host	0

50.3.3.4 Encoder Channel and Stream Interface Registers

50.3.3.4.1 MFC_COMMON_SI_RG_1

- Base Address: 0x1340_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ENC_STREAM_SIZE	[31:0]	R	Specifies encoded stream size in byte count	0

50.3.3.4.2 MFC_COMMON_SI_RG_2

- Base Address: 0x1340_0000
- Address = Base Address + 0x2008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ENC_PICTURE_CNT	[31:0]	R	Encoded Picture Count For interlaced streams the picture count increments field by field.	0

50.3.3.4.3 MFC_COMMON_SI_RG_3

- Base Address: 0x1340_0000
- Address = Base Address + 0x200C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
WRITE_POINTER	[31:0]	R	Stream Buffer Write Pointer EDFU updates external memory address at the end of encoding a frame.	0

50.3.3.4.4 MFC_COMMON_SI_RG_4

- Base Address: 0x1340_0000
- Address = Base Address + 0x2010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ENC_SLICE_TYPE	[31:0]	RW	Specifies Slice Type 0 = Not coded frame 1 = I frame 2 = P frame 3 = B frame 4 = Skipped frame 5 = Others	0

50.3.3.4.5 MFC_COMMON_SI_RG_5

- Base Address: 0x1340_0000
- Address = Base Address + 0x2014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ENCODED_Y_ADDR	[31:0]	RW	Specifies address of encoded luminance picture	0

50.3.3.4.6 MFC_COMMON_SI_RG_6

- Base Address: 0x1340_0000
- Address = Base Address + 0x2018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
ENCODED_C_ADDR	[31:0]	RW	Specifies address of encoded chrominance picture	0

50.3.3.4.7 MFC_COMMON_CHx_RG_1

- Base Address: 0x1340_0000
- Address = Base Address + 0x2044 or 0x2088, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CH_SB_ADDR	[31:0]	RW	Specifies start address of stream buffer at encoder	0

NOTE: The address should be in Port_A. It means that the address should lie between MC_DRAMBASE_ADDR_A and MC_DRAMBASE_ADDR_A+256MB).

50.3.3.4.8 MFC_COMMON_CHx_RG_3

- Base Address: 0x1340_0000
- Address = Base Address + 0x204C or 0x208C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
BUFFER_SIZE	[31:0]	RW	Specifies buffer size of encoded stream	0

50.3.3.4.9 MFC_COMMON_CHx_RG_4

- Base Address: 0x1340_0000
- Address = Base Address + 0x2050 or 0x2090, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CURRENT_Y_ADDR	[31:0]	RW	Specifies address of current luminance picture to encode	0

NOTE: The address should be in Port_B. It means that the address should lie between MC_DRAMBASE_ADDR_B and (MC_DRAMBASE_ADDR_B + 256MB).
Align the buffer address as:
Tile mode: align (align (x_size, 128) × y_size, 8192)
Linear mode: align (align (x_size, 16) × y_size, 2048)

50.3.3.4.10 MFC_COMMON_CHx_RG_5

- Base Address: 0x1340_0000
- Address = Base Address + 0x2054 or 0x2094, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CURRENT_C_ADDR	[31:0]	RW	Specifies address of current chrominance picture to encode	0

NOTE: The address should be in Port_B. It means that the address should lie between MC_DRAMBASE_ADDR_B and (MC_DRAMBASE_ADDR_B + 256MB).
Align the buffer address as:
Tile mode: align (align (x_size, 128) × y_size/2, 8192)
Linear mode: align (align (x_size, 16) × y_size/2, 2048)

50.3.3.4.11 MFC_COMMON_CHx_RG_6

- Base Address: 0x1340_0000
- Address = Base Address + 0x2058 or 0x2098, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:0]	–	Reserved	0
NOT_CODED	[1]	RW	Encodes current frame into a not coded frame	0
I_FRAME	[0]	RW	Encodes current frame into an I frame. It starts being into effect only from the next anchor frame.	0

50.3.3.4.12 MFC_COMMON_CHx_RG_9

- Base Address: 0x1340_0000
- Address = Base Address + 0x2064 or 0x20A4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
HOST_WR_ADR	[31:0]	RW	Specifies address that point to a space of shared memory which consists of multiple commands. The commands enable the host to Read/Write. Refer to Section *50.4 Shared Memory Interface* for detailed structure of the shared memory.	0

50.3.3.4.13 MFC_COMMON_CHx_RG_10

- Base Address: 0x1340_0000
- Address = Base Address + 0x2068 or 0x20A8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:15]	–	Reserved	0
INPUT_BUFFER _FLUSH	[14]	RW	Specifies flushing of input buffer that discards all frame in input buffer 0 = Normal operation 1 = Flushes input buffer	0
RSVD	[13:0]	–	Reserved	0

50.3.3.4.14 MFC_COMMON_CHx_RG_11

- Base Address: 0x1340_0000
- Address = Base Address + 0x206C or 0x20AC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
CMD_SEQ_NUM	[31:0]	RW	Specifies command sequence number from host.	0

50.3.4 Encoding Registers

50.3.4.1 Common Encoder Register

50.3.4.1.1 ENC_PIC_TYPE_CTRL

- Base Address: 0x1340_0000
- Address = Base Address + 0xC504, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:19]	–	Reserved	0
ENC_PIC_TYPE_ENABLE	[18]	RW	Specifies enable of ENC_PIC_TYPE_CTRL 0 = Disables ENC_PIC_TYPE_CTRL 1 = Enables ENC_PIC_TYPE_CTRL[17:0] for picture type setting	0
B_FRM_CTRL	[17:16]	RW	Specifies number of B frames 0 = The number of B frames is zero 1 = The number of B frames is one 2 = The number of B frames is two 3 = Reserved	0
I_FRM_CTRL	[15:0]	RW	Specifies P and I frames 0 = All P frames 1 = All I frames 2 = I – P – I – P 3 = I – P – P – I N = (N – 1) P frames between two I frames	0

50.3.4.1.2 ENC_B_RECON_WRITE_ON

- Base Address: 0x1340_0000
- Address = Base Address + 0xC508, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
B_RECON_ON	[0]	RW	Use this register for debugging. By default, it is set to zero. If it is set, it is required to allocate the required memory. Specifies recon data write at B-frame 0 = Disables recon data write at B-frame 1 = Enables recon data write at B-frame	0

NOTE: When it enables B_RECON_ON, host allocates B_FRAME_RECON_LUMA_ADDR (0x062C) and B_FRAME_RECON_CHROMA_ADDR (0x0630). The size should be:
Size of (B_FRAME_RECON_LUMA_ADDR) = align (align (x_size, 128) × align (y_size, 32), 8192)
Size of (B_FRAME_RECON_CHROMA_ADDR) = align (align (x_size, 128) × align (y_size/2, 32), 8192)

50.3.4.1.3 ENC_MSLICE_CTRL

- Base Address: 0x1340_0000
- Address = Base Address + 0xC50C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0
MSLICE_MODE	[2:1]	RW	Multi Slice Mode 0 = MB count does the multi slicing 1 = Byte count does the multi slicing	0
MSLICE_ENA	[0]	RW	Multi Slice Enable 0 = One slice per frame 1 = Enables multi slice or resync marker	0

50.3.4.1.4 ENC_MSLICE_MB

- Base Address: 0x1340_0000
- Address = Base Address + 0xC510, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
MSLICE_MB	[15:0]	RW	Specifies number of macroblocks in one slice. It is valid if MSLICE_MODE=0 and MSLICE_ENA = 1.	0

50.3.4.1.5 ENC_MSLICE_BYTE

- Base Address: 0x1340_0000
- Address = Base Address + 0xC514, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
MSLICE_BIT	[31:0]	RW	Specifies number of bit count in one slice. It holds valid if MSLICE_MODE=1 and MSLICE_ENA = 1.	0

NOTE: The minimum size of MSLICE_BIT is 1900 bits

50.3.4.1.6 ENC_CIR_CTRL

- Base Address: 0x1340_0000
- Address = Base Address + 0xC518, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
CIR_NUM	[15:0]	RW	Specifies number of intra refresh macroblocks	0

50.3.4.1.7 ENC_MAP_FOR_CUR

- Base Address: 0x1340_0000
- Address = Base Address + 0xC51C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0
ENC_MAP_FOR_CUR	[1:0]	RW	Specifies memory structure of current frame 0 = Linear mode 3 = 64x32 tiled mode	0

50.3.4.1.8 ENC_PADDING_CTRL

- Base Address: 0x1340_0000
- Address = Base Address + 0xC520, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PAD_CTRL_ON	[31]	RW	0 = Uses boundary pixel for current image padding in case that its image size is not a multiple of 16 1 = Uses ENC_PADDING_CTRL[23:0] for current image padding	0
RSVD	[30:24]	–	Reserved	0
CR_PAD_VAL	[23:16]	RW	Specifies value for original CR image padding when PAD_CTRL_ON is set to1.	0
CB_PAD_VAL	[15:8]	RW	Specifies value for original CB image padding when PAD_CTRL_ON is set to 1.	0
LUMA_PAD_VAL	[7:0]	RW	Specifies value for original Y image padding when PAD_CTRL_ON is set to 1.	0

50.3.4.1.9 NV21_SEL

- Base Address: 0x1340_0000
- Address = Base Address + 0xC548, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
NV21_SEL	[0]	RW	Specifies chroma interleaving order 0 = Cb, Cr, Cb, Cr 1 = Cr, Cb, Cr, Cb	0

50.3.4.1.10 ENC_COMMON_INTRA_BIAS

- Base Address: 0x1340_0000
- Address = Base Address + 0xC588, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
ENC_COMMON_INTRA_BIAS	[15:0]	RW	Use this register for intra mode in the weighted macroblock mode decision. Mode decision compares: 1) inter_cost + ENC_COMMON_INTRA_BIAS and 2) intra_cost. If inter_cost is zero, then it determines the mode to inter macroblock. To disable this option, you should set ENC_COMMON_INTRA_BIAS to 0.	0

50.3.4.1.11 ENC_COMMON_BI_DIRECT_BIAS

- Base Address: 0x1340_0000
- Address = Base Address + 0xC58C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
ENC_COMMON_BI_DIRECT_BIAS	[15:0]	RW	Use this register against the bi-directional mode in the weighted macroblock mode decision. Mode decision compares: 1) uni_direction_cost and 2) bi_direction_cost + ENC_COMMON_BI_DIRECT_BIAS. If the macroblock type is INTRA, then this register shows no effect. To disable this option, you should setENC_COMMON_BI_DIRECT_BIAS to 0. This register sets effective in MPEG4 encoding only.	0

50.3.4.2 Rate Control Register

50.3.4.2.1 RC_CONFIG

- Base Address: 0x1340_0000
- Address = Base Address + 0xC5A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	0
FR_RC_EN	[9]	RW	Frame Level Rate Control Enable 0 = Disables frame level rate control 1 = Enables frame level rate control	0
MB_RC_EN	[8]	RW	Macroblock Level Rate Control Enable 0 = Disables MB level rate control 1 = Enables MB level rate control It holds valid only for H.264.	0
RSVD	[7:6]	–	Reserved	0
FRAME_QP	[5:0]	RW	Frame Quantization Parameter (QP) Use FRAME_QP for the first macroblock QP in a frame. This value should be set in the range of MIN_QP to MAX_QP in RC_QBOUND. You can change the QP of the next macroblocks as per the value of FR_RC_EN and MB_RC_EN. The interpretation of FRAME_QP varies as per RC_CONFIG [9:8] as: <RC_CONFIG[9:8]> • 2'b00: Applies constant QP to all macroblocks. It uses FAME_QP for I frame. It uses P_FRAME_QP and B_FRAME_QP for P and B frames.2'b01: QP of the next macroblocks varies with macroblock adaptive scaling. But it does not take into account the size of generated bits. • 2'b10: You can change the QP of the next macroblocks through the difference between the numbers of target bit and generated bit during encoding a picture. But macroblock adaptive scaling does not get applied. • 2'b11: You can change the QP of the next macroblocks through the difference between the numbers of target and generated bit during encoding a picture. It also varies with macroblock adaptive scaling.	0

50.3.4.2.2 RC_FRAME_RATE

- Base Address: 0x1340_0000
- Address = Base Address + 0xD0D0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
FRAME_RATE	[31:0]	RW	Specifies frames per second in 1000x scale For example, 7500 stands for 7.5 frames/sec 0 remains a forbidden value. It holds valid only when it enables frame level RC	0

50.3.4.2.3 RC_BIT_RATE

- Base Address: 0x1340_0000
- Address = Base Address + 0xC5A8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
BIT_RATE	[31:0]	RW	Specifies bits per second 0 remains a forbidden value. It holds valid only when it enables Frame level RC	0

50.3.4.2.4 RC RC_QBOUND

- Base Address: 0x1340_0000
- Address = Base Address + 0xC5AC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	–	Reserved	0
MAX_QP	[13:8]	RW	Maximum OP The range is: H.264: 0 to 51 MPEG4, H.263: 1 to 31	0
RSVD	[7:6]	–	Reserved	0
MIN_QP	[5:0]	RW	Minimum QP The range is: H.264: 0 to 51 MPEG4, H.263: 1 to 31	0

NOTE: MAX_QP should be greater than or equal to MIN_QP.

50.3.4.2.5 RC_RPARA

- Base Address = 0x1340_0000
- Address = Base Address + 0xC5B0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	0
REACT_PARA	[15:0]	RW	Specifies rate control reaction coefficient. "0" remains a forbidden value.	0

NOTE:

1. It is valid only when it enables the frame level RC.
2. For tight CBR, this field should be small (for example, 2 to 10).
   For VBR, this field should be large (for example, 100 to 1000).
3. We do not recommend to use a number greater than FRAME_RATE × (10^9/BIT_RATE).

50.3.4.2.6 RC_MB_CTRL

- Base Address: 0x1340_0000
- Address = Base Address + 0xC5B4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31: 4]	–	Reserved	0
DARK_DISABLE	[3]	RW	Disable Dark Region Adaptive Feature 0 = Enables dark region adaptive feature 1 = Disables dark region adaptive feature QP of dark MB may not be smaller than frame QP although it is smooth, static or it has small activity	0
SMOOTH_DISABLE	[2]	RW	Disable Smooth Region Adaptive Feature 0 = Enables smooth region adaptive feature 1 = Disables smooth region adaptive feature QP of smooth MB may be smaller than frame QP	0
STATIC_DISABLE	[1]	RW	Disable Static Region Adaptive Feature 0 = Enables static region adaptive feature 1 = Disables static region adaptive feature QP of static MB may be smaller than frame QP.	0
ACT_DISABLE	[0]	RW	Disable MB Activity Adaptive Feature 0 = Enables MB activity adaptive feature 1 = Disables MB activity adaptive feature QP of MB that has small activity may be smaller than frame QP and QP of MB that has large activity may be larger than frame QP	0

NOTE: It is valid only when it enables H.264 and macroblock level.

50.3.4.3 H.264 Encoder Register

50.3.4.3.1 H264_ENC_ENTRP_MODE

- Base Address: 0x1340_0000
- Address = Base Address + 0xD004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
H264_ENC _ENTRP_MODE	[0]	RW	Entropy Register 0 = CAVLC 1 = CABAC	0

50.3.4.3.2 H264_ENC_NUM_OF_REF

- Base Address: 0x1340_0000
- Address = Base Address + 0xD010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:5]	–	Reserved	0
H264_ENC_P _NUM_OF_REF	[6:5]	RW	Specifies number of reference pictures of P-picture 1 = One reference frame 2 = Two reference frames	0
H264_ENC _NUM_OF_REF	[4:0]	RW	Specifies maximum number of reference pictures 1 = One reference frame 2 = Two reference frames	0

50.3.4.3.3 H264_ENC_TRANS_8X8_FLAG

- Base Address: 0x1340_0000
- Address = Base Address + 0xD034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
ENC_TRANS _8X8_FLAG	[0]	RW	Transform Enable Flag 0 = Disables flag 1 = Enables flag	0

50.3.4.3.4 H264_ENC_MB_INFO_ENABLE

- Base Address: 0x1340_0000
- Address = Base Address + 0xD140, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
H264_ENC_MB_INFO_ENABLE	[0]	RW	H.264 MB Information Dump Enable 0 = Disables info dump 1 = Enables info dump	0

50.3.4.4 MPEG4 Encoder Register

50.3.4.4.1 MPEG4_ENC_QUART_PXL

- Base Address: 0x1340_0000
- Address = Base Address + 0xE008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
MPEG4_QUART_PXL	[0]	RW	Quarter Pixel Search 0 = Disables quarter pixel search 1 = Enables quarter pixel search	0

50.4 Shared Memory Interface

MFC provides shared memory interface to exchange information with host. This also satisfies the growth in diverse requirements for codec features and additional enhancements of MFC firmware. Host gets returned parameters or set parameter values through shared memory. Since it exchanges parameter through external memory, there is no limitation on the number of fields. You can consider it as a register group whose physical memory gets allocated in external memory.

This section describes the shared memory interface, such as, shared memory allocation and shared memory structure that consists of multiple fields.

50.4.1 Host Interface

Host allocates shared memory and informs MFC of the buffer pointer through HOST_WR_ADR register. We recommend allocating shared memory before sequence header parses through SEQ_START. Since the number of fields in the shared memory is fixed in 0, host allocates a buffer in the external memory to accommodate them.

After it allocates shared memory, host and MFC will exchange information through shared memory. The host reads and writes to each field in shared memory at the defined byte offset.

The pseudo code that initializes shared memory is:

```
// shared mem allocation. Should be done before SEQ_START
shared_mem_ptr = (int *) malloc (SHARED_MEM_SIZE);
host_write_word(HOST_WR_ADR, shared_mem_ptr);
```


The pseudo code that reads and writes frame tags are:

```
// write frame_tag
host_write_word (shared_mem_ptr+ADR_SET_FRAE_TAG, frame_tag);

// read frame_tag
get_frame_tag = host_read_word (shared_mem_ptr+ADR_GET_FRAME_TAG_TOP);
```

NOTE: At the start of stream encoding/decoding, the host initializes all fields in shared memory that corresponds to the appropriate stream to avoid any undefined behaviours.

50.4.2 Shared Memory Structure

50.4.2.1 Decoding Control

50.4.2.1.1 EXTENDED_DECODE_STATUS (Extended Decode Status)

- Base Address: 0x1340_0000
- Address = Base Address + 0x0000, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	R	Reserved	–

50.4.2.1.2 SET_FRAME_TAG

- Base Address: 0x1340_0000
- Address = Base Address + 0x0004, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
SET_FRAME_TAG	[31:0]	W	Set Frame Tag Host sets a unique frame ID for output buffer For example, application specific timestamp.	–

50.4.2.1.3 GET_FRAME_TAG_TOP

- Base Address: 0x1340_0000
- Address = Base Address + 0x0008, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
GET_FRAME_TAG_TOP	[31:0]	R	Get Frame Tag Top MFC sets a frame tag for the output frame that corresponds to one set of host that uses SET_FRAME_TAG. Host reads a unique ID on display. This tag returns an application specific ID for the progressive frame. For an interlaced picture, this tag returns an application specific ID for top field. NOTE: The value (– 1) indicates that there is no displayable picture.	–

50.4.2.1.4 GET_FRAME_TAG_BOTTOM

- Base Address: 0x1340_0000
- Address = Base Address + 0x000C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
GET_FRAME_TAG_BOTTOM	[31:0]	R	Get Frame Tag Bottom MFC sets a frame tag for the output frame that corresponds to one set of host that uses SET_FRAME_TAG. Host reads the ID for the bottom field. This tag holds valid for the interlaced picture only. NOTE: The value (– 1) indicates that there is no displayable picture or a progressive picture.	–

50.4.2.1.5 PIC_TIME_TOP

- Base Address: 0x1340_0000
- Address = Base Address + 0x0010, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
PIC_TIME_TOP	[31:0]	R	Picture Time for Top Field The header information in the bit stream determines the presentation time of the output frame. For an interlaced picture, it returns the presentation time for top field. The time unit is msec. If the standard does not provide the presentation time, for example, H.263 and H.264, then it returns a sequence number.	–

NOTE: Specific interpretations for each standard are:

1. MPEG2, H.263: Temporal reference
2. MPEG4: VOP time
3. H.264: POC
4. VC1 simple/main profile: zero
5. VC1 advanced profile: Temporal reference frame counter. If it is not present, zero should be written.

50.4.2.1.6 PIC_TIME_BOTTOM

- Base Address: 0x1340_0000
- Address = Base Address + 0x0014, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
PIC_TIME_BOTTOM	[31:0]	R	Picture Time for Bottom Field Presentation time of the bottom field. It is valid for interlaced picture only. The specific interpretation for each standard remains same as that of PIC_TIME_TOP.	–

50.4.2.1.7 START_BYTE_NUM (Start Byte Number)

- Base Address: 0x1340_0000
- Address = Base Address + 0x0018, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
START_BYTE_NUM	[31:0]	RW	Start Byte Number Specifies an offset of stream when it is not aligned	–

50.4.2.1.8 CROP_INFO1 (Cropping Information One)

- Base Address: 0x1340_0000
- Address = Base Address + 0x0020, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
CROP_RIGHT_OFFSET	[31:16]	R	Specifies cropping right offset value	–
CROP_LEFT_OFFSET	[15:0]	R	Specifies cropping left offset value	–

50.4.2.1.9 CROP_INFO2 (Cropping Information Two)

- Base Address: 0x1340_0000
- Address = Base Address + 0x0024, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
CROP_BOTTOM_OFFSET	[31:16]	R	Specifies cropping bottom offset value	–
CROP_TOP_OFFSET	[15:0]	R	Specifies cropping top offset value	–

50.4.2.1.10 DISP_PIC_PROFILE (Profile info for displayed picture)

- Base Address: 0x1340_0000
- Address = Base Address + 0x007C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	–
DISP_PIC_LEVEL	[15:8]	R	Specifies display pixel level in MPEG4 and H.264 In H.264, 31 stands for level 3.1 and 9 stands for level 1b. In MPEG4, 3 stands for level 3 and 7 stands for level 3b. In VC1, simple and main profile defines: 0 = Low 2 = Medium 4 = High Advanced profile defines: 0 = Level 0 1 = Level 1 2 = Level 2 3 = Level 3 4 = Level 4 In MPEG2, defined levels are: 4 = High 6 = High 1440 8 = Main 10 = Low	–
RSVD	[7:5]	–	Reserved	–
DISP_PIC_PROFILE	[4:0]	R	In MPEG4, the display picture profile defined are: 0 = SP 1 = ASP In H.264, the display picture profile defined are: 0 = Baseline 1 = Main 2 = High In H.263, the display picture profile defined are: 0 = Bit stream does not carry profile information) In VC1, the display picture profile defined are: 0 = Simple 1 = Main 2 = Advanced In MPEG2, the display picture profile defined are: 4 = Main 5 = Simple	–

50.4.2.1.11 H264_POC_TYPE

- Base Address: 0x1340_0000
- Address = Base Address + 0x00B8, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
H264_POC_TYPE	[31:0]	R	POC_TYPE of H.264 Decoder If POC_TYPE is 0 or 1, then MaxDpbSize in the standard determines the initial display delay If POC_TYPE is 2, then there is no initial display delay	–

NOTE: The two reasons for which the actual display delay differs are:

1. If there are multiple SPS's, then MFC returns the POC_TYPE of the first SPS. If the first frame uses a different SPS, then it results in different initial display delay.
2. If the second IDR frame appears within MaxDpbSize, then MFC starts the DPB buffer flushing which causes a reduced initial display delay.

50.4.2.1.12 DISP_PIC_FRAME_TYPE

- Base Address: 0x1340_0000
- Address = Base Address + 0x00C0, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
DISP_PIC_FRAME_TYPE	[31:0]	R	Specifies frame type of displayed picture 0 = Not coded frame 1 = I frame 2 = P frame 3 = B frame	–

50.4.2.1.13 FREE_LUMA_DPB

- Base Address: 0x1340_0000
- Address = Base Address + 0x00C4, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
FREE_LUMA_DPB	[31:0]	R	Specifies free luma DPB address Host copies DPB to the free DPB address. Use the free DPB to handle frames not coded in VC1 and MPEG4 decoders.	–

50.4.2.1.14 ASPECT_RATIO_INFO

- Base Address: 0x1340_0000
- Address = Base Address + 0x00C8, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	–
ASPECT_RATIO_INFO	[3:0]	R	Aspect Ratio Information 1 = Square (1:1) 2 = 625 type for 4:3 picture (12:11) 3 = 525 type for 4:3 picture (10:11) 4 = 625 type stretched for 16:9 picture (16:11) 5 = 525 type stretched for 16:9 picture (40:33) 15 = Extended PAR	–

50.4.2.1.15 EXTENDED_PAR

- Base Address: 0x1340_0000
- Address = Base Address + 0x00CC, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
PAR_WIDTH	[31:16]	R	The value indicates horizontal size of the sample aspect ratio. This holds valid only if ASPECT_RATIO_INFO = 15.	–
PAR_HEIGHT	[15:0]	R	The value indicates vertical size of the sample aspect ratio. This holds valid only if ASPECT_RATIO_INFO = 15.	–

50.4.2.2 Encoding Control

50.4.2.2.1 EXT_ENC_CONTROL

- Base Address: 0x1340_0000
- Address = Base Address + 0x0028, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
VBV_BUFFER_SIZE	[31:16]	W	Specifies VBV Buffer Size Host defines the VBV buffer size. The unit of buffer size is KB. Actual buffer size= VBV_BUFFER_SIZE × 1024 bytes It holds valid only when FRAME_SKIP_ENABLE = 2.	–
ASPECT_RATIO _VUI_ENABLE	[15]	W	Specifies Aspect Ratio VUI Enable 0 = Disables aspect ratio VUI in H.264 encoding 1 = Enables aspect ratio VUI in H.264 encoding	–
RSVD	[14:5]	–	Reserved	–
HIERARCHICAL _P_ENABLE	[4]	W	Specifies Hierarchical P Frame Enable 0 = Disables hierarchical P frame 1 = Enables hierarchical P frame	–
SEQ_HEADER _CONTROL	[3]	W	Specifies Sequence Header Control 0 = Does not generate sequence header on the first FRAME_START 1 = Generates sequence header on both SEQ_START and first FRAME_START	–
FRAME_SKIP _ENABLE	[2:1]	W	Specifies Frame Skip Enable 0 = Disables frame skip 1 = Enables frame skip that uses maximum buffer size which is level-defined 2 = Enables frame skip that uses VBV_BUFFER_SIZE which is HOST defined The chance of the rate overshoot increases when the generate bit rate is high.	–
HEC_ENABLE	[0]	W	Specifies HEC Enable 0 = Disables Header Extension Code (HEC) in MPEG4 encoding 1 = Enables HEC	–

50.4.2.2.2 ENC_PARAM_CHANGE

- Base Address: 0x1340_0000
- Address = Base Address + 0x002C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	–
RC_BIT_RATE_ CHANGE	[2]	W	Specifies RC Bit Range Change 0 = Normal operation 1 = Changes target bit rate	–
RC_FRAME_RATE_ CHANGE	[1]	W	Specifies RC Frame Range Change 0 = Normal operation 1 = Changes target frame rate	–
I_PERIOD_CHANGE	[0]	W	Specifies I Period Change 0 = Normal operation 1 = Changes GOP size	–

50.4.2.2.3 VOP_TIMING

- Base Address: 0x1340_0000
- Address = Base Address + 0x0030, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
VOP_TIMING_ENABLE	[31]	W	Enables computing vop_time_increment and modulo_time_base in MPEG4	–
VOP_TIME _RESOLUTION	[30:16]	W	Computes vop_time_increment and modulo_time_base in MPEG4	–
FRAME_DELTA	[15:0]	W	Computes vop_time_increment and modulo_time_base in MPEG4	–

50.4.2.2.4 HEC_PERIOD

- Base Address: 0x1340_0000
- Address = Base Address + 0x0034, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
HEC_PERIOD	[31:0]	W	HEC Period Specifies insertion of a HEC for every HEC_PERIOD number of packets in MPEG4 encoding	–

50.4.2.2.5 _B_FRAME_QP

- Base Address: 0x1340_0000
- Address = Base Address + 0x0070, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	–
B_FRAME_QP	[11:6]	W	Use this value for B frame QP	–
P_FRAME_QP	[5:0]	W	Use this value for P frame QP	–

NOTE: The frame QPs are valid only if FR_RC_EN = 0 and MB_RC_EN = 0

50.4.2.2.6 ASPECT_RATIO_IDC

- Base Address: 0x1340_0000
- Address = Base Address + 0x0074, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
ASPECT _RATIO_IDC	[7:0]	W	VUI Aspect Ratio IDC for H.264 encoding VUI Table E-1 defines values in the standard. It holds valid only if ASPECT_RATIO_VUI_ENABLE = 1.	–

50.4.2.2.7 EXTENDED_SAR

- Base Address: 0x1340_0000
- Address = Base Address + 0x0078, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
SAR_WIDTH	[31:16]	W	The value indicates the horizontal size of sample aspect ratio. It holds valid only if ASPECT_RATIO_VUI_ENABLE = 1.	–
SAR_HEIGHT	[15:0]	W	The value indicates the vertical size of sample aspect ratio. It holds valid only if ASPECT_RATIO_VUI_ENABLE = 1.	–

50.4.2.2.8 NEW_RC_BIT_RATE

- Base Address: 0x1340_0000
- Address = Base Address + 0x0090, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
NEW_RC _BIT_RATE	[31:0]	W	New RC Bit Rate Updates target bit rate at encoder which has the same format as RC_BIT_RATE (0xC5A8). This holds valid only if RC_BIT_RATE_CHANGE = 1.	–

50.4.2.2.9 NEW_RC_FRAME_RATE

- Base Address: 0x1340_0000
- Address = Base Address + 0x0094, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
NEW_RC _FRAME_RATE	[31:0]	W	New RC Frame Rate Updates target frame rate at encoder which has the same format as RC_FRAME_RATE. The unit of RC_FRAME_RATE is frames per second in 1000x scale. It holds valid only if RC_FRAME_RATE_CHANGE = 1.	–

50.4.2.2.10 NEW_I_PERIOD

- Base Address: 0x1340_0000
- Address = Base Address + 0x0098, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
NEW_I_PERIOD	[31:0]	W	New Intra Period Updates intra period at encoder which has the same format as I_FRM_CTRL (0xC504). 0 = All P frames 1 = All I frames 2 = I – P – I – P 3 = I – P – P – I N = (N – 1) P frames between two I frames This holds valid only if I_PERIOD_CHANGE = 1.	–

50.4.2.2.11 H264_I_PERIOD

- Base Address: 0x1340_0000
- Address = Base Address + 0x009C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	–
H264_I_PERIOD _ENABLE	[16]	W	Specifies enable I picture (not IDR) for H.264 encoder. This option results in open GOP.	–
H264_I_PERIOD _CONFIG	[15:0]	W	H264 Intra Period Configuration 0 = All P frames 1 = All I frames 2 = I – P – I – P 3 = I – P – P – I N = (N-1) P frames between two I frames This holds valid only if H264_I_PERIOD_ENABLE = 1.	–

NOTE: It uses I and IDR differently in H.264. It uses IDR for closed GOP and I for open GOP.
The interpretation of the MFC registers is:
[H.264 encoding]
- IDR: I_FRAME, I_FRM_CTRL, NEW_I_PERIOD
- I: H264_I_PERIOD

[MPEG4 encoding]
- I: I_FRAME, I_FRM_CTRL, NEW_I_PERIOD
- There is no concept of IDR in MPEG4

50.4.2.2.12 RC_CONTROL_CONFIG

- Base Address: 0x1340_0000
- Address = Base Address + 0x00A0, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	–
RC_FIXED _TARGET_BIT	[0]	W	Rate Control Fixed Target Bit 0 = Disables rate control 1 = Enables rate control	–

50.4.2.2.13 BATCH_INPUT_ ADDR

- Base Address: 0x1340_0000
- Address = Base Address + 0x00A4, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
BATCH_INPUT_ADDR	[31:0]	W	Specifies start address of input structure for batch encoding	–

50.4.2.2.14 BATCH_OUTPUT_ADDR

- Base Address: 0x1340_0000
- Address = Base Address + 0x00A8, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
BATCH_OUTPUT _ADDR	[31:0]	W	Specifies start address of output structure for batch encoding	–

50.4.2.2.15 BATCH_OUTPUT_SIZE

- Base Address: 0x1340_0000
- Address = Base Address + 0x00AC, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
BATCH_OUTPUT_SIZE	[31:0]	W	Specifies size of output structure for batch encoding	–

50.4.2.2.16 HIERARCHICAL_P_QP

- Base Address: 0x1340_0000
- Address = Base Address + 0x00E0, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:18]	–	Reserved	–
T3_FRAME_QP	[17:12]	W	QP for T3 frames	–
T2_FRAME_QP	[11:6]	W	QP for T2 frames	–
T0_FRAME_QP	[5:0]	W	QP for T0 frames	–

50.4.2.2.17 H264_ENC_MB_INFO_ADDR

- Base Address: 0x1340_0000
- Address = Base Address + 0x0720, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	–
H264_ENC_MB _INFO_ADDR	[16:0]	RW	This value adds to DRAMBASE as 11-bit shift left.	–

50.4.2.3 Common Control

50.4.2.3.1 ALLOCATED_LUMA_DPB_SIZE

- Base Address: 0x1340_0000
- Address = Base Address + 0x0064, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
ALLOCATED _LUMA_DPB_SIZE	[31:0]	W	Allocated Luma DPB Size Register It is the size of Luma DPB that host allocates for decoding. The allocated DPB size takes into account the tile mode format.	–

50.4.2.3.2 ALLOCATED_ CHROMA_DPB_SIZE

- Base Address: 0x1340_0000
- Address = Base Address + 0x0068, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
ALLOCATED_CHROMA _DPB_SIZE	[31:0]	W	Allocated Chroma DPB Size Register It is the size of chroma DPB that host allocates for decoding. The allocated DPB size takes into account the tile mode format.	–

50.4.2.3.3 ALLOCATED_MV_SIZE

- Base Address: 0x1340_0000
- Address = Base Address + 0x006C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
ALLOCATED _MV_SIZE	[31:0]	W	Allocated Motion Vector Size Register It is the size of motion vector buffers that host allocates for decoding. The allocated DPB size takes into account the tile mode format. It holds valid for H.264 only.	–

50.4.2.3.4 FLUSH_CMD_TYPE

- Base Address: 0x1340_0000
- Address = Base Address + 0x0080, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
FLUSH_CMD_TYPE	[31:0]	R	Flush Command Type Register 0 = Encoder 1 = Decoder	–

50.4.2.3.5 FLUSH_CMD_INBUF1

- Base Address: 0x1340_0000
- Address = Base Address + 0x0084, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
FLUSH_CMD_INBUF1	[31:0]	R	Flush Command Input Buffer 1 Register The input buffer pointer holds the 11bit right-shifted address Encoder: Specifies current Y address Decoder: Specifies CPB buffer address	–

50.4.2.3.6 FLUSH_CMD_INBUF2

- Base Address: 0x1340_0000
- Address = Base Address + 0x0088, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
FLUSH_CMD_INBUF2	[31:0]	R	Flush Command Input Buffer 2 Register The input buffer pointer holds the 11bit right-shifted address Encoder: Specifies current C address Decoder: Specifies descriptor buffer address	–

50.4.2.3.7 FLUSH_CMD_OUTBUF

- Base Address: 0x1340_0000
- Address = Base Address + 0x008C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
FLUSH_CMD _OUTBUF	[31:0]	R	Flush Command Output Buffer Register The output buffer pointer holds the 11bit right-shifted address Encoder: Specifies stream buffer start address Decoder: N/A	–

50.4.2.3.8 MIN_LUMA_DPB_SIZE

- Base Address: 0x1340_0000
- Address = Base Address + 0x00B0, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
MIN_LUMA_DPB _SIZE	[31:0]	R	Minimum Luma DPB Size Register Specifies minimum byte size of Luma DPB that host allocates for decoding. The size takes into account the tile mode format.	–

50.4.2.3.9 MIN_CHROMA_DPB_SIZE

- Base Address: 0x1340_0000
- Address = Base Address + 0x00BC, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
MIN_CHROMA _DPB_SIZE	[31:0]	R	Specifies minimum byte size of chroma DPB that host allocates for decoding. The size takes into account the tile mode format.	–

50.4.2.3.10 DBG_HISTORY_INPUT0

- Base Address: 0x1340_0000
- Address = Base Address + 0x00D0, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
ALLOCATED_MEM _SIZE	[31:16]	W	Specifies size of allocated memory for stage counter history dump.	–
RSVD	[15:1]	–	Reserved	–
ENABLE_DEBUG _HISTORY	[0]	W	Enable Debug History 0 = Disables debug history 1 = Enables debug history	–

50.4.2.3.11 DBG_HISTORY_INPUT1

- Base Address: 0x1340_0000
- Address = Base Address + 0x00D4, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
ALLOCATED _MEM_ADDR	[31:0]	W	Specifies address of allocated memory for the stage counter history dump	–

50.4.2.3.12 DBG_HISTORY_OUTPUT

- Base Address: 0x1340_0000
- Address = Base Address + 0x00D8, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
DBG_HISTORY _SIZE	[31:0]	R	Specifies number of bytes which MFC dumps as stage counter history.	–

50.4.2.4 Metadata Control

50.4.2.4.1 METADATA_ENABLE

- Base Address: 0x1340_0000
- Address = Base Address + 0x0038, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	–
NUM_CONCEALED_MB_ENABLE	[7]	W	Number of Concealed Macroblock Enable 0 = Disables number of concealed macroblock info 1 = Enables number of concealed macroblock info	–
EXTRADATA_ENABLE	[6]	W	Extra Data Enable 0 = Disables extra metadata 1 = Enables extra metadata	–
ENC_SLICE_SIZE_ENABLE	[5]	W	Slice Size Enable 0 = Disables slice size info 1 = Enables slice size info	–
VUI_ENABLE	[4]	W	VUI Enable 0 = Disables VUI info 1 = Enables VUI info	–
SEI_NAL_ENABLE	[3]	W	SEI NAL Enable 0 = Disables SEI NAL info 1 = Enables SEI NAL info	–
VC1_PARAM_ENABLE	[2]	W	VC1 Parameter Enable 0 = Disables VC1 parameter store 1 = Enables VC1 parameter store	–
CONCEALED_MB_ENABLE	[1]	W	Concealed Macroblock 0 = Disables concealed macroblock info 1 = Enables concealed macroblock info	–
QP_ENABLE	[0]	W	QP Enable 0 = Disables QP info 1 = Enables QP info	–

NOTE: When QP_ENABLE=1, MFC sets MFC_QP_OUT_EN and MFC_QP_OFFSET internally METADATA_ENABLE and EXT_METADATA_START_ADDR should be set on SEQ_START.

50.4.2.4.2 METADATA_STATUS

- Base Address: 0x1340_0000
- Address = Base Address + 0x003C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	–
METADATA_STATUS	[0]	R	Metadata Status 0 = Metadata does not exist 1 = Metadata exists	–

50.4.2.4.3 METADATA_DISPLAY_INDEX

- Base Address: 0x1340_0000
- Address = Base Address + 0x0040, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
METADATA_DISPLAY_INDEX	[31:0]	R	Metadata Display Index When it enables concealed macroblock or QP in metadata info, it returns the information of displayed frame. It does not return information of decoded frame. It returns the index of metadata in shared memory.	–

50.4.2.4.4 EXT_METADATA_START_ADDR

- Base Address: 0x1340_0000
- Address = Base Address + 0x0044, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
EXT_METADATA_START_ADDR	[31:0]	W	Metadata Start Address The start address of metadata memory configures the QP, concealed macroblock number, VC1 parameters, SEI, VUI, and slice size information	–

NOTE: The size of EXT_METADATA_START_ADDR buffer is 54 words (216 bytes).
METADATA_ENABLE and EXT_METADATA_START_ADDR should be set on SEQ_START.

50.4.2.4.5 PUT_EXTRADATA

- Base Address: 0x1340_0000
- Address = Base Address + 0x0048, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	–
PUT _EXTRADATA	[0]	W	Put Extra Data Host informs MFC if extra metadata exists for each frame decoding. It is valid only for set value of EXTRADATA_ENABLE 0 = No extra metadata 1 = Extra metadata exists	–

50.4.2.4.6 EXTRADATA_ADDR

- Base Address: 0x1340_0000
- Address = Base Address + 0x004C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
EXTRADATA _ADDR	[31:0]	W	Extra Data Address Register Host informs MFC the extra metadata address. MFC copies the extra metadata to shared memory only for set values of EXTRADATA_ENABLE and PUT_EXTRADATA.	–

50.5 Metadata Interface

Use shared memory to exchange information between the MFC core and an external host. The host allocates metadata input and output buffers in shared memory. The host also informs MFC about the pointer status that uses EXT_METADATA_START_ADDR register.

The structure of the metadata buffer output is OpenMax compliant which is described as:

```
Typedef struct OMX_OTHER_EXTRADATATYPE {
OMX_U32                 nSize;
OMX_VERSIONTYPE         nVersion;
OMX_U32                 nPortIndex;
OMX_EXTRADATATYPE       eType;
OMX_U32                 nDataSize;
OMX_U8                  data[1];
} OMX_OTHER_EXTRADATATYPE;
```

NOTE: It should inform the start and end addresses through metadata buffer <n> addr and metadata buffer size.

50.5.1 Shared Memory Interface for Decoders

MFC decoders use shared memory to report QP, concealed macroblock information, VC1 parameters, SEI, and VUI information.

Table 50-1 describes the payload in the shared memory.

Table 50-1 Payload in the Shared Memory

Element	Payload
Metadata[0]	No more metadata
Metadata[1]	QP information of each decoded macroblocks
Metadata[2]	An error map of concealed macroblock information
Metadata[3]	VC1 parameters
Metadata[4]	SEI NAL information
Metadata[5]	VUI information
Metadata[6]	Number of concealed macroblocks

Figure 50-6 illustrates the shared memory input for decoders. It should provide the shared memory input structure to MFC on the INIT_BUFFERS command.

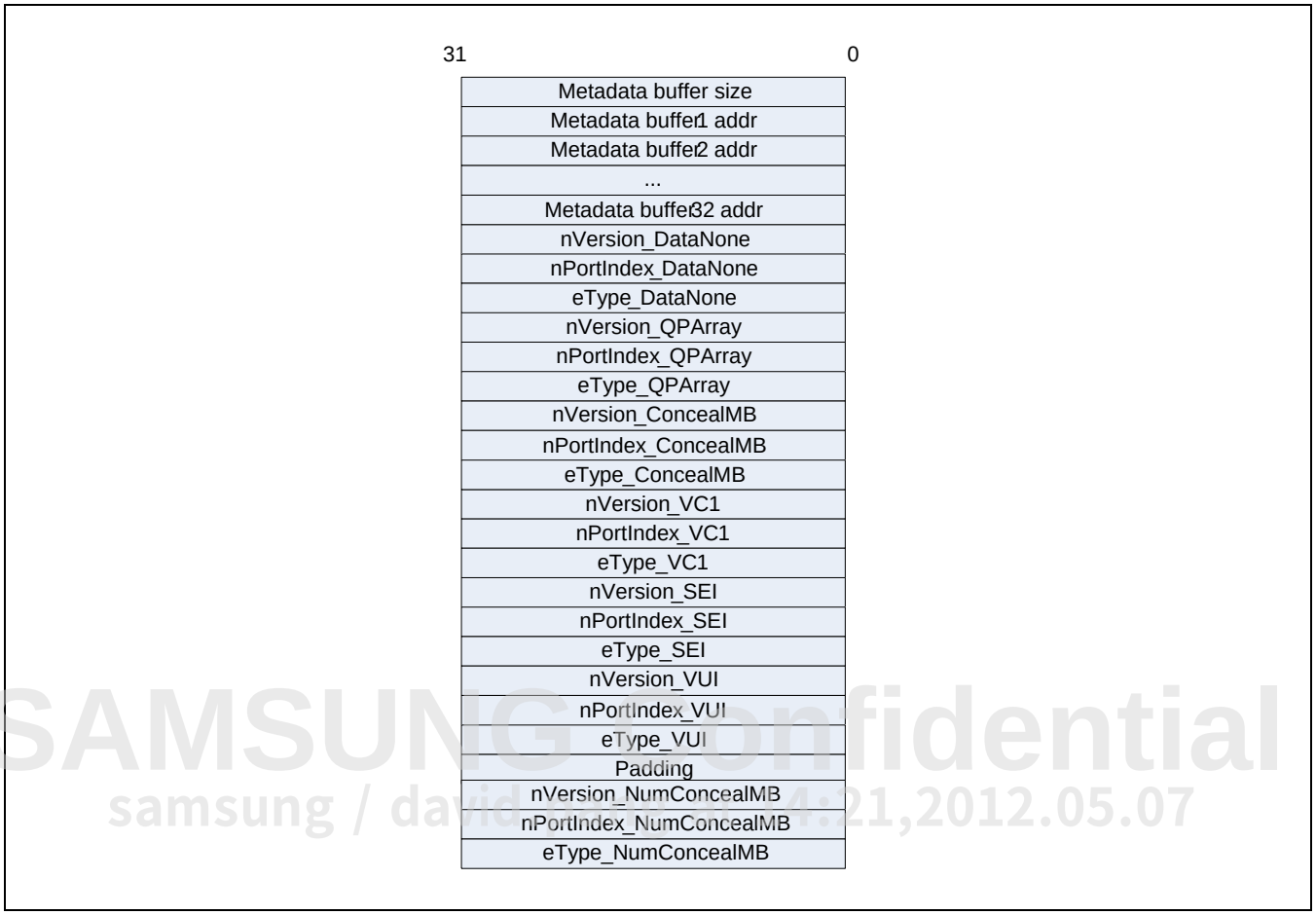


Figure 50-6 Shared Memory Input for Decoders

There is a one-to-one mapping between each metadata output buffer and corresponding DPB. Hence the host takes the responsibility and allocates NUM_DPB number of metadata buffers of same sizes. It does not use the remaining metadata buffer addresses for metadata storage. It communicates the size through first fields in the shared memory input.

However, the VUI information gets embedded in the sequence header, SPS in H.264. So there exists no one-to-one relationship between VUI and metadata buffers. MFC returns the appropriate VUI for a specific frame for an SPS change.

In metadata output structure, the 20 bytes header and payload for each field are present only if corresponding bits in METADATA_ENABLE register are set. However, if the metadata buffer is full, it discards the payload.

The input bit stream that consists of one frame data and one or more extra data gets copied to the ExtraData metadata. The EXTRADATA_ADDR register provides the address for ExtraData metadata. *Figure 50-7* illustrates the shared memory output for decoders.

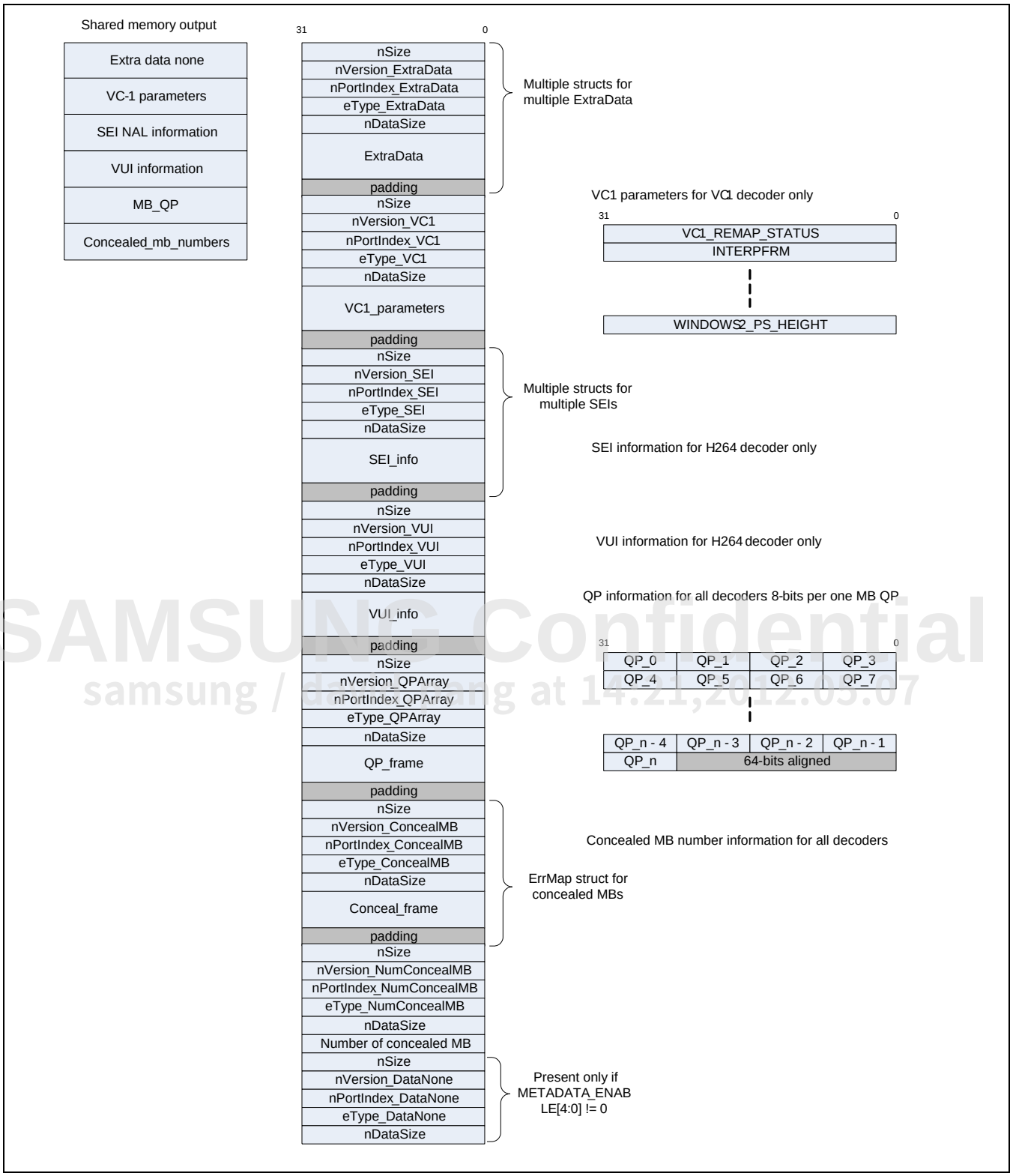


Figure 50-7 Shared Memory Output for Decoders

Figure 50-8 illustrates the detailed data structure for VC1 parameters. The numbers in the parenthesis specify the effective number of bits for each field.

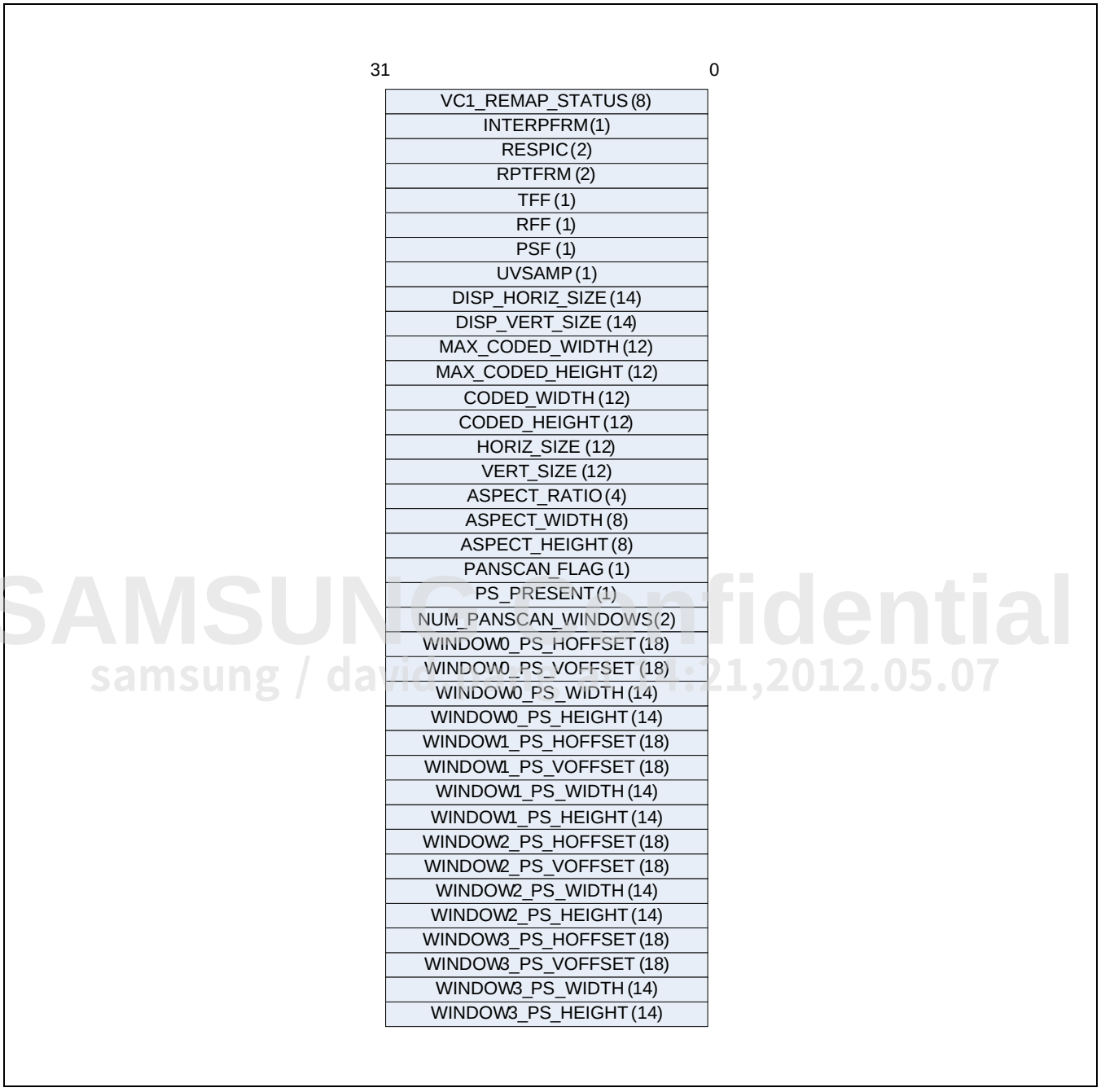


Figure 50-8 VC1 Parameters

50.5.2 Shared Memory Interface for Encoders

MFC encoder uses shared memory that reports slice information when a frame consists of multiple slices.

Figure 50-9 illustrates the shared memory input for encoders. It provides metadata input structure to MFC for each FRAME_START command so that host can update the metadata buffer address accordingly

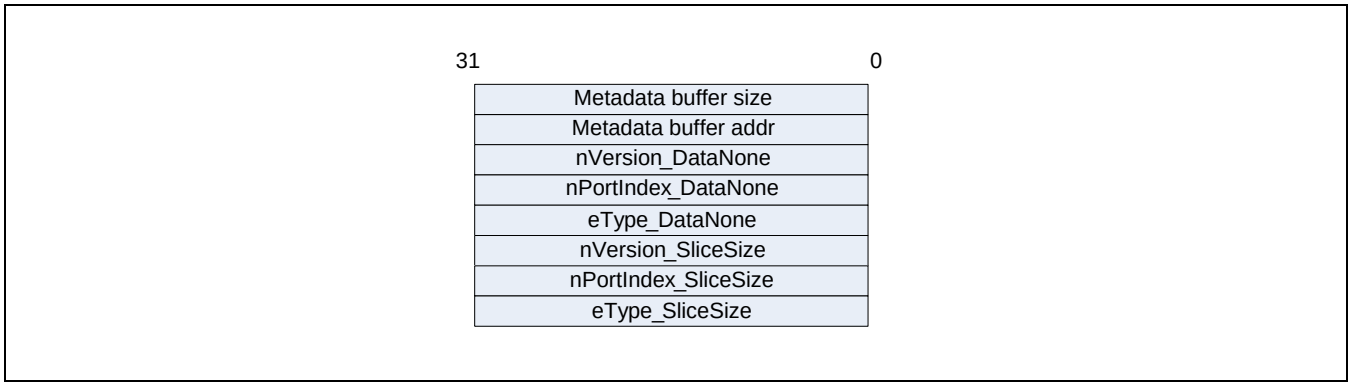


Figure 50-9 Shared Memory Input for Encoders

Figure 50-10 illustrates the metadata output for encoders. There exists only one metadata field for encoders in which the slice size metadata provides offset and length information for multiple slices.

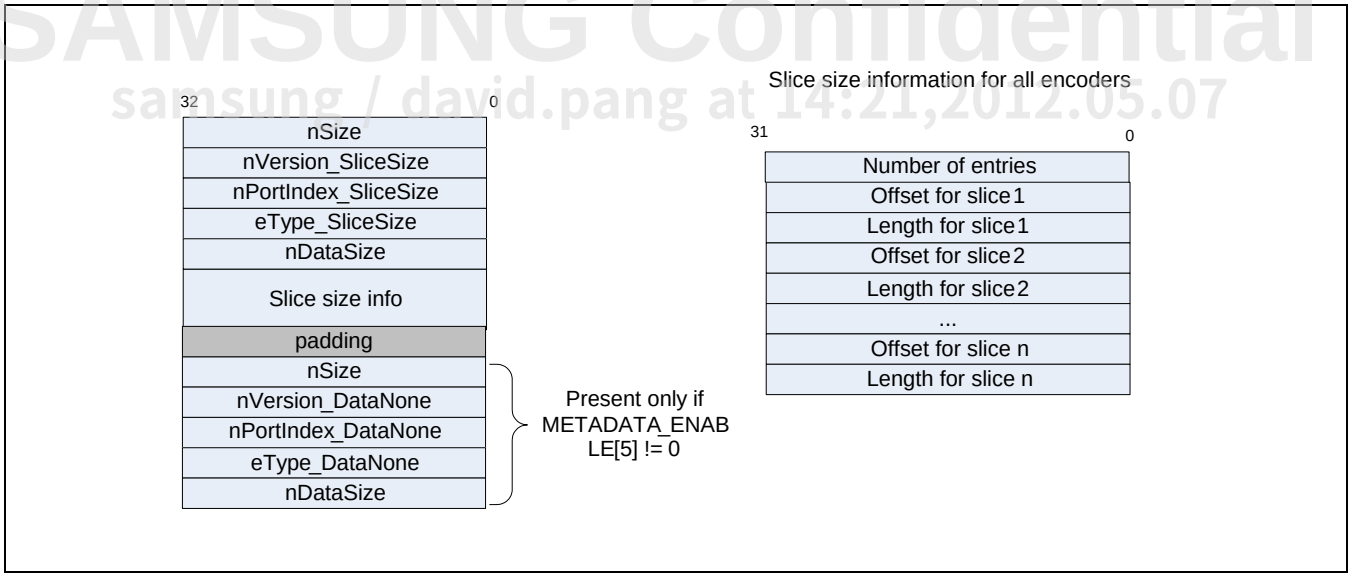


Figure 50-10 Shared Memory Output for Encoders

50.6 Appendix

50.6.1 Summary of Buffer Requirements

There are different types of buffers in MFC. Memory buffers in firmware/host interface have different alignment and size requirements compared to the buffers in firmware/hardware engine interface.

Type	Address	Size
Linear memory	Specifies alignment and format of linear memory Alignment: 2 KB aligned Format: 11-bit right shifted	Refer to Section *50.3.2.4 Common Address Control* for more information.
	Example: CH_ES_ADDR, CH_DESC_ADDR, DEC_NB_DCAC, and so on.	
Tiled memory	Specifies alignment and format of tiled memory Alignment: 8 KB aligned Format: 11-bit right shifted	Multiple of 8 KB
	Example: DEC_LUMA_x, DEC_CHROMA_x, DEC_MV_x	
Shared memory	Specifies alignment of shared memory Alignment: 4 byte aligned	Multiple of 4 bytes
	Example: HOST_WR_ADDR, EXT_METDATA_START_ADDR, and so on.	

NOTE:

1. MC_DRAMBASE_ADDR computes the physical address.
2. Linear memory and shared memory are set as an offset from MC_DRAMBASE_ADDR_A. The tiled memory is set as an offset from MC_DRAMBASE_ADDR_A or MC_DRAMBASE_ADDR_B according to the base address index.

50.6.2 Batch Encoding Interface

MFC supports frame batch encoding so that host submits multiple frames in a single command and receives a response. MFC processes multiple frames and raises an interrupt when all of the frames get encoded.

The batch encoding interface uses shared memory in which the host specified information are:

- Number of frames
- List of information for each frame

MFC uses the listed information and encodes multiple frames in the encoding order and generates an interrupt along with an output structure. The output structure describes details of encoded frames. The required shared memory fields are:

- BATCH_INTPUT_ADDR
- BATCH_OUTPUT_ADDR
- BATCH_OUTPUT_SIZE

Example 50-2 and *Example 50-3* describes the input and output structures.

Example 50-2 describes the input structure used in shared memory output for encoders.

Example 50-2 Shared Memory Output for Encoders

```
typedef struct
{
      unsigned int data_id;  // FRAME_BATCH_CMD, FRAME_BATCH_RESP
      unsigned int data_size;
      unsigned int number_of_frames;

      struct frame_info_t
      {
              unsigned int cur_luma_addr;
              unsigned int cur_chroma_addr;
              unsigned int set_frame_tag;
              unsigned int vop_timing;
      } frame_info[number_of_frames];

      unsigned int number_of_stream_buffer;
      unsigned int stream_buffer_size;
      unsigned int stream_buffer_address[number_of_stream_buffer];

} input_struct;
```

Example 50-3 illustrates the output structure used in shared memory output for encoders.

Example 50-3 Shared Memory Output for Encoders

```
typedef struct
{
	unsigned int data_id;                                   // FRAME_BATCH_CMD, FRAME_BATCH_RESP
	unsigned int data_size;                                 // data size that firmware generated

	unsigned int number_of_frames;
	struct stream_info_t
	{
		unsigned int stream_info_id;
		unsigned int stream_info_size;
		unsigned int cur_luma_addr;
		unsigned int cur_chroma_addr;
		unsigned int pic_count;
		unsigned int slice_type;
		unsigned int get_frame_tag;
		unsigned int size_of_encoded_streams;
		unsigned int number_of_stream_buffers;
		unsigned int stream_address[number_of_stream_buffers];
	} stream_info[number_of_frames];

	unsigned int number_of_frames_accepted;
	unsigned int number_of_stream_buffers_unused;           // unused output buffers
	unsigned int stream_address_unused[number_of_stream_buffers_unused];

} output_struct;
```


When host invokes FRAME_BATCH_START, the MFC first reads an input structure.
The BATCH_INTPUT_ADDR points to the address from which the MFC reads the address of the input structure.
The fields in the input_struct are:

- data_id: "data_id' is either "FRAME_BATCH_CMD" or "FRAME_BATCH_RESP"
- data_size: It is the size of input_struct. It also includes the size of data_id.
- number_of_frames: It is the number of frames that host wants to encode
- cur_luma_addr: It is the address of input Luma picture that corresponds to CURRENT_Y_ADDR.
- cur_chroma_addr: It is the address of input Chroma picture that corresponds to CURRENT_C_ADDR.
- set_frame_tag: It is the frame tag of an input frame. It holds the same definition as SET_FRAME_TAG.
- vop_timing: It is the VOP timing of an input frame.
- number_of_stream_buffer: It is the number of output buffers of encoded frames.
- stream_buffer_size: It is the size of an output buffer of an encoded frame.
- stream_buffer_address: The address of output buffers of encoded frames

When MFC completes the frame encode, it generates an output structure stored at BATCH_OUTPUT_ADDR. The BATCH_OUTPUT_SIZE defines the size of output_struct structure. The structure provides information of the encoded stream. Fields of structure are:

- data_id: "data_id' is either "FRAME_BATCH_CMD" or "FRAME_BATCH_RESP"
- data_size: It is the size of the output_struct. It also includes the size of data_id.
- number_of_frames: It is the number of frames that host
- number_of_frames_accepted: It is the number of frames that host accepts to complete encoding.
- cur_luma_addr: It is the address of the encoded input Luma picture.
- cur_chroma_addr: It is the address of the encoded input Chroma picture.
- pic_count: It is the picture count in display order. The pic_count provides information for reordering, since MFC encodes input frames in decoding order, not in display order.
- slice_type: It specifies the slice type. It holds the same definition as that of ENC_SLICE_TYPE.
- get_frame_tag: It is the frame tag of an output frame. It holds the same definition as that of GET_FRAME_TAG.
- size_of_encoded_streams: It is the size of an encoded bit stream.
- number_of_stream_buffers: It is the number of encoded stream buffers. Multiple buffers are required if the size of generated stream is greater than stream_buffer_size in the input_struct.
- stream_address: It is the address of encoded stream buffers.
- number_of_stream_buffers_unused: It is the number of unused stream buffers after the batch gets encoded.
- stream_address_unused: It is the address of unused stream buffers after the batch gets encoded.

Note that MFC stops encoding when either stream buffer is full or output_struct is full. In either case, number_of_frames_accepted will be smaller than number_of_frames. Host may resubmit the last frames from the previous input_struct as many as unaccepted in output_struct. The resubmitted frames will be encoded continuously.

When number_of_stream_buffer is greater than number_of_frames, it is possible that some of the stream buffers are not used. MFC returns them over number_of_stream_buffers_unused and stream_address_unused. Host may reuse them in the next batch.

51 Video Processor

51.1 Overview of Video Processor

Video Processor (VP) is responsible for video scaling, de-interlacing, and video post processing of TV-out data path. VP reads reconstructed YCbCr 4:2:0 video sequences from DRAM. It then processes the sequence, and sends it to on-the- fly Mixer.

Figure 51-1 illustrates the video data path.

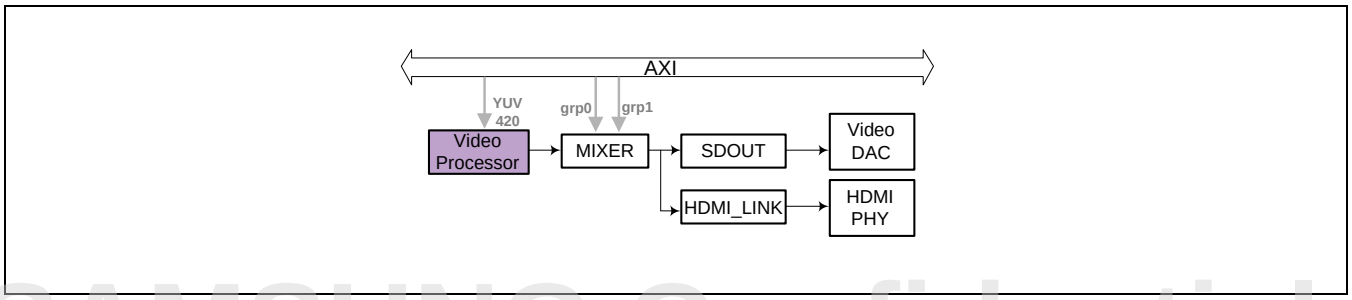


Figure 51-1 Video Data Path

51.1.1 Features

The features of video processor are:

- Input YCbCr sequence of VP is up to 1920 × 1080 at 60 Hz.
- Supports BOB/TILE (Interlaced versions of YUV420 support NV12 and NV21 type)
- Input source size up to 1920x1080, minimum size is 32 × 4
- Produces YCbCr 4:4:4 outputs to help Mixer to blend video and graphics
- Supports 1/4x to 16x vertical scaling with 4-tap/16-phase poly-phase filter
- Supports 1/4x to 16x horizontal scaling with 8-tap/16-phase poly-phase filter
- Supports Pan and Scan, Letterbox
- Supports flexible scaled video positioning within display area
- Supports 1/16 pixel resolution pan and scan mode
- Supports flexible post video processing that includes:
  - Color saturation, brightness/ contrast enhancement, and edge enhancement for SD(480p/576p)
  - Color space conversion between BT.601 and BT.709

NOTE: Refer the user's manual of MFC on TILE information.

51.2 Block Diagram

Figure 51-2 illustrates the block diagram of video processor.

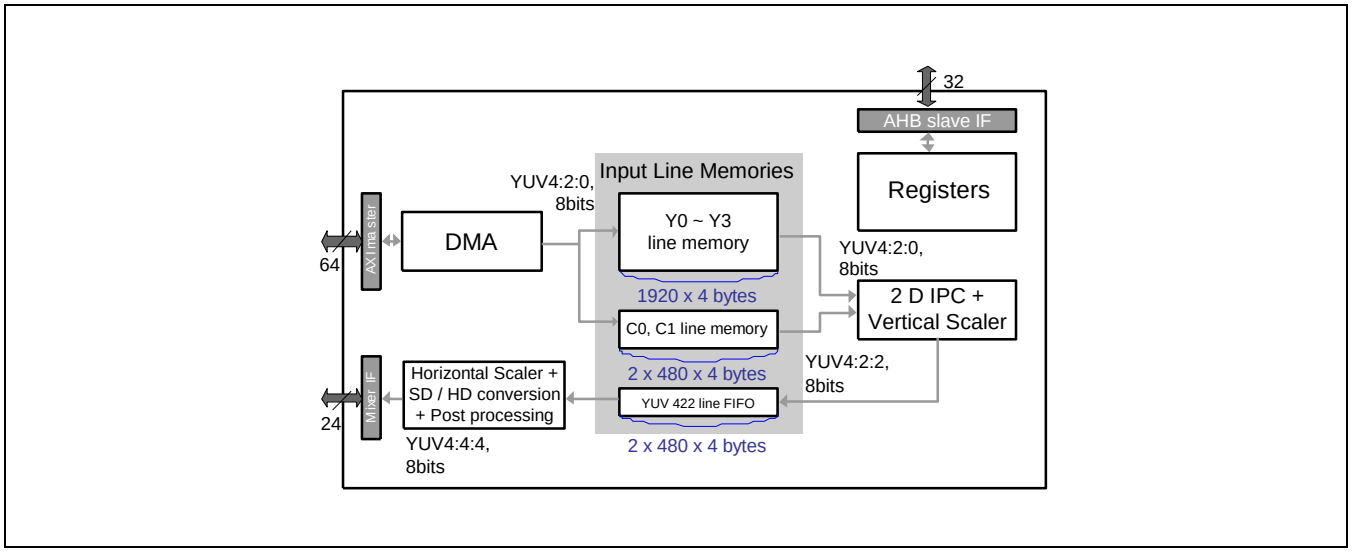


Figure 51-2 Block Diagram of Video Processor

The components of VP are:

- DMA: DMA reads the image from memory.
- Input Line Memory: It stores data to process the image.
- Registers: Registers are responsible for the configuration of VP.
- 2D-IPC and Vertical Scaler: It performs IPC and vertical scaling.
- Horizontal Scaler: It performs horizontal scaling and post processing.

51.3 Functional Description

This section describes the functional description of VP.

This section includes:

- BOB in VP
- Nterlace to Progressive Conversion

51.3.1 BOB in VP

Figure 51-3 illustrates the data type for BOB.

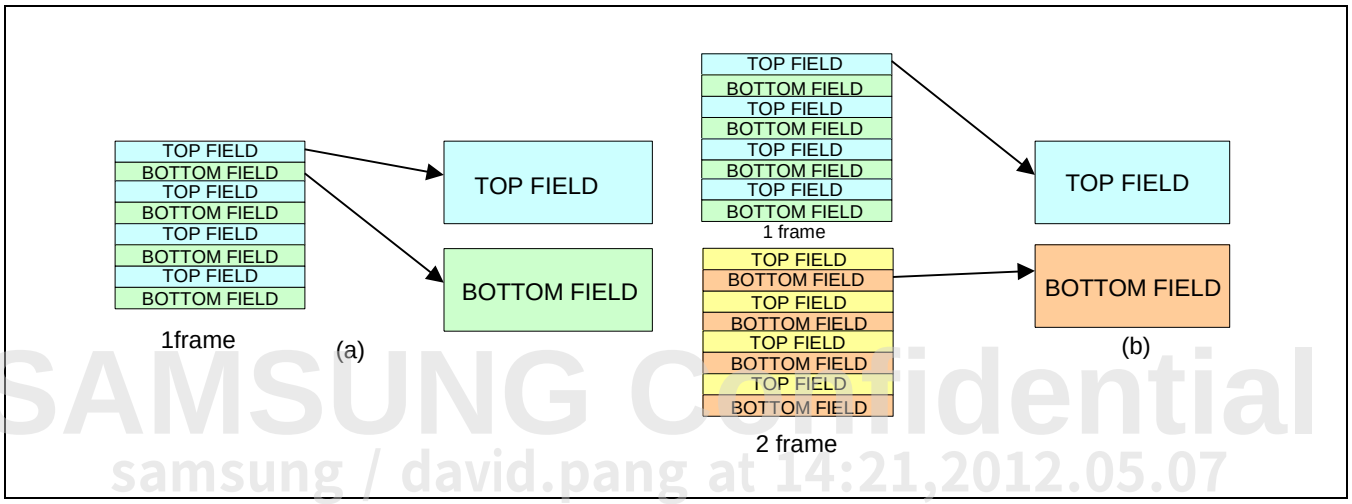


Figure 51-3 Data Type for BOB

In some applications, it is necessary to display an interlaced video signal on a non-interlaced display. Thus, some form of "de-interlacing" or "progressive conversion" is required. Video mode is one of fundamental de-interlacing algorithm. Video mode de-interlacing can be further broken down into inter-filed and intra-field processing. Particular, Intra-field processing in video mode is the simplest method to generate additional scan lines using only information in the original field. The computer industry has coined this technique as "BOB".

BOB in VP includes intra-field or inter-field. Inter-field emerges from a single frame as illustrated in *Figure 51-3* (a). Intra-field emerges from two frames as illustrated in *Figure 51-3* (b).

51.3.2 Nterlace to Progressive Conversion

Interlace to Progressive Conversion (IPC) engine converts an interlaced signal to a progressive signal. It is different with the vertical x2 scale.

Figure 51-4 illustrates the difference between IPC and X2 scale-up.

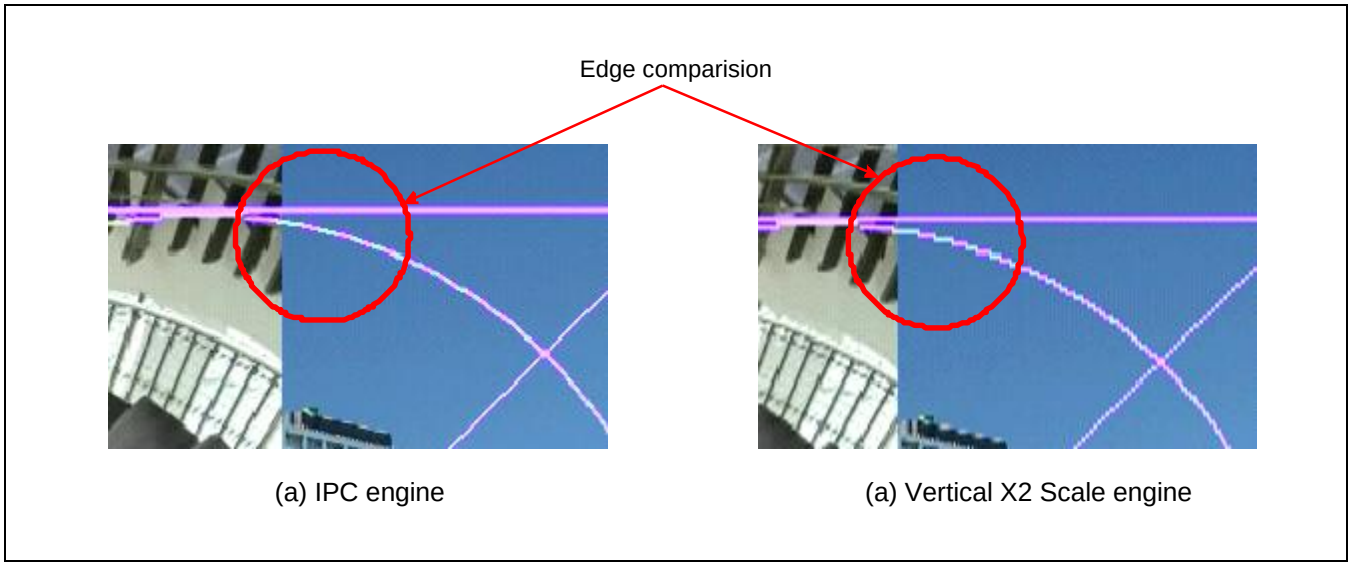


Figure 51-4 Difference Between IPC and X2 Scale-up

IPC engine executes the edge detection function. This function is based on the edge diagnosis method. This enables the IPC to estimate edge line and display a natural image.

51.4 Register Description

51.4.1 Register Map Summary

- Base Address: 0x12C0_0000

Register	Offset	Description	Reset Value
VP_ENABLE	0x0000	Specifies power-down ready and enable	0x0000_0002
VP_SRESET	0x0004	Specifies software reset	0x0000_0000
VP_SHADOW_UPDATE	0x0008	Specifies shadow register update enable	0x0000_0000
VP_FIELD_ID	0x000C	Specifies field ID of source image	0x0000_0000
VP_MODE	0x0010	Specifies VP operation mode	0x0000_0000
VP_IMG_SIZE_Y	0x0014	Specifies luminance date size	0x0000_0000
VP_IMG_SIZE_C	0x0018	Specifies chrominance date size	0x0000_0000
VP_TOP_Y_PTR	0x0028	Specifies base address for Y of top field	0x0000_0000
VP_BOT_Y_PTR	0x002C	Specifies base address for Y of bottom field	0x0000_0000
VP_TOP_C_PTR	0x0030	Specifies base address for C of top field	0x0000_0000
VP_BOT_C_PTR	0x0034	Specifies base address for C of bottom field	0x0000_0000
VP_ENDIAN_MODE	0x03CC	Specifies big/little endian mode selection	0x0000_0000
VP_SRC_H_POSITION	0x0044	Specifies horizontal offset in the source image	0x0000_0000
VP_SRC_V_POSITION	0x0048	Specifies vertical offset in the source image	0x0000_0000
VP_SRC_WIDTH	0x004C	Specifies width of the source image	0x0000_0000
VP_SRC_HEIGHT	0x0050	Specifies height of the source image	0x0000_0000
VP_DST_H_POSITION	0x0054	Specifies horizontal offset in the display	0x0000_0000
VP_DST_V_POSITION	0x0058	Specifies vertical offset in the display	0x0000_0000
VP_DST_WIDTH	0x005C	Specifies width of the display	0x0000_0000
VP_DST_HEIGHT	0x0060	Specifies height of the display	0x0000_0000
VP_H_RATIO	0x0064	Specifies horizontal zoom ratio of SRC:DST	0x0000_0000
VP_V_RATIO	0x0068	Specifies vertical zoom ratio of SRC:DST	0x0000_0000
VP_POLY8_Y0_LL	0x006C	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y0_LH	0x0070	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y0_HL	0x0074	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y0_HH	0x0078	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y1_LL	0x007C	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y1_LH	0x0080	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y1_HL	0x0084	Specifies 8-tap poly-phase filter coefficients for	0x0000_0000

Register	Offset	Description	Reset Value
		luminance horizontal scaling	
VP_POLY8_Y1_HH	0x0088	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y2_LL	0x008C	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y2_LH	0x0090	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y2_HL	0x0094	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y2_HH	0x0098	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y3_LL	0x009C	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y3_LH	0x00A0	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y3_HL	0x00A4	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY8_Y3_HH	0x00A8	Specifies 8-tap poly-phase filter coefficients for luminance horizontal scaling	0x0000_0000
VP_POLY4_Y0_LL	0x00EC	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y0_LH	0x00F0	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y0_HL	0x00F4	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y0_HH	0x00F8	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y1_LL	0x00FC	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y1_LH	0x0100	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y1_HL	0x0104	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y1_HH	0x0108	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y2_LL	0x010C	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y2_LH	0x0110	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y2_HL	0x0114	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y2_HH	0x0118	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000

Register	Offset	Description	Reset Value
VP_POLY4_Y3_LL	0x011C	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y3_LH	0x0120	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y3_HL	0x0124	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_Y3_HH	0x0128	Specifies 4-tap poly-phase filter coefficients for luminance vertical scaling	0x0000_0000
VP_POLY4_C0_LL	0x012C	Specifies 4-tap poly-phase filter coefficients for chrominance horizontal scaling	0x0000_0000
VP_POLY4_C0_LH	0x0130	Specifies 4-tap poly-phase filter coefficients for chrominance horizontal scaling	0x0000_0000
VP_POLY4_C0_HL	0x0134	Specifies 4-tap poly-phase filter coefficients for chrominance horizontal scaling	0x0000_0000
VP_POLY4_C0_HH	0x0138	Specifies 4-tap poly-phase filter coefficients for chrominance horizontal scaling	0x0000_0000
VP_POLY4_C1_LL	0x013C	Specifies 4-tap poly-phase filter coefficients for chrominance horizontal scaling	0x0000_0000
VP_POLY4_C1_LH	0x0140	Specifies 4-tap poly-phase filter coefficients for chrominance horizontal scaling	0x0000_0000
VP_POLY4_C1_HL	0x0144	Specifies 4-tap poly-phase filter coefficients for chrominance horizontal scaling	0x0000_0000
VP_POLY4_C1_HH	0x0148	Specifies 4-tap poly-phase filter coefficients for chrominance horizontal scaling	0x0000_0000
PP_CSC_Y2Y_COEF	0x01D4	Specifies Y to Y CSC coefficient setting	0x0000_0000
PP_CSC_CB2Y_COEF	0x01D8	Specifies CB to Y CSC coefficient setting	0x0000_0000
PP_CSC_CR2Y_COEF	0x01DC	Specifies CR to Y CSC coefficient setting	0x0000_0000
PP_CSC_Y2CB_COEF	0x01E0	Specifies Y to Y CSC coefficient setting	0x0000_0000
PP_CSC_CB2CB_COEF	0x01E4	Specifies CB to Y CSC coefficient setting	0x0000_0000
PP_CSC_CR2CB_COEF	0x01F0	Specifies CR to Y CSC coefficient setting	0x0000_0000
PP_CSC_Y2CR_COEF	0x01EC	Specifies Y to Y CSC coefficient setting	0x0000_0000
PP_CSC_CB2CR_COEF	0x01E8	Specifies CB to Y CSC coefficient setting	0x0000_0000
PP_CSC_CR2CR_COEF	0x01F4	Specifies CR to Y CSC coefficient setting	0x0000_0000
PP_BYPASS	0x0200	Specifies disable the post image processor	0x0000_0001
PP_SATURATION	0x020C	Specifies color saturation factor	0x0000_0080
PP_SHARPNESS	0x0210	Specifies control for the edge enhancement	0x0000_0500
PP_LINE_EQ0	0x0218	Specifies line equation for contrast duration 0	0x0000_0000
PP_LINE_EQ1	0x021C	Specifies line equation for contrast duration 1	0x0000_0000
PP_LINE_EQ2	0x0220	Specifies line equation for contrast duration 2	0x0000_0000
PP_LINE_EQ3	0x0224	Specifies line equation for contrast duration 3	0x0000_0000

Register	Offset	Description	Reset Value
PP_LINE_EQ4	0x0228	Specifies line equation for contrast duration 4	0x0000_0000
PP_LINE_EQ5	0x022C	Specifies line equation for contrast duration 5	0x0000_0000
PP_LINE_EQ6	0x0230	Specifies line equation for contrast duration 6	0x0000_0000
PP_LINE_EQ7	0x0234	Specifies line equation for contrast duration 7	0x0000_0000
PP_BRIGHT_OFFSET	0x0238	Specifies brightness offset control for Y	0x0000_0000
PP_CSC_EN	0x023C	Specifies color space conversion control	0x0000_0002
VP_VERSION_INFO	0x03FC	Specifies VP version information	0x0000_0011
VP_FIELD_ID_S	0x016C	Specifies field ID of the "Source" image	0x0000_0000
VP_MODE_S	0x0170	Specifies VP operation mode	0x0000_0000
VP_IMG_SIZE_Y_S	0x0174	Specifies luminance date tiled size	0x0000_0000
VP_IMG_SIZE_C_S	0x0178	Specifies chrominance date tiled size	0x0000_0000
VP_TOP_Y_PTR_S	0x0190	Specifies base address for Y of top field	0x0000_0000
VP_BOT_Y_PTR_S	0x0194	Specifies base address for Y of bottom field	0x0000_0000
VP_TOP_C_PTR_S	0x0198	Specifies base address for C of top frame	0x0000_0000
VP_BOT_C_PTR_S	0x019C	Specifies base address for C of bottom field	0x0000_0000
VP_ENDIAN_MODE_S	0x03EC	Specifies big/little endian mode selection	0x0000_0000
VP_SRC_H_POSITION_S	0x01AC	Specifies horizontal offset in the source image	0x0000_0000
VP_SRC_V_POSITION_S	0x01B0	Specifies vertical offset in the source image	0x0000_0000
VP_SRC_WIDTH_S	0x01B4	Specifies width of the source image	0x0000_0000
VP_SRC_HEIGHT_S	0x01B8	Specifies height of the source image	0x0000_0000
VP_DST_H_POSITION_S	0x01BC	Specifies horizontal offset in the display	0x0000_0000
VP_DST_V_POSITION_S	0x01C0	Specifies vertical offset in the display	0x0000_0000
VP_DST_WIDTH_S	0x01C4	Specifies width of the display	0x0000_0000
VP_DST_HEIGHT_S	0x01C8	Specifies height of the display	0x0000_0000
VP_H_RATIO_S	0x01CC	Specifies horizontal zoom ratio of SRC:DST	0x0000_0000
VP_V_RATIO_S	0x01D0	Specifies vertical zoom ratio of SRC:DST	0x0000_0000
PP_BYPASS_S	0x0258	Specifies disabling the post image	0x0000_0000
PP_SATURATION_S	0x025C	Specifies color saturation factor	0x0000_0000
PP_SHARPNESS_S	0x0260	Specifies control for the edge enhancement	0x0000_0000
PP_LINE_EQ0_S	0x0268	Specifies line equation for contrast duration 0	0x0000_0000
PP_LINE_EQ1_S	0x026C	Specifies line equation for contrast duration 1	0x0000_0000
PP_LINE_EQ2_S	0x0270	Specifies line equation for contrast duration 2	0x0000_0000
PP_LINE_EQ3_S	0x0274	Specifies line equation for contrast duration 3	0x0000_0000
PP_LINE_EQ4_S	0x0278	Specifies line equation for contrast duration 4	0x0000_0000
PP_LINE_EQ5_S	0x027C	Specifies line equation for contrast duration 5	0x0000_0000
PP_LINE_EQ6_S	0x0280	Specifies line equation for contrast duration 6	0x0000_0000
PP_LINE_EQ7_S	0x0284	Specifies line equation for contrast duration 7	0x0000_0000

Register	Offset	Description	Reset Value
PP_BRIGHT_OFFSET_S	0x0288	Specifies brightness offset control for Y	0x0000_0000
PP_CSC_EN_S	0x028C	Specifies color space conversion control	0x0000_0000
PP_CSC_Y2Y_COEF_S	0x0290	Specifies Y to Y CSC coefficient setting	0x0000_0000
PP_CSC_CB2Y_COEF_S	0x0294	Specifies CB to Y CSC coefficient setting	0x0000_0000
PP_CSC_CR2Y_COEF_S	0x0298	Specifies CR to Y CSC coefficient setting	0x0000_0000
PP_CSC_Y2CB_COEF_S	0x029C	Specifies Y to Y CSC coefficient setting	0x0000_0000
PP_CSC_CB2CB_COEF_S	0x02A0	Specifies CB to Y CSC coefficient setting	0x0000_0000
PP_CSC_CR2CB_COEF_S	0x02AC	Specifies CR to Y CSC coefficient setting	0x0000_0000
PP_CSC_Y2CR_COEF_S	0x02A8	Specifies Y to Y CSC coefficient setting	0x0000_0000
PP_CSC_CB2CR_COEF_S	0x02A4	Specifies CB to Y CSC coefficient setting	0x0000_0000
PP_CSC_CR2CR_COEF_S	0x02B0	Specifies CR to Y CSC coefficient setting	0x0000_0000

NOTE: The software sets the appropriate values to VP registers. And this information are copied to the corresponding shadow registers when V-SYNC is invoked. Video processor is actually working according to these shadow registers.

51.4.1.1 VP_ENABLE

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0002

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	0
VP_ON_S	[2]	RW	Shadow bit of the bit[0] This bit is Read-Only	0
VP_OPERATION_STATUS	[1]	RW	Specifies Operation Status of VP 0 = VP is in operating 1 = VP is in idle This bit is Read-Only	1
VP_ON	[0]	RW	Specifies On status of VP 0 = Disables the VP 1 = Enables the VP NOTE: The SFRs of VP and Image Mixer is updated by Vertical Sync of Timing Generator. Therefore, SFRs must be configured before enabling this bit. The sequence to enable TVSS is: "VP -> MIXER → HDMI". Also, because SFRs are updated by Vertical Sync, the disabling sequence is following as: "VP → MIXER → HDMI". This bit is Read-Write.	0

51.4.1.2 VP_SRESET

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
VP_SRESET	[0]	RW	Specifies Software Reset Status 0 = Software reset is set and the last soft reset is complete. 1 = VP is now processing software reset sequence.	0

51.4.1.3 VP_SHADOW_UPDATE

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
VP_SHADOW_UPDATE	[0]	RW	Specifies Update of Shadow registers 0 = Does not update the shadow registers at the rising edge of vertical sync. 1 = Updates shadow registers. (This register is cleared by Hardware at the rising edge of vertical sync.) NOTE: Shadow registers are listed in shadow register map table.	0

51.4.1.4 VP_FIELD_ID

- Base Address: 0x12C0_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
VP_FIELD_ID	[0]	RW	Specifies Field ID value of VP. 0 = Top field 1 = Bottom field When VP_MODE[2] is set to HIGH, this bit shows the current FIELD information. If VP_MODE[2] is set to LOW, then this bit controls the pointer of top and bottom field.	0

51.4.1.5 VP_MODE

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:26]	–	Reserved	0
RTQoSTH	[25:16]	RW	Specifies RTQoS Threshold Level Configure. 0 = Do not use QoS 1 to 719 = Specifies the threshold level 720 to 1023 = Reserved NOTE: The video processor contains a 720-depth internal DMA FIFO. Therefore, you can adjust FIFO threshold level.	0
RSVD	[15:7]	–	Reserved (Read as zero. Do not modify this bit.)	0
IMG_TYPE	[6]	RW	Specifies Image Type 0 = YUV420 NV12 1 = YUV420 NV21	0
LINE_SKIP	[5]	RW	Specifies Line Skip 0 = DMA OFF 1 = DMA ON If this bit sets to 1, then DMA skips a line per two lines while it reads line data. This bit controls DMA operation.	0
MEM_MODE	[4]	RW	Specifies Memory Mode 0 = Linear Mode 1 = Tile Mode Refer to user's manual of MFC for more information.	0
CROMA_EXPANSION	[3]	RW	Specifies Croma Expansion 0 = Uses only C_TOP_PTR 1 = Uses both C_TOP_PTR and C_BOT_PTR If this bit sets to 0, it uses only the chrominance of TOP field. If set to 1, it uses the chrominance of both TOP and BOTTOM fields.	0
FIELD_ID_AUTO_TOGGLING	[2]	RW	Specifies FIELD_ID sets as user-defined or automatically toggles 0 = FIELD_ID is user defined 1 = FIELD_ID automatically toggles by V_SYNC The V-SYNC signal is responsible for auto-toggle. The FIELD_ID_AUTO_TOGGLING bit changes the DMA base address. NOTE: The VP_FIELD_ID_S register toggles only if FIELD_ID_AUTO_TOGGLING = 1. It does not depend on the VP_FIELD_ID bit status.	0
2D_IPC	[1]	RW	Specifies interlace to Progressive Conversion. 0 = Disables 2D-IPC 1 = Enables 2D-IPC VP displays progressive scan using one field image.	0

Name	Bit	Type	Description	Reset Value
RSVD	[0]	–	Reserved (This bit should be set to zero.)	0

The Guide of Configuration

Figure 51-5 illustrates the examples of usage cases.

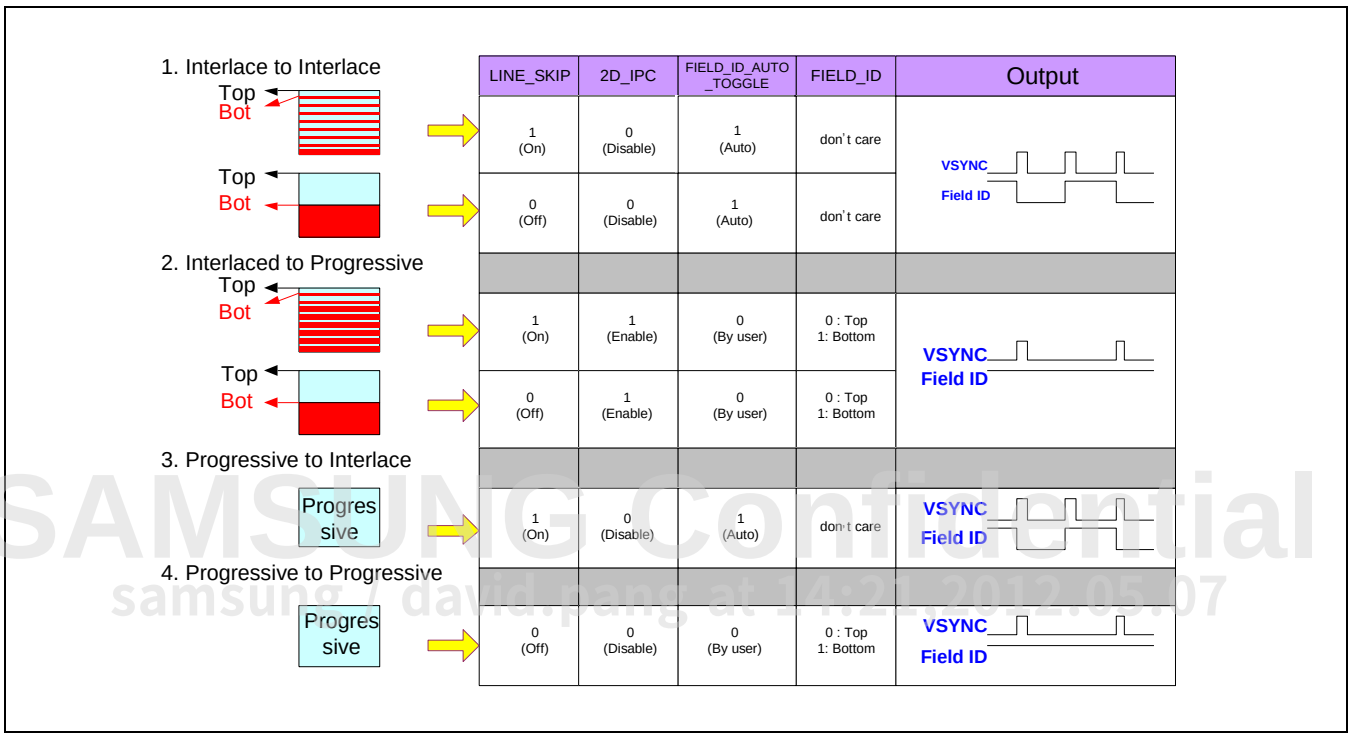


Figure 51-5 Examples of Usage Cases

51.4.1.6 VP_IMG_SIZE_Y

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
VP_IMG_HSIZE_Y	[29:16]	RW	Specifies Horizontal Size of Image in the range (8 to 8192) The range (8 to 8192) excludes –1. NOTE: LSB [2:0] should be 3'b000 for 64-bit interface. It does not allow 0 and values greater than 8192.	0
RSVD	[15:14]	–	Reserved	0
VP_IMG_VSIZE_Y	[13:0]	RW	Specifies Vertical Size of Image in the range (1 to 8192)	0

51.4.1.7 VP_IMG_SIZE_C

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
VP_IMG_HSIZE_C	[29:16]	RW	Specifies Horizontal Size of Image in the range of (8 to 8192) The range (8 to 8192) excludes –1. NOTE: LSB [2:0] should be 3'b000 for 64-bit interface. It does not allow 0 and values greater than 8192.	0
RSVD	[15:14]	–	Reserved	0
VP_IMG_VSIZE_C	[13:0]	RW	Specifies Vertical Size of Image in the range of (1 to 8192)	0

51.4.1.8 VP_TOP_Y_PTR

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
VP_TOP_Y_PTR	[31:0]	[31:0]	Specifies base address for Luminance of Top Field It should be integer multiples of 8. NOTE: LSB[2:0] should be 3'b000	0

51.4.1.9 VP_BOT_Y_PTR

- Base Address: 0x12C0_0000
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
VP_BOT_Y_PTR	[31:0]	RW	Specifies base address for Luminance of Bottom field It should be integer multiples of 8. NOTE: LSB[2:0] should be 3'b000. If TILE mode is enable, VP_BOT_Y_PTR = VP_TOP_Y_PTR + 0x40	0

51.4.1.10 VP_TOP_C_PTR

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
VP_TOP_C_PTR	[31:0]	RW	Specifies base address for Chrominance of Top field It should be integer multiples of 8. NOTE: LSB[2:0] should be 3'b000.	0

51.4.1.11 VP_BOT_C_PTR

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
VP_BOT_C_PTR	[31:0]	RW	Specifies base address for Chrominance of Bottom field It should be integer multiples of 8. NOTE: LSB[2:0] should be 3'b000. If TILE mode is enable, VP_BOT_C_PTR = VP_TOP_C_PTR + 0x40	0

51.4.1.12 VP_ENDIAN_MODE

- Base Address: 0x12C0_0000
- Address = Base Address + 0x03CC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved (Read as zero. Do not modify this bit.)	0
VP_ENDIAN_MODE	[0]	RW	Specifies Big Endian or Little Endian Mode 0 = Big Endian 1 = Little Endian Refer to *Figure 51-6* for more information.	0

Figure 51-6 illustrates the endian mode.

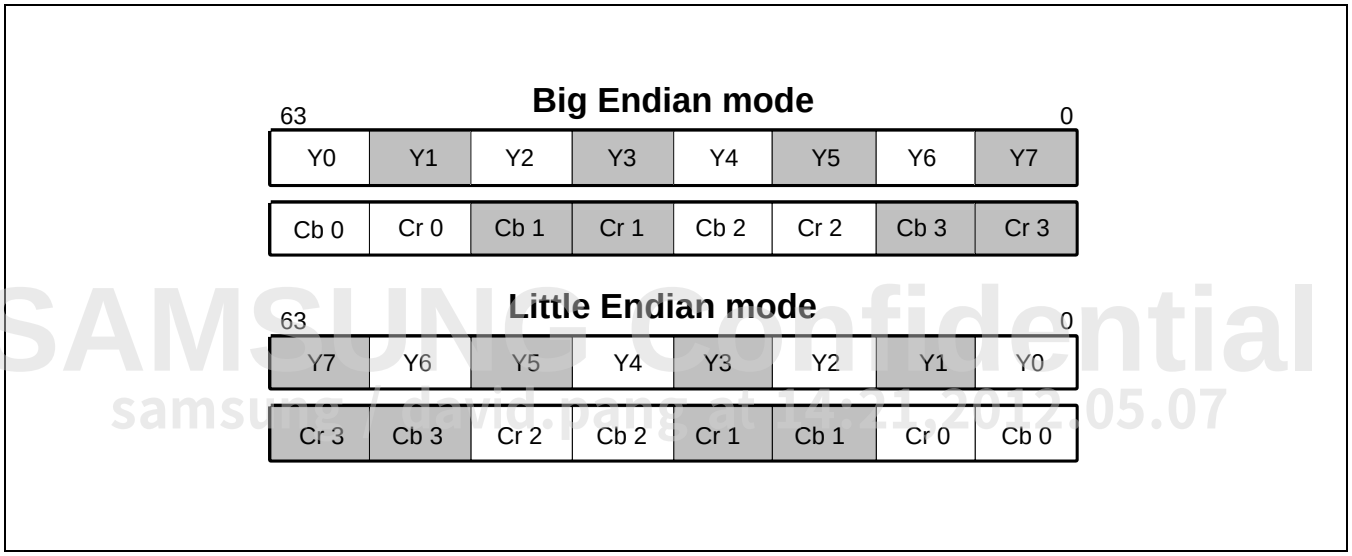


Figure 51-6 Endian Mode

51.4.1.13 VP_SRC_H_POSITION

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:15]	–	Reserved	0
VP_SRC_H_POSITION	[14:0]	RW	Horizontal offset in the source image,(11.4) format NOTE: 1. For source image cropping, (VP_SRC_H_POSITION + VP_SRC_WIDTH) <= (VP_IMG_HSIZE_Y) 2. (11.4) format implies that 11 is an integer and 4 is a fraction. For example, In case of H Position = 4 4 (0x4 (h) = 0100 (b)) is integer. Due to 4-bit fraction, 0100(b), is the left shift operation and it should perform four times. As a result, register value is 4 × 2^4 = 64 = 0x40.	0

51.4.1.14 VP_SRC_V_POSITION

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0048, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	0
VP_SRC_V_POSITION	[10:0]	RW	Specifies Vertical offset in the Source image. This value should be in the range between 0 and VP_SRC_HEIGHT. If LINE_SKIP = 1, then VP_SRC_V_POSITION should be half of its value as when LINE_SKIP = 0.	0

51.4.1.15 VP_SRC_WIDTH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x004C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved	0
VP_SRC_WIDTH	[10:0]	RW	Width of the Source image NOTE: Min = 32	0

51.4.1.16 VP_SRC_HEIGHT

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	0
VP_SRC_HEIGHT	[10:0]	RW	Height of the Source image If LINE_SKIP = 1, then VP_SRC_HEIGHT should be half of its value as when LINE_SKIP is 0. NOTE: Min = 4	0

51.4.1.17 VP_DST_H_POSITION

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000_00000

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved	0
VP_DST_H_POSITION	[10:0]	RW	Horizontal offset in the display	0

51.4.1.18 VP_DST_V_POSITION

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0058, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	0
VP_DST_V_POSITION	[10:0]	RW	Vertical offset in the display	0

51.4.1.19 VP_DST_WIDTH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x005C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved	0
VP_DST_WIDTH	[10:0]	RW	Width of the display	0

51.4.1.20 VP_DST_HEIGHT

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	0
VP_DST_HEIGHT	[10:0]	RW	Height of the display	0

Figure 51-7 illustrates the video scaling and positioning on TV display.

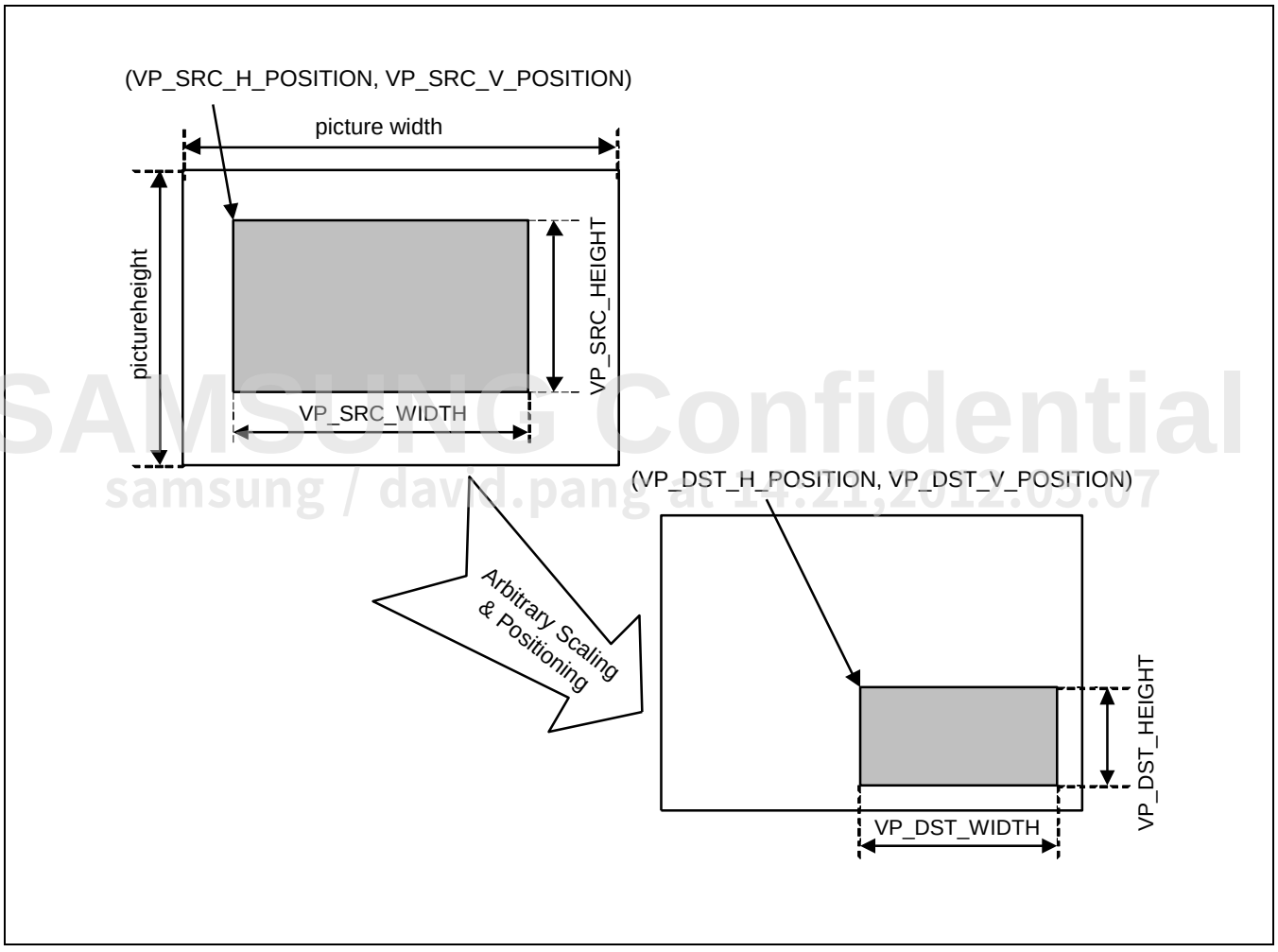


Figure 51-7 Video Scaling and Positioning on TV Display

51.4.1.21 VP_H_RATIO

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0064, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:19]	–	Reserved	0
VP_H_RATIO	[18:0]	RW	Horizontal Zoom Ratio of SRC:DST,3.16 format NOTE: (3.16) format means that "3" is an integer, "16" is a fraction. For example, SRC:DST = 1:2 Because of 16-bit fraction, it has to do 16 times left shift operation. As a result, register value is 1/2 × 2 ^ 16 = 0x8000	0

51.4.1.22 VP_V_RATIO

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0068, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:19]	–	Reserved	0
VP_V_RATIO	[18:0]	RW	Vertical Zoom Ratio of SRC:DST,3.16 format This register should be : • IPC disable, VP_V_RATIO = SRC/DST • IPC enable, VP_V_RATIO = 2 × SRC/DST The destination line number doubles by itself as it is a de-interlacing process. NOTE: (3.16) format means that "3" is an integer, "16" is a fraction. For example, SRC:DST = 1:2 Because of 16-bit fraction, it has to do 16 times left shift operation. As a result, register value is 1/2 × 2 ^ 16 = 0x8000	0

51.4.1.23 VP_POLY8_Y0_LL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x006C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0
vp_poly8_y0_ph0	[26:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:19]	–	Reserved	0
vp_poly8_y0_ph1	[18:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:11]	–	Reserved (Read as zero. Do not modify this bit.)	0
vp_poly8_y0_ph2	[10:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:3]	–	Reserved	0
vp_poly8_y0_ph3	[2:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.24 VP_POLY8_Y0_LH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0070, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0
vp_poly8_y0_ph4	[26:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:19]	–	Reserved	0
vp_poly8_y0_ph5	[18:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:11]	–	Reserved (Read as zero. Do not modify this bit.)	0
vp_poly8_y0_ph6	[10:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:3]	–	Reserved	0
vp_poly8_y0_ph7	[2:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.25 VP_POLY8_Y0_HL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0074, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0
vp_poly8_y0_ph8	[26:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:19]	–	Reserved	0
vp_poly8_y0_ph9	[18:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:11]	–	Reserved	0
vp_poly8_y0_ph10	[10:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:3]	–	Reserved	0
vp_poly8_y0_ph11	[2:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.26 VP_POLY8_Y0_HH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0078, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved	0
vp_poly8_y0_ph12	[26:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:19]	–	Reserved	0
vp_poly8_y0_ph13	[18:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:11]	–	Reserved	0
vp_poly8_y0_ph14	[10:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:3]	–	Reserved	0
vp_poly8_y0_ph15	[2:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.27 VP_POLY8_Y1_LL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x007C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0
vp_poly8_y1_ph0	[28:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:21]	–	Reserved	0
vp_poly8_y1_ph1	[20:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:13]	–	Reserved	0
vp_poly8_y1_ph2	[12:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:5]	–	Reserved	0
vp_poly8_y1_ph3	[4:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.28 VP_POLY8_Y1_LH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0
vp_poly8_y1_ph4	[28:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:21]	–	Reserved	0
vp_poly8_y1_ph5	[20:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:13]	–	Reserved	0
vp_poly8_y1_ph6	[12:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:5]	–	Reserved	0
vp_poly8_y1_ph7	[4:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.29 VP_POLY8_Y1_HL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0084, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0
vp_poly8_y1_ph8	[28:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:21]	–	Reserved	0
vp_poly8_y1_ph9	[20:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:13]	–	Reserved	0
vp_poly8_y1_ph10	[12:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:5]	–	Reserved	0
vp_poly8_y1_ph11	[4:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.30 VP_POLY8_Y1_HH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0088, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0
vp_poly8_y1_ph12	[28:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:21]	–	Reserved	0
vp_poly8_y1_ph13	[20:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:13]	–	Reserved	0
vp_poly8_y1_ph14	[12:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:5]	–	Reserved	0
vp_poly8_y1_ph15	[4:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.31 VP_POLY8_Y2_LL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x008C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly8_y2_ph0	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly8_y2_ph1	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly8_y2_ph2	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly8_y2_ph3	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.32 VP_POLY8_Y2_LH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly8_y2_ph4	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly8_y2_ph5	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly8_y2_ph6	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly8_y2_ph7	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.33 VP_POLY8_Y2_HL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0094, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly8_y2_ph8	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly8_y2_ph9	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly8_y2_ph10	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly8_y2_ph11	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.34 VP_POLY8_Y2_HH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0098, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly8_y2_ph12	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly8_y2_ph13	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly8_y2_ph14	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly8_y2_ph15	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.35 VP_POLY8_Y3_LL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x009C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly8_y3_ph0	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly8_y3_ph1	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly8_y3_ph2	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly8_y3_ph3	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.36 VP_POLY8_Y3_LH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly8_y3_ph4	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly8_y3_ph5	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly8_y3_ph6	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly8_y3_ph7	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.37 VP_POLY8_Y3_HL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x00A4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly8_y3_ph8	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly8_y3_ph9	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly8_y3_ph10	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly8_y3_ph11	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.38 VP_POLY8_Y3_HH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x00A8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly8_y3_ph12	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly8_y3_ph13	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly8_y3_ph14	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly8_y3_ph15	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.39 VP_POLY4_Y0_LL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x00EC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
vp_poly4_y0_ph0	[29:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:22]	–	Reserved	0
vp_poly4_y0_ph1	[21:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:14]	–	Reserved	0
vp_poly4_y0_ph2	[13:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:6]	–	Reserved	0
vp_poly4_y0_ph3	[5:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.40 VP_POLY4_Y0_LH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x00F0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
vp_poly4_y0_ph4	[29:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:22]	–	Reserved	0
vp_poly4_y0_ph5	[21:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:14]	–	Reserved	0
vp_poly4_y0_ph6	[13:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:6]	–	Reserved	0
vp_poly4_y0_ph7	[5:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.41 VP_POLY4_Y0_HL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x00F4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
vp_poly4_y0_ph8	[29:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:22]	–	Reserved	0
vp_poly4_y0_ph9	[21:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:14]	–	Reserved	0
vp_poly4_y0_ph10	[13:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:6]	–	Reserved	0
vp_poly4_y0_ph11	[5:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.42 VP_POLY4_Y0_HH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x00F8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
vp_poly4_y0_ph12	[29:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:22]	–	Reserved	0
vp_poly4_y0_ph13	[21:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:14]	–	Reserved	0
vp_poly4_y0_ph14	[13:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:6]	–	Reserved	0
vp_poly4_y0_ph15	[5:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.43 VP_POLY4_Y1_LL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x00FC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly4_y1_ph0	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly4_y1_ph1	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly4_y1_ph2	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly4_y1_ph3	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.44 VP_POLY4_Y1_LH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0100, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly4_y1_ph4	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly4_y1_ph5	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly4_y1_ph6	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly4_y1_ph7	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.45 VP_POLY4_Y1_HL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0104, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly4_y1_ph8	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly4_y1_ph9	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly4_y1_ph10	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly4_y1_ph11	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.46 VP_POLY4_Y1_HH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0108, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly4_y1_ph12	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly4_y1_ph13	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly4_y1_ph14	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly4_y1_ph15	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.47 VP_POLY4_Y2_LL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x010C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly4_y2_ph0	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly4_y2_ph1	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly4_y2_ph2	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly4_y2_ph3	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.48 VP_POLY4_Y2_LH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0110, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly4_y2_ph4	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly4_y2_ph5	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly4_y2_ph6	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly4_y2_ph7	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.49 VP_POLY4_Y2_HL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0114, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly4_y2_ph8	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly4_y2_ph9	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly4_y2_ph10	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly4_y2_ph11	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.50 VP_POLY4_Y2_HH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0118, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly4_y2_ph12	[30:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23]	–	Reserved	0
vp_poly4_y2_ph13	[22:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15]	–	Reserved	0
vp_poly4_y2_ph14	[14:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7]	–	Reserved	0
vp_poly4_y2_ph15	[6:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.51 VP_POLY4_Y3_LL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x011C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
vp_poly4_y3_ph0	[29:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:22]	–	Reserved	0
vp_poly4_y3_ph1	[21:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:14]	–	Reserved	0
vp_poly4_y3_ph2	[13:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:6]	–	Reserved	0
vp_poly4_y3_ph3	[5:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.52 VP_POLY4_Y3_LH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0120, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
vp_poly4_y3_ph4	[29:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:22]	–	Reserved	0
vp_poly4_y3_ph5	[21:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:14]	–	Reserved	0
vp_poly4_y3_ph6	[13:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:6]	–	Reserved	0
vp_poly4_y3_ph7	[5:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.53 VP_POLY4_Y3_HL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0124, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
vp_poly4_y3_ph8	[29:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:22]	–	Reserved	0
vp_poly4_y3_ph9	[21:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:14]	–	Reserved	0
vp_poly4_y3_ph10	[13:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:6]	–	Reserved	0
vp_poly4_y3_ph11	[5:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.54 VP_POLY4_Y3_LL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0128, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
vp_poly4_y3_ph12	[29:24]	RW	Poly-phase Filter Coefficients	0
RSVD	[23:22]	–	Reserved	0
vp_poly4_y3_ph13	[21:16]	RW	Poly-phase Filter Coefficients	0
RSVD	[15:14]	–	Reserved	0
vp_poly4_y3_ph14	[13:8]	RW	Poly-phase Filter Coefficients	0
RSVD	[7:6]	–	Reserved	0
vp_poly4_y3_ph15	[5:0]	RW	Poly-phase Filter Coefficients	0

51.4.1.55 VP_POLY4_C0_LL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x012C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved	0
vp_poly4_c0_ph0	[30:24]	RW	Signed 7-bit integer (– 64 to 63).	0
RSVD	[23]	–	Reserved	0
vp_poly4_c0_ph1	[22:16]	RW	Signed 7-bit integer (– 64 to 63).	0
RSVD	[15]	–	Reserved	0
vp_poly4_c0_ph2	[14:8]	RW	Signed 7-bit integer (– 64 to 63).	0
RSVD	[7]	–	Reserved	0
vp_poly4_c0_ph3	[6:0]	RW	Signed 7-bit integer (– 64 to 63).	0

Unlike VP_POLY4_Y registers, there are only a half of the coefficient registers for horizontal Chroma Scaler. The coefficients are assumed to be symmetric so that only half of them are kept. Some parts of them are unsigned integer and the other parts are signed integer. You must be careful while setting them.

51.4.1.56 VP_POLY4_C0_LH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0130, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
vp_poly4_c0_ph4	[29:24]	RW	Signed 6-bit integer (– 32 to 31).	0
RSVD	[23:22]	–	Reserved	0
vp_poly4_c0_ph5	[21:16]	RW	Signed 6-bit integer (– 32 to 31).	0
RSVD	[15:14]	–	Reserved	0
vp_poly4_c0_ph6	[13:8]	RW	Signed 6-bit integer (– 32 to 31).	0
RSVD	[7:6]	–	Reserved	0
vp_poly4_c0_ph7	[5:0]	RW	Signed 6-bit integer (– 32 to 31).	0

51.4.1.57 VP_POLY4_C0_HL

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0134, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved	0
vp_poly4_c0_ph8	[29:24]	RW	Signed 6-bit integer (– 32 to 31).	0
RSVD	[23:22]	–	Reserved	0
vp_poly4_c0_ph9	[21:16]	RW	Signed 6-bit integer (– 32 to 31).	0
RSVD	[15:14]	–	Reserved	0
vp_poly4_c0_ph10	[13:8]	RW	Signed 6-bit integer (– 32 to 31).	0
RSVD	[7:6]	–	Reserved	0
vp_poly4_c0_ph11	[5:0]	RW	Signed 6-bit integer (– 32 to 31).	0

51.4.1.58 VP_POLY4_C0_HH

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0138, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0
vp_poly4_c0_ph12	[28:24]	RW	Signed 5-bit integer (– 16 to 15).	0
RSVD	[23:21]	–	Reserved	0
vp_poly4_c0_ph13	[20:16]	RW	Signed 5-bit integer (– 16 to 15).	0
RSVD	[15:13]	–	Reserved	0
vp_poly4_c0_ph14	[12:8]	RW	Signed 5-bit integer (– 16 to 15).	0
RSVD	[7:5]	–	Reserved	0
vp_poly4_c0_ph15	[4:0]	RW	Signed 5-bit integer (– 16 to 15).	0

51.4.1.59 PP_CSC_Y2Y_COEF

- Base Address: 0x12C0_0000
- Address = Base Address + 0x01D4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0
PP_CSC_Y2Y_COEF	[11:0]	RW	Specifies BT.601 to BT.709 or BT.709 to BT.601 CSC Coefficients for Y to Y [11]: Sign bit [10]: Integer bit [9:0]: Fraction bit 0x7FF = 1.999 … 0x400 = 1.0 … 0x0 = 0 0Xfff = – 0.0001 … 0xC00 = – 1.0 … 0x800 = – 2.0	0

- The equations for BT.601 to BT.709 Color Space Conversion Matrix are:
  - Y709 = 1.0 × Y601 – 0.118188 × Cb601 – 0.212685 × Cr601
  - Cb709 = 0.0 × Y601 + 1.018640 × Cb601 – 0.114618 × Cr601
  - Cr709 = 0.0 × Y601 + 0.075049 × Cb601 + 1.025327 × Cr601

- The equations for BT.709 to BT.601 Color Space Conversion Matrix are:
  - Y601 = 1.0 × Y709 + 0.101579 × Cb709 + 0.196076 × Cr709
  - Cb601 = 0.0 × Y709 + 0.989854 × Cb709 – 0.110653 × Cr709
  - Cr601 = 0.0 × Y709 – 0.072453 × Cb709 + 0.983398 × Cr709

- These equations are written without interface offsets of "+ 16" for Luminance and "+ 128" for Chrominance.

- CSC module computes these equations without '+ 128' offset for Chrominance, and generates final CSC results with "+ 128" offset.

In case of two Luminance equations, "+ 16" offset is selectable by control register (PP_CSC_EN[1]).
If Y offset (+ 16) exists in matrix input data, the coefficient of above equations should be redefined

51.4.1.60 PP_CSC_CB2Y_COEF

- Base Address: 0x12C0_0000
- Address = Base Address + 0x01D8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0
PP_CSC_CB2Y_COEF	[11:0]	RW	Specifies BT.601 to BT.709 or BT.709 to BT.601 CSC Coefficients for CB to Y	0

51.4.1.61 PP_CSC_CR2Y_COEF

- Base Address: 0x12C0_0000
- Address = Base Address + 0x01DC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0
PP_CSC_CR2Y_COEF	[11:0]	RW	Specifies BT.601 to BT.709 or BT.709 to BT.601 CSC Coefficients for CR to Y	0

51.4.1.62 PP_CSC_Y2CB_COEF

- Base Address: 0x12C0_0000
- Address = Base Address + 0x01E0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0
PP_CSC_Y2CB_COEF	[11:0]	RW	Specifies BT.601 to BT.709 or BT.709 to BT.601 CSC Coefficients for Y to CB	0

51.4.1.63 PP_CSC_CB2CB_COEF

- Base Address: 0x12C0_0000
- Address = Base Address + 0x01E4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0
PP_CSC_CB2CB_COEF	[11:0]	RW	Specifies BT.601 to BT.709 or BT.709 to BT.601 CSC Coefficients for CB to CB	0

51.4.1.64 PP_CSC_CR2CB_COEF

- Base Address: 0x12C0_0000
- Address = Base Address + 0x01F0, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0
PP_CSC_CR2CB_COEF	[11:0]	RW	Specifies BT.601 to BT.709 or BT.709 to BT.601 CSC Coefficients for CR to CB	0

51.4.1.65 PP_CSC_Y2CR_COEF

- Base Address: 0x12C0_0000
- Address = Base Address + 0x01EC, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0
PP_CSC_Y2CR_COEF	[11:0]	RW	Specifies BT.601 to BT.709 or BT.709 to BT.601 CSC Coefficients for Y to CR	0

51.4.1.66 PP_CSC_CB2CR_COEF

- Base Address: 0x12C0_0000
- Address = Base Address + 0x01E8, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0
PP_CSC_CB2CR_COEF	[11:0]	RW	Specifies BT.601 to BT.709 or BT.709 to BT.601 CSC Coefficients for CB to CR	0

51.4.1.67 PP_CSC_CR2CR_COEF

- Base Address: 0x12C0_0000
- Address = Base Address + 0x01F4, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved	0
PP_CSC_CR2CR_COEF	[11:0]	RW	Specifies BT.601 to BT.709 or BT.709 to BT.601 CSC Coefficients for CR to CR	0

51.4.1.68 PP_BYPASS

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0200, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0
PP_BYPASS	[0]	RW	Disables Post Image Processor 0 = Enables 1 = Disables (default) NOTE: This functionality is only for SD (480p/576p). We do not recommend you to use functions for HD. The post image processor executes color saturation control, sharpness enhancement, contrast and brightness control.	1

51.4.1.69 PP_SATURATION

- Base Address: 0x12C0_0000
- Address = Base Address + 0x020C, Reset Value = 0x0000_0080

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0
PP_SATURATION	[7:0]	RW	Color Saturation Factor, unsigned 1.7 format 0x00 = 0.0 … 0x80 = 1.0 … 0xFF = 1.992188, (128 × 2 – 1)/128	80

51.4.1.70 PP_SHARPNESS

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0210, Reset Value = 0x0000_0500

Name	Bit	Type	Description	Reset Value
RSVD	[31:17]	–	Reserved	0
PP_DISPLAY_DEVICE	[16]	RW	Selects SD (480p/576p) or HD device for display 0 = SD 1 = HD	0
PP_TH_HNOISE	[15:8]	RW	Specifies threshold value setting that decides the minimum vertical edge value.	0x5
RSVD	[7:2]	–	Reserved	0
PP_SHARPNESS	[1:0]	RW	Specifies control for edge enhancement 0 = No effect 1 = Minimum edge enhancement 2 = Moderate edge enhancement 3 = Maximum edge enhancement	0

51.4.1.71 PP_LINE_EQn (n = 0 to 7)

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0218, Reset Value = 0x0000_0000
- Address = Base Address + 0x021C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0220, Reset Value = 0x0000_0000
- Address = Base Address + 0x0224, Reset Value = 0x0000_0000
- Address = Base Address + 0x0228, Reset Value = 0x0000_0000
- Address = Base Address + 0x022C, Reset Value = 0x0000_0000
- Address = Base Address + 0x0230, Reset Value = 0x0000_0000
- Address = Base Address + 0x0234, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0
LINE_INTC	[23:8]	RW	Line Intercept, signed 9.7 format NOTE: (9.7) format means that "1" is a signed bit, "8" is a integer, "7" is a fraction. For example, INTC = 3 Because to the 7-bit fraction, it has to do 7 times left shift operation. As a result, register value is 3 × 2 ^ 7 = 0x18000	0
LINE_SLOPE	[7:0]	RW	Line Slope, unsigned 1.7 format Due to 1-bit integer, the LINE_SLOPE falls in the range of 0 to 1.9921875. NOTE: (1.7) format means that "1" is a integer, "7" is a fraction. For example, LINE_SLOPE = 0.5 = 1 × 2 ^ (– 1) Because of the 7-bit fraction, it has to do 7 times left shift operation. As a result, register value is 1/2 × 2 ^ 7 = 0x40	0

Eight equations are related with *Figure 51-8* Input luminance value between 0 and 255 (it is divide by 8 steps). Each of them is matched with each of 8 equations. Thus, we can make the new curve of the contrast and luminance as using 8 equation's combination. Each equation is matched like following:

1. PP_LINE_EQ0 = LINE_SLOPE0 × Y + LINE_INTC0 (0 ≤ Y ≤ 31)
2. PP_LINE_EQ1 = LINE_SLOPE1 × Y + LINE_INTC1 (32 ≤ Y ≤ 63)
3. PP_LINE_EQ2 = LINE_SLOPE2 × Y + LINE_INTC2 (64 ≤ Y ≤ 95)
4. PP_LINE_EQ3 = LINE_SLOPE3 × Y + LINE_INTC3 (96 ≤ Y ≤ 127)
5. PP_LINE_EQ4 = LINE_SLOPE4 × Y + LINE_INTC4 (128 ≤ Y ≤ 159)
6. PP_LINE_EQ5 = LINE_SLOPE5 × Y + LINE_INTC5 (160 ≤ Y ≤ 191)
7. PP_LINE_EQ6 = LINE_SLOPE6 × Y + LINE_INTC6 (192 ≤ Y < 223)
8. PP_LINE_EQ7 = LINE_SLOPE7 × Y + LINE_INTC7 (224 ≤ Y ≤ 255)

51.4.1.72 PP_BRIGHT_OFFSET

- Base Address: 0x12C0_0000
- Address = Base Address + 0x0238, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	0
PP_BRIGHT_OFFSET	[8:0]	RW	Offset for Y Brightness Control, signed 1.8 format Bright enhanced Y = Org Y + BRIGHT_OFFSET 0xFF = + 255 … 0x1 = + 1 0x0 = 0 0x1FF = – 1 … 0x100 = – 256	0

Figure 51-8 illustrates examples of how VP controls brightness and contrast of image sequence using PP_LINE_EQ0 to PP_LINE_EQ7 registers and PP_BRIGHT_OFFSET register. *Figure 51-8* luminance mapping curve is approximated by 8 sub-lines described from PP_LINE_EQ0 to PP_LINE_EQ7. Consequently, brightness and contrast are controlled in very flexible way.

Figure 51-8 illustrates the image brightness and contrast control.

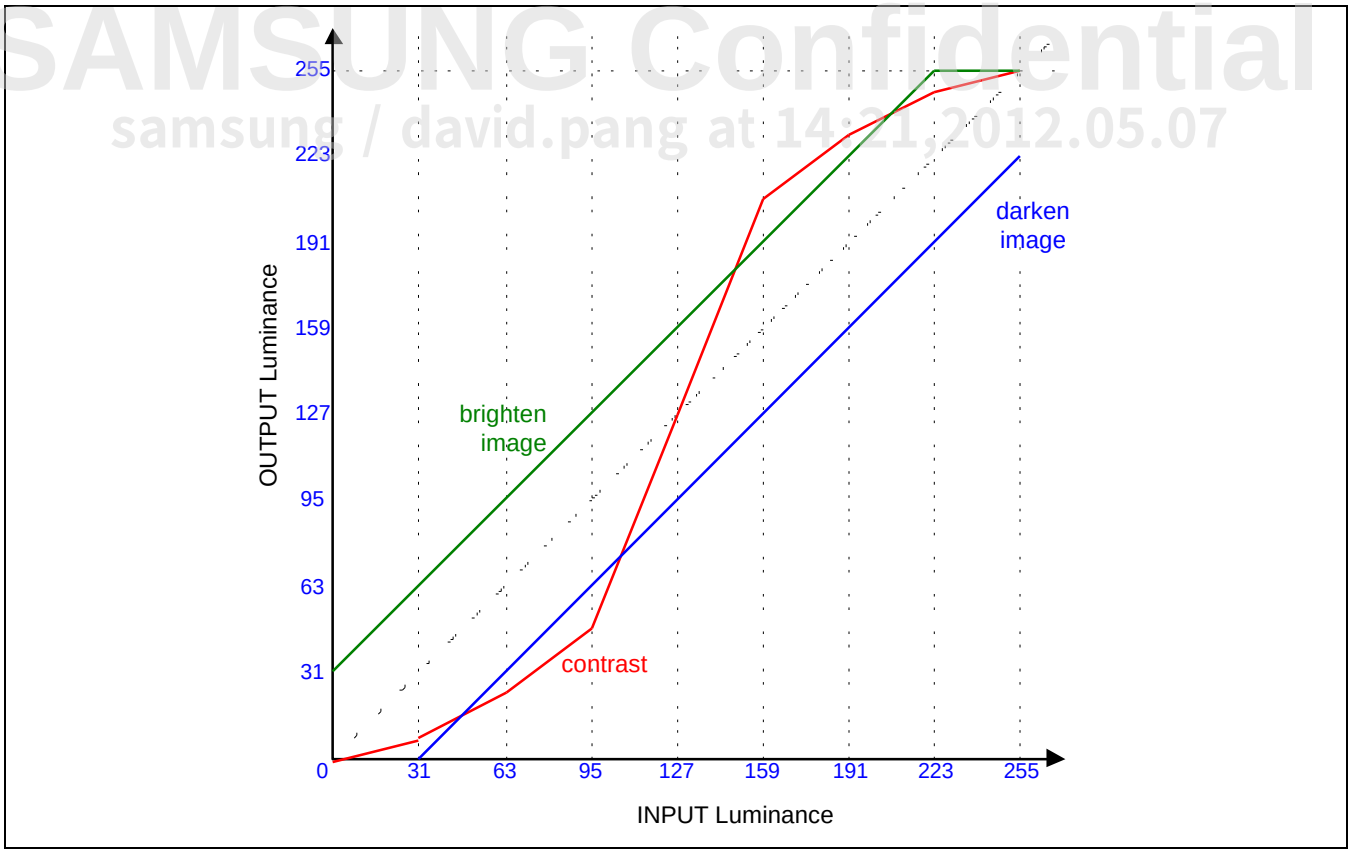


Figure 51-8 Image Brightness and Contrast Control

51.4.1.73 PP_CSC_EN

- Base Address: 0x12C0_0000
- Address = Base Address + 0x023C, Reset Value = 0x0000_0002

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved	0
SUB_Y_OFFSET_EN	[1]	RW	Y Offset Control for Color Space Conversion 0 = Disable 1 = "+16" offset enable NOTE: If "SUB_Y_OFFSET_EN" is "1", Y', Cb', Cr' are given by: • Y' = (Y-16) × Y2Y_coef + (Cb-128) × Cb2Y_coef + (Cr-128) × Cr2Y_coef • Cb' = (Y-16) × Y2Cb_coef + (Cb-128) × Cb2Cb_coef + (Cr-128) × Cr2Cb_coef • Cr' = (Y-16) × Y2Cr_coef + (Cb-128) × Cb2Cr_coef + (Cr-128) × Cr2Cr_coef else, Y', Cb', Cr' are given by: • Y' = Y × Y2Y_coef + (Cb-128) × Cb2Y_coef + (Cr-128) × Cr2Y_coef • Cb' = Y × Y2Cb_coef + (Cb-128) × Cb2Cb_coef + (Cr-128) x Cr2Cb_coef • Cr' = Y × Y2Cr_coef + (Cb-128) × Cb2Cr_coef + (Cr-128) × Cr2Cr_coef	1
CSC_EN	[0]	RW	Color Space Conversion(CSC) Enable Control 0 = Disables CSC 1 = Enables CSC	0

51.4.1.74 VP_VERSION_INFO

- Base Address: 0x12C0_0000
- Address = Base Address + 0x03FC, Reset Value = 0x0000_0011

Name	Bit	Type	Description	Reset Value
VERSION_INFO	[31:0]	RW	VP Version Information	0x0000_0010

51.4.2 The Idea of Poly-phase Filtering in Video Processor

Figure 51-9 shows basic concept of poly-phase filtering in video processor, in case of 4-tap vertical luminance filter. Pixels highlighted in grey color are from decoded pictures and used to interpolate the dotted pixels. They are transferred to MIXER. The vertical positions of pixels to be interpolated are calculated by using VP_SRC_V_POSITION and VP_V_RATIO. Once the vertical position is calculated, the nearest pixel phase (with 1/16 resolution) and which pixels are used for interpolation are decided.

Figure 51-9 illustrates the 4-tap vertical poly-phase filter.

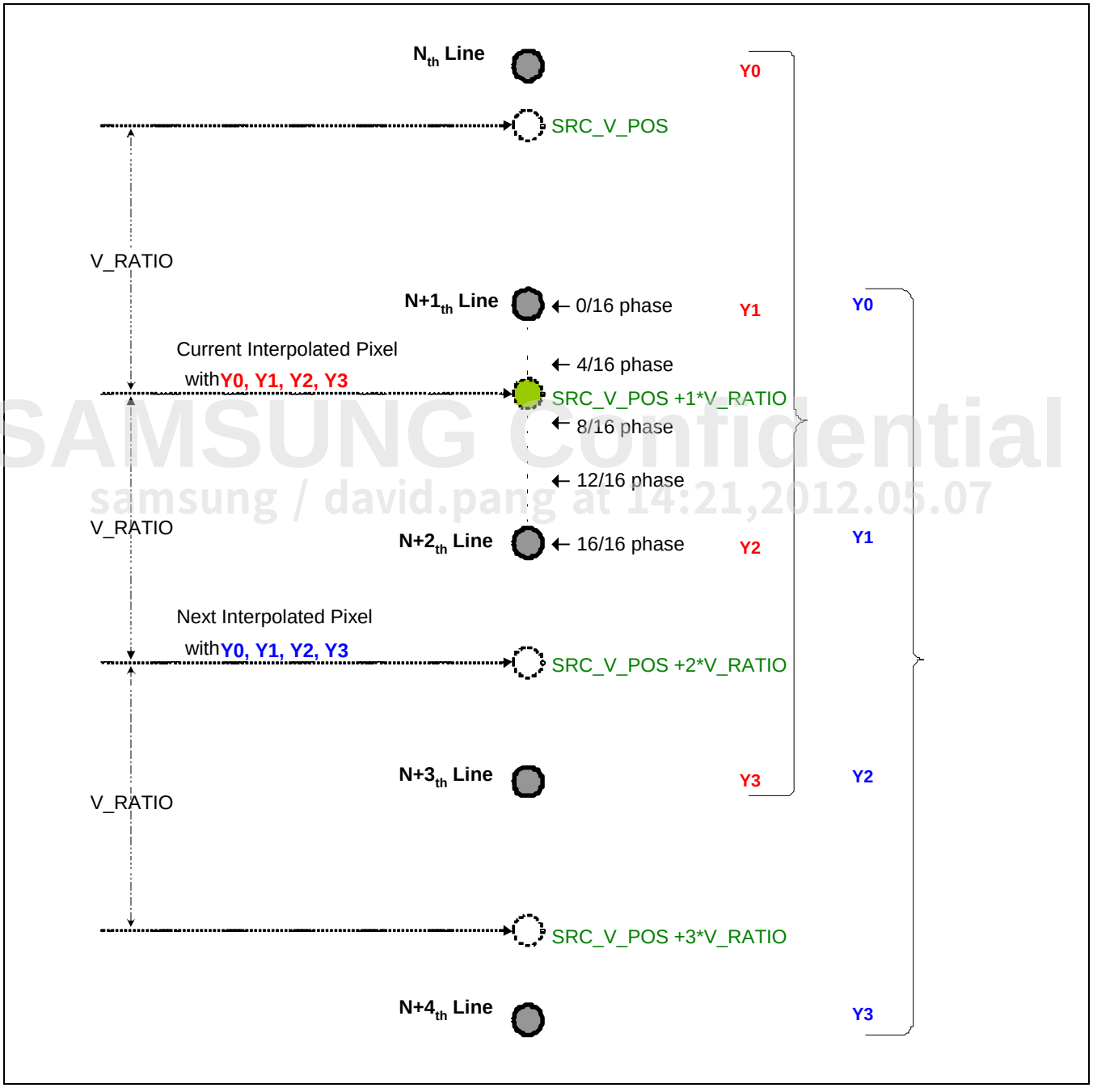


Figure 51-9 4-Tap Vertical Poly-phase Filter

If the calculated vertical position is 10.45 (for example, pixels of 9th, 10th, 11th, and 12th lines are used for poly-phase filtering and the pixel phase is 7/16), then it means the filter coefficients are vp_poly4_y0_ph7, vp_poly4_y1_ph7, vp_poly4_y2_ph7, and vp_poly4_y3_ph7.

The 8-tap luminance horizontal poly-phase filter and 4-tap chrominance horizontal poly-phase filter use the exact same scheme.

Some pixels in the filter window are not available at the top, bottom, left, and right boundaries of the pictures. In this case, the value of the nearest pixel repeats.

Figure 51-10 illustrates the pixel repetition at picture boundary.

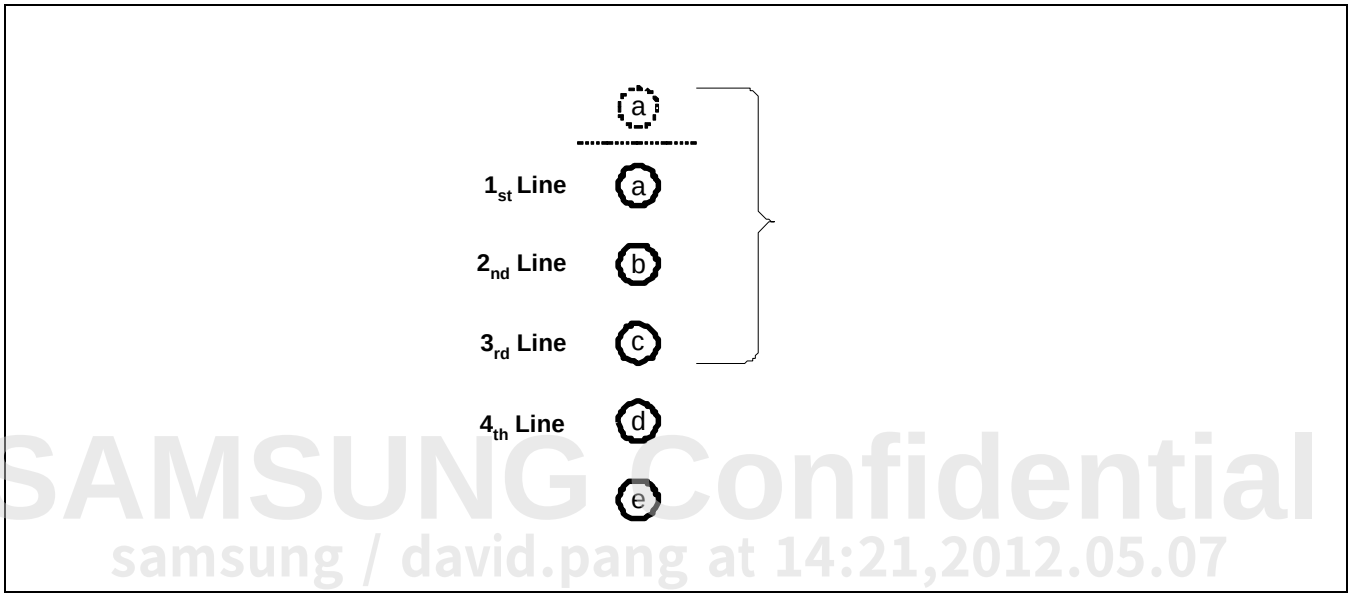


Figure 51-10 Pixel Repetition at Picture Boundary

52 Mixer

52.1 Overview of Mixer

Mixer overlaps or blends the input data such as graphic, video, background and sends the resulting data to the TVOUT module. The TVOUT module generates all the video control signals.

52.2 Features

The features of mixer are:

- Supports AXI Master and AHB Slave interface
- AXI Master interface for graphic layer data fetch
- AHB Slave interface for control register setup
- Supports Little Endian and Big Endian data format at graphic layers
- Multiple layers of mixer are:
  - Background layer
  - Graphic0 layer
  - Graphic1 layer
  - Video layer
- Input control features of mixer are:
  - Blending between each layers
  - Selectable graphic layer frame buffer
  - Enable/Disable each layer
  - Source cropping for graphic layer
- Supports overlapping and blending input layers
- Supports YCbCr 4:4:4 /RGB 8:8:8 format as output
- Supports interlace and progressive scan
- Supports 480i/p, 576i/p, 720p and 1080i/1080p display sizing (1080p is 60 Hz.)
- Supports Graphic Layer 0 and Graphic Layer 1
- Supports an external DRAM frame buffer memory source

- Color formats that are differently configurable between each graphic layer are:
  - 16 bpp Direct RGB[565]
  - 16 bpp Direct ARGB[1555]
  - 16 bpp Direct ARGB[4444]
  - 32 bpp Direct ARGB[8888]
- Maximum graphic layer size are:
  - 480i/p: 720 × 480 pixel
  - 576i/p: 720 × 576 pixel
  - 720p: 1280 × 720 pixel
  - 1080i/p: 1920 × 1080 pixel
- Blending
  - Maximum 256 level pixel and layer blending
  - Separate configurable layer blending factor between each layer
- Scale
  - Vertical line duplication: x2
  - Horizontal win-scale: x2
- Video layer
- Source: Video processor module
- Color Format: 24 bpp Direct YCbCr[888]
- Maximum Resolution
  - 480i/p: 720 × 480 pixel
  - 576i/p: 720 × 576 pixel
  - 720p: 1280 × 720 pixel
  - 1080i/p: 1920 × 1080 pixel
- Supports xvYcc limiter.
- Background layer
- Source: Configuration register
- Lowest layer
- Layer ordering
- Background → (Video ↔ Graphic0 ↔ Graphic1)
- Supports complete ordering and blending of video layer and two graphic layers
- Blending
  - Supports pixel blending and layer blending
  - Supports alpha blending
  - Supports pre-multiplied blending

52.3 Block Diagram

Figure 52-1 illustrates the block diagram of mixer.

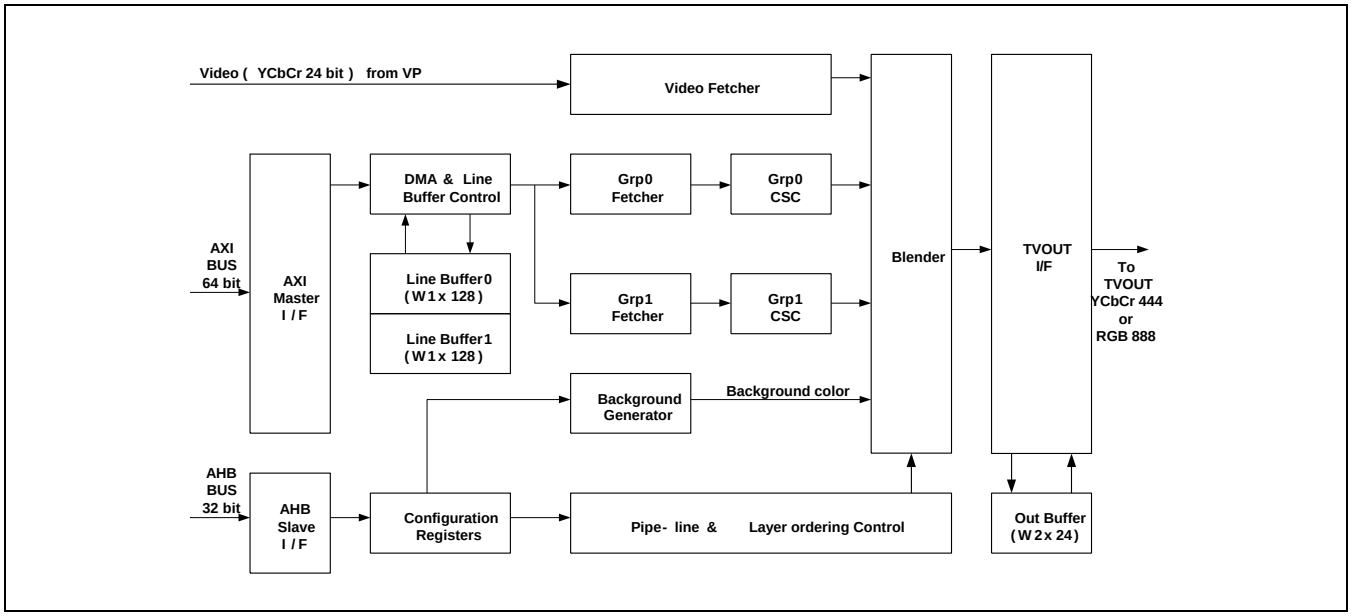
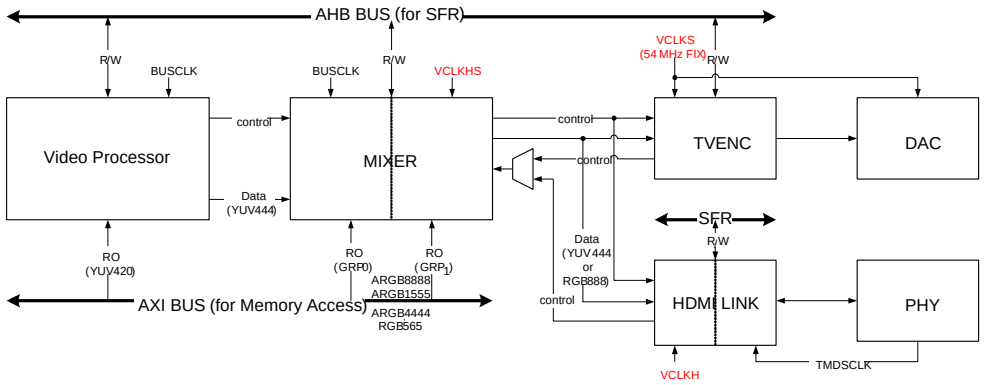


Figure 52-1 Block Diagram of Mixer

- AXI Master I/F, DMA and Line Buffer Control:
  These blocks fetch data from memory and store data in Line Buffer 0 and Line Buffer 1.
- AHB Slave I/F and Configuration Registers:
  These blocks contain Special Function Registers (SFRs) to control mixer.
- Video Fetcher:
  This block fetches data from VP module.
- Grp0/1 Fetcher and Grp0/1 Color Space Conversion (CSC):
  Grp0/1 Fetcher pops up data from Line Buffer0/Line Buffer1 and delivers image data to Blender block through the respective color space converters (Grp0/1 CSC).
- Background Generator:
  This block generates background patterns according to the configurations.
- Blender:
  The role of this block is to mix four image-layers. They are Video, Graphic0, Graphic1, and Background.
- TVOUT I/F:
  This block temporarily stores data from Blender until TVENC or HDMI requests data.

52.3.1 Video Clock Relation

Figure 52-2 illustrates the TV sub-system block diagram and usage frequency.



VCLKHS(or VCLKH) usage frequency

Vertical Freq	Format	Pixel Freq
59.94Hz	480P	27MHz(or54MHz)
	720P	74.175MHz
	1080I	74.175MHz
60.00Hz	480P	27.027MHz(or54.054MHz)
	720P	74.250MHz
	1080I	74.250MHz
50.00Hz	576P	27MHz(or54MHz)
	720P	74.250MHz
	1080I	74.250MHz
29.97Hz	1080P	74.175MHz
30Hz	1080P	74.250MHz
25Hz	1080P	74.250MHz
Summary		27MHz
		27.027MHz
		54MHz
		54.054MHz
		74.175MHz
		74.250MHz

Figure 52-2 TV Sub-System Block Diagram and Usage Frequency

Configure the REG_DST_SEL register for HDMI-OUT selection. After configuring, set the same clock configuration for MIXER I/F clock (VCLKHS) and HDMI pixel clock (VCLKH). The embedded PLL in HDMI PHY generates a clock that should be selected using the Usage Frequency Table. Refer to VCLKHS or VCLKH for more information. You can configure clock source by using CLK_SRC1 register.

52.4 Register Description

52.4.1 Register Map Summary

- Base Address: 0x12C1_0000

Register	Offset	Description	Reset Value
Mixer Global Setting			
MIXER_STATUS	0x0000	Specifies status of mixer operation	0x0000_0006
MIXER_CFG	0x0004	Specifies mode setting of mixer	0x0000_0000
Mixer Interrupt			
MIXER_INT_EN	0x0008	Specifies mixer interrupt Enable	0x0000_0000
MIXER_INT_STATUS	0x000C	Specifies mixer interrupt status	0x0000_0000
Video and Blender Configuration			
MIXER_LAYER_CFG	0x0010	Specifies video and graphic layer priority and on/off of mixer	0x0000_0000
MIXER_VIDEO_CFG	0x0014	Specifies the video layer configuration of mixer	0x0000_0000
MIXER_VIDEO_LIMITER_PARA_CFG	0x0018	Specifies parameter of video layer limiter configuration of mixer	0xEB10_F010
Graphic0 Layer Configuration			
MIXER_GRAPHIC0_CFG	0x0020	Specifies graphic layer0 configuration of mixer	0x0000_0000
MIXER_GRAPHIC0_BASE	0x0024	Specifies base address of graphic layer0 of mixer	0x0000_0000
MIXER_GRAPHIC0_SPAN	0x0028	Specifies span for graphic layer0 of mixer	0x0000_0000
MIXER_GRAPHIC0_SXY	0x002C	Specifies source X/Y positions of graphic layer0 of mixer	0x0000_0000
MIXER_GRAPHIC0_WH	0x0030	Specifies width/ height for graphic layer0 of mixer	0x0000_0000
MIXER_GRAPHIC0_DXY	0x0034	Specifies destination X/Y positions for graphic layer0 of mixer	0x0000_0000
MIXER_GRAPHIC0_BLANK	0x0038	Specifies blank pixel value for graphic layer0 of mixer	0x0000_0000
Graphic1 Layer Configuration			
MIXER_GRAPHIC1_CFG	0x0040	Specifies graphic layer1 configuration of mixer	0x0000_0000
MIXER_GRAPHIC1_BASE	0x0044	Specifies base address for graphic layer1 of mixer	0x0000_0000
MIXER_GRAPHIC1_SPAN	0x0048	Specifies span for graphic layer1 of mixer	0x0000_0000
MIXER_GRAPHIC1_SXY	0x004C	Specifies source X/Y positions for graphic layer1 of mixer	0x0000_0000
MIXER_GRAPHIC1_WH	0x0050	Specifies width/ height for graphic layer1 of mixer	0x0000_0000
MIXER_GRAPHIC1_DXY	0x0054	Specifies destination X/Y positions for graphic layer1 of mixer	0x0000_0000

Register	Offset	Description	Reset Value
MIXER_GRAPHIC1_BLANK	0x0058	Specifies blank pixel value for graphic layer1 of mixer	0x0000_0000
Background Layer Configuration			
MIXER_BG_COLOR0	0x0064	Specifies mixer background color of first point	0x0000_0000
MIXER_BG_COLOR1	0x0068	Specifies mixer background color of second point	0x0000_0000
MIXER_BG_COLOR2	0x006C	Specifies mixer background color of last point	0x0000_0000
Color Space Conversion Coefficient			
MIXER_CM_COEFF_Y	0x0080	Specifies scaled color space conversion (RGB to Y) co-efficient for graphic layer of mixer	0x0844_0832
MIXER_CM_COEFF_CB	0x0084	Specifies scaled color space conversion (RGB to CB) co-efficient for graphic layer of mixer	0x3b5d_b0e1
MIXER_CM_COEFF_CR	0x0088	Specifies scaled color space conversion (RGB to Cr) co-efficient for graphic layer of mixer	0x0e1d_13dc
Mixer Global Setting Shadowing Register			
MIXER_TVOUT_CFG	0x0100	Specifies stereo scopic mode of mixer	0x0000_0000
MIXER_3D_ACTIVE_VIDEO	0x0104	Specifies active video of mixer	0x0000_0000
MIXER_3D_ACTIVE_SPACE	0x0108	Specifies active space of mixer	0x0000_0000
Mixer Global Setting Shadowing Register			
MIXER_STATUS_S	0x2000	Specifies status of mixer operation (shadow)	0x0000_0006
MIXER_CFG_S	0x2004	Specifies mixer mode setting (shadow)	0x0000_0000
Video and Blender Configuration Shadowing Register			
MIXER_LAYER_CFG_S	0x2010	Specifies video and graphic layer priority and on/ off (shadow) of mixer	0x0000_0000
MIXER_VIDEO_CFG_S	0x2014	Specifies video layer configuration (shadow) of mixer	0x0000_0000
MIXER_VIDEO_LIMITER_PARA_CFG_S	0x2018	Specifies parameter of video layer limiter configuration of mixer	0xEB10_F010
Graphic0 Layer Configuration Shadowing Register			
MIXER_GRAPHIC0_CFG_S	0x2020	Specifies graphic layer0 configuration (shadow) of mixer	0x0000_0000
MIXER_GRAPHIC0_BASE_S	0x2024	Specifies graphic0 base address (shadow) of mixer	0x0000_0000
MIXER_GRAPHIC0_SPAN_S	0x2028	Specifies graphic0 span (shadow) of mixer	0x0000_0000
MIXER_GRAPHIC0_SXY_S	0x202C	Specifies graphic0 source X/Y coordinates (shadow) of mixer	0x0000_0000
MIXER_GRAPHIC0_WH_S	0x2030	specifies graphic0 width/ height (shadow) of mixer	0x0000_0000
MIXER_GRAPHIC0_DXY_S	0x2034	Specifies graphic0 destination X/Y coordinates	0x0000_0000

Register	Offset	Description	Reset Value
		(shadow) of mixer	
MIXER_GRAPHIC0_BLANK_PIXEL_S	0x2038	Specifies graphic0 blank pixel (shadow) of mixer	0x0000_0000
Graphic1 Layer Configuration Shadowing Register			
MIXER_GRAPHIC1_CFG_S	0x2040	Specifies graphic layer1 configuration (shadow) of mixer	0x0000_0000
MIXER_GRAPHIC1_BASE_S	0x2044	Specifies graphic1 base address (shadow) of mixer	0x0000_0000
MIXER_GRAPHIC1_SPAN_S	0x2048	Specifies graphic1 span (shadow) of mixer	0x0000_0000
MIXER_GRAPHIC1_SXY_S	0x204C	Specifies graphic1 source X/Y coordinates (shadow) of mixer	0x0000_0000
MIXER_GRAPHIC1_WH_S	0x2050	Specifies graphic1 width/ height (shadow) of mixer	0x0000_0000
MIXER_GRAPHIC1_DXY_S	0x2054	Specifies graphic1 destination X/Y coordinates (shadow) of mixer	0x0000_0000
MIXER_GRAPHIC1_BLANK_PIXEL_S	0x2058	Specifies graphic1 blank pixel (shadow) of mixer	0x0000_0000
Background Layer Configuration Shadowing Register			
MIXER_BG_COLOR0_S	0x2064	Specifies background first color (shadow) of mixer	0x0000_0000
MIXER_BG_COLOR1_S	0x2068	Specifies background second color (shadow) of mixer	0x0000_0000
MIXER_BG_COLOR2_S	0x206C	Specifies background last color (shadow) of mixer	0x0000_0000
Version Register			
MIXER_VER	0x3100	Specifies mixer version	0x0000_0011

52.4.2 Shadow Registers (Read Only)

If SYNC_ENABLE signal is set to 1, then the written values to internal registers does not apply directly to the mixer operation. It stores the written values temporarily in the internal registers and waits for the next v_sync signal. When the v_sync signal occurs, it updates the stored internal register values to the shadow registers.

In interlaced display mode, it updates the shadow registers only at top-field. In progressive display mode, it updates the shadow registers at every v-sync.

52.4.2.1 MIXER_STATUS

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0006

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved (Read as zero. Do not modify this bit.)	0
16_BURST_MODE	[7]	RW	Specifies 16 Burst Mode(64-bit bus) in DMA 0 = Enables 8-beat Burst Mode 1 = Enables 16-beat Burst Mode	0
RSVD	[6:4]	–	Reserved	0
BIG_ENDIAN	[3]	RW	Specifies Big Endian source format 0 = Little Endian source format 1 = Big Endian source format	0
SYNC_ENABLE	[2]	RW	Specifies Sync Enable 0 = It will not apply the values you set, to the mixer operation although it detects v_sync 1 = It applies the values you set to the mixer operation after detecting v_sync	1
MIXER_OPERATION_STATUS	[1]	RW	Specifies Mixer Operation Status 0 = Mixer is operating 1 = Mixer is idle mode NOTE: To stop operation you should set REG_RUN to "0" and check whether this bit is "1" This bit is Read only.	1
REG_RUN	[0]	RW	Specifies Mixer Operation Control0 = Mixer stops processing 1 = Mixer starts processing The register updates after it generates V_SYNC signal. NOTE: The VSYNC of timing generator of HDMI updates the SFRs of Video Processor and Image Mixer. Therefore, you should configure the SFRs before enabling this bit. The sequence to enable TVSS is VP → MIXER → HDMI. Similarly, the disabling sequence is VP → MIXER → HDMI.	0

52.4.2.2 MIXER_CFG

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved (Read as zero. Do not modify this bit.)	0
REG_RGB_ FORMAT	[10:9]	RW	Specifies range selection of RGB. 0 = RGB601 (0 to 255) 1 = RGB601 (16 to 235) 2 = RGB709 (0 to 255) 3 = RGB709 (16 to 235)	0
REG_OUT_TYPE	[8]	RW	Specifies output type selection of the mixer 0 = YCbCr444 1 = RGB888	0
REG_DST_SEL	[7]	RW	You should set this value as "1"	0
REG_HD_MODE	[6]	RW	Specifies mode selection between 720p or 1080i. 0 = 720p 1 = 1080i/1080p NOTE: 1080i = REG_SCAN_MODE is "0" 1080p = REG_SCAN_MODE is "1"	0
REG_GRAPHIC1_ EN	[5]	RW	Specifies Graphic1 layer display control bit. 0 = Disables 1 = Enables	0
REG_GRAPHIC0_ EN	[4]	RW	Specifies Graphic0 layer display control bit. 0 = Disables 1 = Enables	0
REG_VIDEO_EN	[3]	RW	Specifies video layer display control bit. 0 = Disables 1 = Enables	0
REG_SCAN_MODE	[2]	RW	Specifies display scanning mode of TV. 0 = Interlaced mode 1 = Progressive mode	0
REG_NTSC_PAL	[1]	RW	Specifies display standard of TV 0 = NTSC (720 × 480) 1 = PAL (720 × 576) If you set the REG_NTSC_PAL bit to 0 and REG_SCAN_MODE bit to 1, then the output is Call 480p Standard. This is valid only if REG_HD_SD bit is set to 0.	0
REG_HD_SD	[0]	RW	Specifies HD or SD selection 0 = SD 1 = HD If REG_HD_SD is set to 1, then the REG_HD_MODE = 0 for 720p REG_HD_MODE = 1 for 1080i.	0

NOTE: All changes to this register are valid on a vertical sync signal of next frame.

–	Wide	Narrow
CSCY2R (601)	R = Y + 1.371 (Cr-128) G = Y – 0.698 (Cr-128) – 0.336 (Cb-128) B = Y + 1.732 (Cb-128)	R = 1.164 (Y-16) + 1.596 (Cr-128) G = 1.164 (Y-16) – 0.813 (Cr-128) – 0.391 (Cb-128) B = 1.164 (Y-16) + 2.018 (Cb – 128)
CSCY2R (709)	R = Y + 1.540 (Cr-128) G = Y – 0.459 (Cr-128) – 0.183 (Cb-128) B = Y + 1.816 (Cb-128)	R = 1.164 (Y-16) + 1.793 (Cr-128) G = 1.164 (Y-16) – 0.534 (Cr-128) – 0.213 (Cb-128) B = 1.164 (Y-16) + 2.115 (Cb-128)

NOTE: This table refers to Video Demystified (Keith Jack).

52.4.2.3 MIXER_INT_EN

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved (Read as zero. Do not modify this bit.)	0
INT_EN_VSYNC	[11]	RW	Specifies VSYNC interrupt enable 0 = Disables interrupt 1 = Enables interrupt This bit is Write only.	0
INT_EN_VP	[10]	RW	Specifies VP underflow interrupt enable 0 = Disables interrupt 1 = Enables interrupt If this bit is set to 0, then it disables the interrupt request to host controller. It does not mask the change in MIXER_INT_STATUS [10] bit status	0
INT_EN_GRP1	[9]	RW	Specifies graphic layer1 line buffer underflow Interrupt Enable 0 = Disables interrupt 1 = Enables interrupt If this bit is set to 0, then it disables the interrupt request to host controller. It does not mask the change in the MIXER_INT_STATUS [9] bit status	0
INT_EN_GRP0	[8]	RW	Specifies graphic layer0 line buffer underflow interrupt enable 0 = Disables interrupt 1 = Enables interrupt If this bit is set to 0, then it disables the interrupt request to host controller. It does not mask the change in the MIXER_INT_STATUS [8] bit status	0
RSVD	[7:0]	–	Reserved	0

52.4.2.4 MIXER_INT_STATUS

- Base Address: 0x12C1_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:11]	–	Reserved (Read as zero Do not modify this bit.)	0
INT_CLEAR_VSYNC	[11]	RW	Specifies vertical synchronization interrupt clear 0 = Does not clear the interrupt 1 = Clears interrupt If you write 1 to this bit, then it clears the interrupt. You should Write 1 to this bit before you set INT_EN_VSYNC. This bit is Write only.	0
INT_STATUS_VP	[10]	RW	Specifies VP underflow interrupts status 0 = Does not fire an interrupt 1 = Fires an interrupt When you write 1 to this bit, then it clears the interrupt. The line buffer controller automatically asserts the interrupt when it generates an underflow in the line buffer.	0
INT_STATUS_GRP1	[9]	RW	Specifies graphic layer1 line buffer Underflow Interrupt Status 0 = Does not fire an interrupt 1 = Fires an interrupt If you write 1 to this bit, then it clears the interrupt. The line buffer controller automatically asserts the interrupt if it generates an underflow in the line buffer.	0
INT_STATUS_GRP0	[8]	RW	Specifies graphic layer0 line buffer Underflow Interrupt Status 0 = Does not fire an interrupt 1 = Fires an interrupt If you write 1 to this bit, then it clears the interrupt. The line buffer controller automatically asserts the interrupt if it generates an underflow in the line buffer.	0
RSVD	[7:1]	–	Reserved (Read it as zero. Do not modify this bit.)	0
INT_STATUS_VSYNC	[0]	RW	Specifies vertical synchronization status 0 = Does not fire an interrupt 1 = Fires an interrupt NOTE If (INT_STATUS_VSYNC) & (!INT_STATUS_VP) & (!INT_STATUS_GRP1) & (!INT_STATUS_GRP0) is HIGH, then it fires vertical synchronization signal. This bit is Read only.	0

52.4.2.5 MIXER_LAYER_CFG

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:12]	–	Reserved (Read as zero. Do not modify this bit.)	0
Graphic layer1 priority	[11:8]	RW	Specifies graphic layer1 priority value 0 = Hides the graphic layer1 15 to 1 = Sets the priority value	0
Graphic layer0 priority	[7:4]	RW	Specifies graphic layer0 priority 0 = Hides the graphic layer0 15 to 1 = Sets the priority value	0
Video layer-0 priority	[3:0]	RW	Specifies video layer priority 0 = Hides the video layer0 15 to 1 = Sets the priority value	0

You can set the priority value for video and each graphic layer. Use the priority field to determine the priority of a graphic layer. This field is also acts as an On/Off switch. If the field is set to zero, it does not display the corresponding graphic layer. If the layers have the same value, the priority is graphic layer 1 > graphic layer 0 > video. The difference in priority values determines the priority order. For example, case 1 and case 2 have the same effect.

- Case1: GRP1 priority is 2, GRP0 priority 3, Video Priority 1
- Case2: GRP1 priority is 14, GRP0 priority 15, Video Priority 13

NOTE: All changes to this register are valid on a vertical synchronization signal of next frame when SYNC_ENABLE flag is set to 1. "Hide" means the data layer is ready but it does not display.

52.4.2.6 MIXER_VIDEO_CFG

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:18]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_LIMITER_EN	[17]	RW	Specifies YUV limiter for xvYcc 0 = Disables 1 = Enables	0
REG_BLEND_EN	[16]	RW	If this bit is set to 1, it enables blending of the entire video layer onto the lower layer using the blending factor, REG_ALPHA_VID.	0
RSVD	[15:8]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_ALPHA_VID	[7:0]	RW	Specifies video layer blending factor Use this factor over all the pixels in the video layer to blend with the lower layer. It is given by: α × video_layer_pixel_value + (1 – α) (NOTE) × lower_layer_pixel_value If REG_ALPHA_VID is set to 0, then the blending factor (α) is 0. If REG_ALPHA_VID is not 0, then the blending factor (α) is given by α = (REG_ALPHA_VID + 1)/256.	0

NOTE: All changes to this register are valid on a vertical synchronization signal of next frame.

52.4.2.7 MIXER_VIDEO_LIMITER_PARA_CFG

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0018, Reset Value = 0xEB10_F010

Name	Bit	Type	Description	Reset Value
REG_PARA_Y_UPPER	[31:24]	RW	Upper bound for Y parameter of the limiter	0xEB
REG_PARA_Y_LOWER	[23:16]	RW	Lower bound for Y parameter of the limiter	0x10
REG_PARA_C_UPPER	[15:8]	RW	Upper bound for C parameter of the limiter	0xF0
REG_PARA_C_LOWER	[7:0]	RW	Lower bound for C parameter of the limiter	0x10

NOTE: All changes to this register are valid on a vertical synchronization signal of next frame.

52.4.2.8 MIXER_GRAPHIC0_CFG

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RTQoS_GRP0	[31:23]	RW	Specifies Real Time QoS configuration 0 = Disables RTQoS 1 to 480 = QoS threshold level 481 to 511 = Reserved The size of the graphic layer0 FIFO has a depth value of 480. Each level comprises of 128 bits.	0
RSVD	[22]	–	Reserved	0
BLANK_CHANGE0	[21]	RW	Specifies the usage of blank key(Color Key) 0 = Enables Blank key (color key) 1 = Disables Black Key (color key)	0
PRE_MUL_MODE0	[20]	RW	Specifies Pre-Multiplied_blending mode in graphic layer0 0 = Normal mode 1 = Pre-Multiplied mode In this mode, graphic pixel data should be pre-multiplied with graphic pixel alpha. In pre-multiplied mode, it should enable REG_PIXEL0_BLEND_EN. Graphic and Lower Layer Blending Factor It uses this factor all over the pixels in the graphic and lower layer to blend with. The blending factor (α) depends on the layer blending factor and on the pixel blending factor values. Refer to *Table 52-1* for more information.	0
RSVD	[19:18]	–	Reserved (Read as zero)	0
REG_WIN0_BLEND_EN	[17]	RW	The blending factor is set using the REG_ALPHA_WIN0 register for blending all over the graphic layer0.	0
REG_PIXEL0_BLEND_EN	[16]	RW	Enables blending factor for each pixel for blending in graphic layer 0.	0
RSVD	[15:12]	–	Reserved (Read as zero)	0
EG_COLOR_FORMAT0	[11:8]	RW	Specifies Graphic layer0 color format 0 = Reserved 1 = Reserved 2 = Reserved 3 = Reserved 4 = RGB 565 5 = ARGB 1555 6 = ARGB 4444 7 = ARGB 8888 8 = Reserved	0

Name	Bit	Type	Description	Reset Value
			9 = Reserved A = Reserved B = Reserved C = Reserved D = Reserved E = Reserved F = Reserved	
REG_ALPHA_WIN0	[7:0]	RW	Specifies Graphic layer0 and lower layer blending factor Use this factor all over the pixels in the graphic and lower layer to blend with. The blending factor (α) depends on the layer blending factor and pixel blending factor values that is given by α × graphic_layer_pixel_value + (1 – α) (NOTE) × lower_layer_pixel_value. Refer *Table 52-1* for more information. • If REG_ALPHA_WIN0 is 0, then the blending_factor_layer is 0. • If REG_ALPHA_WIN0 is not 0, then the blending_factor_layer = (REG_ALPHA_WIN + 1)/256. • If A (blending factor of each pixel) is 0, then the blending_factor_each_pixel is 0 • If A (blending factor of each pixel) is not 0, then the blending_factor_each_pixel = (A + 1)/256. It derives the pixel blending factors from the pixel in direct mode, except for the direct 16 bpp 565 format.	0

NOTE: All the changes to this register are valid on a vertical synchronization signal of next frame.

52.4.2.9 MIXER_GRAPHIC1_CFG

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RTQoS_GRP1	[31:23]	RW	Specifies Real Time QoS configuration 0 = Disables RTQoS 1 to 480 = QoS threshold level 481 to 511 = Reserved The size of the graphic layer0 FIFO has a depth value of 480. Each level comprises of 128 bits.	0
RSVD	[22]	–	Reserved	0
BLANK_CHANGE1	[21]	RW	Specifies the usage of blank key (Color Key) 0 = Enables Blank key (color key). 1 = Disables Black Key (color key).	0
PRE_MUL_MODE1	[20]	RW	Pre-Multiplied_blending Mode in Graphic Layer1. 0 = Normal mode 1 = Pre-Multiplied mode In this mode, graphic pixel data should be pre-multiplied with graphic pixel alpha. In pre-multiplied mode, it should enable REG_PIXEL1_BLEND_EN. Graphic and lower layer blending factor Use this factor all over the pixels in the graphic and lower layer to blend with. A blending factor (α) depends on the layer blending factor and a pixel blending factor values: Refer to *Table 52-1* for more information.	0
RSVD	[19:18]	–	Reserved (Read it as zero)	0
REG_WIN1_BLEND_EN	[17]	RW	The blending factor is set using the REG_ALPHA_WIN1 register for blending all over the graphic layer1.	0
REG_PIXEL1_BLEND_EN	[16]	RW	Enables blending factor for each pixel for blending in graphic layer1.	0
RSVD	[15:12]	–	Reserved (Read it as zero)	0
REG_COLOR_FORMAT1	[11:8]	RW	Specifies graphic layer1 color format 0 = Reserved 1 = Reserved 2 = Reserved 3 = Reserved 4 = RGB 565 5 = ARGB 1555 6 = ARGB 4444 7 = ARGB 8888 8 = Reserved 9 = Reserved A = Reserved	0

Name	Bit	Type	Description	Reset Value
			B = Reserved C = Reserved D = Reserved E = Reserved F = Reserved	
REG_ALPHA_WIN1	[7:0]	RW	Graphic Layer1 and Lower Layer Blending Factor Use this factor all over the pixels in the graphic and lower layer to blend with. The blending factor (α) depends on the layer blending factor and a pixel blending factor values that is given by α × graphic_layer_pixel_value + (1 – α) (NOTE) × lower_layer_pixel_value. Refer to *Table 52-1* for more information. • If REG_ALPHA_WIN1 is 0, then the blending_factor_layer is 0. • If REG_ALPHA_WIN1 is not 0, then the blending_factor_layer = (REG_ALPHA_WIN + 1)/256. • If A (blending factor of each pixel) is 0, then the blending_factor_each_pixel is 0 • If A (blending factor of each pixel) is not 0, then the blending_factor_each_pixel = (A + 1)/256. The pixel blending factors derives from the pixel in direct mode, except for the direct 16 bpp 565 format.	0

NOTE: All the changes to this register are valid on a vertical synchronization signal of the next frame.

Table 52-1 describes the graphic blending factor alpha in case of a normal mode.

Table 52-1 Graphic Blending-factor Alpha in Case of Normal Mode

MIXER_GRAPHICx_CFG. REG_WINx_BLEND_EN	MIXER_GRAPHICx_CFGx. REG_PIXELx_BLEND_EN	Alpha Value (Blending Factor of Each Pixel)
0	0	1
0	1	blending_factor_each_pixel (A)
1	0	blending_factor_layer (MIXER_GRAPHICn_CFG[7:0])
1	1	blending_factor_layer × blending_factor_each_pixel

Table 52-2 describes the graphic blending method.

Table 52-2 Graphic Blending Method

Pixel Blend	Window Blend	Normal Mode	Pre Multiplied Mode
0	0	–	–
0	1	Alpha_gw × graphic_pixel + (1 – alpha_gw) × lower layer	Alpha_gw × graphic_pixel + (1 – alpha_gw) × lower layer
1	0	Alpha_gp × graphic_pixel + (1 – alpha_gp) × lower layer	graphic_pixel + (1 – alpha_gp) × lower layer
1	1	(Alpha_gp × alpha_gw) × graphic_pixel + (1 – alpha_gp × alpha_gw) × lower layer	alpha_gw × graphic_pixel + (1 – alpha_gp * alpha_gw) × lower layer

In pre-multiplied mode, it multiples the input graphic data with the pixel blending factor (alpha_gp) and truncates to the size of source format bits. For example, the result of multiplying 8-bit data with 8-bit pixel blending factor is 16 bits which is the first term in the blending equation in Table 52-2.

In normal mode, it truncates the supplied data to 8 bits resulting in the loss of the LSB during calculation. In direct 32 bpp mode, it cannot distinguish the loss in data visually because of a small difference of ±1. In direct 16 bpp mode, you can visually see data loss when you compare it to the original data.

For example, in 4633 direct mode, it truncates the pre-multiplied Cb data to 3 bits that results in a difference value of ±15. This range of difference leads to a visual difference. You cannot reduce the error that results from 3-bit input source.

52.4.2.10 MIXER_GRAPHICn_BASE (n = 0 to1)

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0024, + 0x0044, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
REG_GRAPHICn_BASE	[31:0]	RW	Specifies base address of frame buffer for graphic layer. You should word align this address. The lease two significant bits ([1:0]) automatically sets to 2'b00	0

52.4.2.11 MIXER_GRAPHICn_SPAN (n = 0 to1)

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0028, + 0x0048, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:15]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_GRAPHICn_SPAN	[14:0]	RW	Specifies horizontal pixel interval between any consequent two lines in the source image of graphic layer NOTE: SPAN is the horizontal pixel count of original image. For example, in 640 × 480, span is 640. It does not relate to bit per pixel (bpp).	0

52.4.2.12 MIXER_GRAPHICn_WH (n = 0 to 1)

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0030, + 0x0050, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_H_SCALEn	[28]	RW	Specifies horizontal scaling configuration 0 = Does not available 1 = x2 Scaling-Up	0
RSVD	[27]	RW	Reserved (Read it as zero. Do not modify this bit.)	0
REG_GRAPHICn_W	[26:16]	RW	Specifies width of graphic layer in pixel unit.	0
RSVD	[15:13]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_V_SCALEn	[12]	RW	Specifies vertical duplication configuration. 0 = Does not available 1 = x2 Duplication	0
RSVD	[11]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_GRAPHICn_H	[10:0]	RW	Specifies height of graphic layer in pixel unit.	0

NOTE: All the changes to this register are valid on a vertical synchronization signal of the next frame.
When specifying the X coordinates and the width of a graphic layer, it should be located inside the display region. For example, 720 × 480 region in NTSC display mode and 720 × 576 region in PAL display mode. The coordinates (x and y) and the size (width and height) should depend on the progressive mode although it is interlaced display mode.
The graphic width and height should be larger than 0, when the corresponding bit fields of REG_GRAPHICx_EN register, namely, MIXER_CFG[5] and MIXER_CFG[4] are set to 1.

Figure 52-3 illustrates the mixer horizontal scale and blank-key.

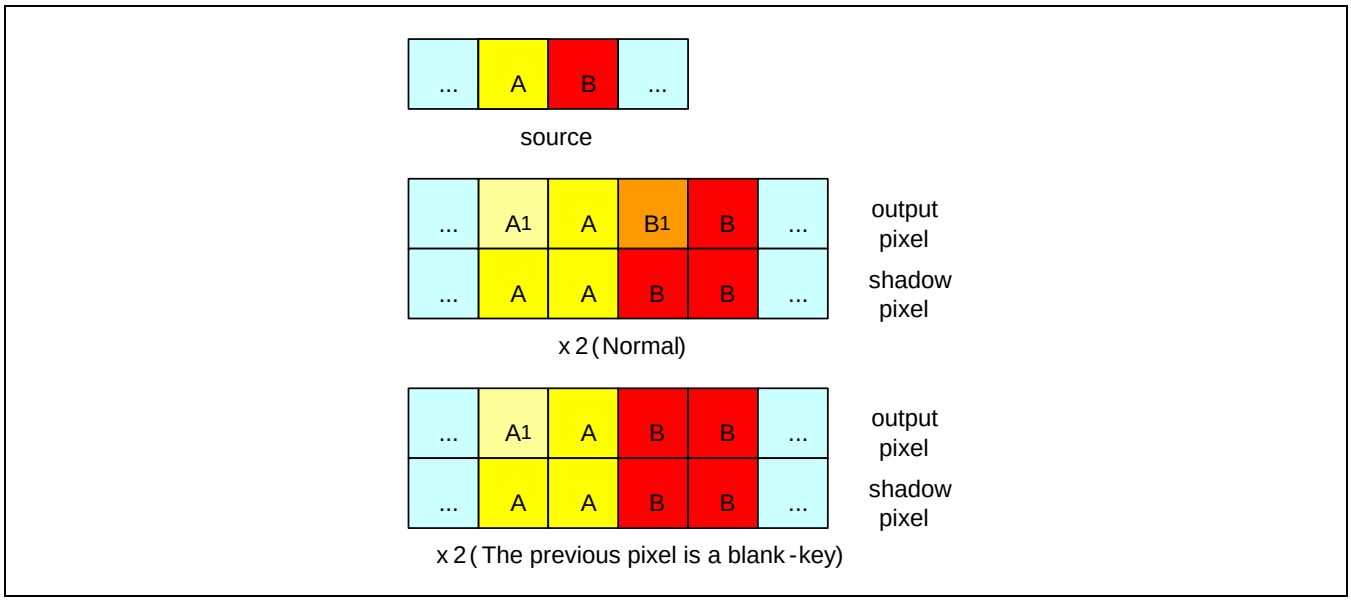


Figure 52-3 Mixer Horizontal Scale and Blank-key

52.4.2.13 MIXER_GRAPHICn_XY (n = 0 to 1)

- Base Address: 0x12C1_0000
- Address = Base Address + 0x002C, + 0x004C , Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_GRAPHICn_SX	[26:16]	RW	Specifies X coordinate of upper left corner of graphic layer in source frame in pixel unit. The allowable ranges are: • 0 to 719 at SD mode • 0 to 1279 at HD mode (720p) • 0 to 1919 at HD mode (1080i/p)	0
RSVD	[15:11]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_GRAPHICn_SY	[10:0]	RW	Specifies Y coordinate of upper left corner of graphic layer in source frame in pixel unit. The allowable ranges are: • 0 to 575 at SD mode • 0 to 719 at HD mode (720p) • 0 to 1079 at HD mode (1080i/p)	0

52.4.2.14 MIXER_GRAPHICn_DXY (n = 0 to 1)

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0034, + 0x0054, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:27]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_GRAPHICn_DX	[26:16]	RW	Specifies X coordinate of upper left corner of graphic layer in destination frame in pixel unit. The allowable ranges are: • 0 to 719 at SD mode • 0 to 1279 at HD mode (720p) • 0 to 1919 at HD mode (1080i/p)	0
RSVD	[15:11]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_GRAPHICn_DY	[10:0]	RW	Specifies Y coordinate of upper left corner of graphic layer in destination frame in pixel unit. The allowable ranges are: • 0 to 575 at SD mode • 0 to 719 at HD mode (720p) • 0 to 1079 at HD mode (1080i/p)	0

52.4.2.15 MIXER_GRAPHICn_BLANK (n = 0 to 1)

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0038, + 0x0058, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
REG_GRAPHICn_BLANK	[31:0]	–	Specifies blank pixel value for Graphic LayerN. NOTE: When blank pixel is ARGB, the register value should be equal to the pixel value and the alpha value.	0

52.4.2.16 MIXER_BG_COLORn (n = 0 to 2)

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0064, + 0x0068, + 0x006C , Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved (Read it as zero. Do not modify this bit.)	0
Y	[23:16]	RW	Y component of background color.	0
Cb	[15:8]	RW	Cb component of background color.	0
Cr	[7:0]	RW	Cr component of background color.	0

NOTE: You can choose an appropriate YCbCr value for BT.601 or BT.709.

52.4.2.17 MIXER_CM_COEFF_Y

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0080, Reset Value = 0x0844_0832

Name	Bit	Type	Description	Reset Value
RSVD	[31]	–	Reserved (Read it as zero. Do not modify this bit.)	–
WIDE_SEL	[30]	RW	Specifies the RGB range mode. 0 = Narrow 1 = Wide	0
REG_COEFF_00	[29:20]	RW	Specifies scaled Color Space Conversion Coefficient (C_{00}) [29] = Sign-bit [28:20] = Fractional bit The default and recommended value of C_{00} is 0.257 in decimal	0x84
REG_COEFF_10	[19:10]	RW	Specifies scaled Color Space Conversion coefficient (C_{10}) [19] = Sign-bit [18:10] = Fractional bit The default and recommended value of C_{10} is 0.504 in decimal	0x102
REG_COEFF_20	[9:0]	RW	Specifies Scaled Color Space Conversion Coefficient	0x32

Name	Bit	Type	Description	Reset Value
			(C_{20}) [9] = Sign-bit [8:0] = Fractional bit The default and recommended value of C_{20} is 0.098 in decimal	

NOTE: RGB to YCbCr Conversion Equations

The conversion equations for narrow RGB to YCbCr are:

Y601 = 0.299R + 0.587G + 0.114B Y709 = 0.213R + 0.715G + 0.072B
Cb = – 0.172R – 0.339G + 0.511B + 128 Cb = – 0.117R – 0.394G + 0.511B + 128
Cr = 0.511R – 0.428G – 0.083B + 128 Cr = 0.511R – 0.464G – 0.047B + 128

The conversion equations for wide RGB to YCbCr are:

Y601 = 0.257R + 0.504G + 0.098B + 16 Y709 = 0.183R + 0.614G + 0.062B + 16
Cb = – 0.148R – 0.291G + 0.439B + 128 Cb = – 0.101R – 0.338G + 0.439B + 128
Cr = 0.439R – 0.368G – 0.071B + 128 Cr = 0.439 – 0.399G – 0.040B + 128

Fraction Number (Example of 1-bit is Signed bit and 9 bits are Fraction Bit)

0.098 = 0.5 × "0" + 0.25 × "0" + 0.125 × "0" + 0.0625 × "1" + 0.03125 × "1" + 0.015625 × "0" + 0.0078125 × "0" + 0.00390625 × "1" + 0.001953125 × "0" = 0 (signed bit) 000110010 = 0x032
– 0.148 → Consider only the unsigned part 0.148
0.148 = 0.5 × "0" + 0.25 × "0" + 0.125 × "1" + 0.0625 × "0" + 0.03125 × "0" + 0.015625 × "1" + 0.0078125 × "0" + 0.00390625 × "1" + 0.001953125 × "1" = 0 (signed bit) 001001011 = 10'b0001001011

For the correct representation of – 0.148, you should calculate the 2's compliment of 0.148

10'b0001001011 → 10'b1110110100 (bitwise invert) + 1 = 10'b1110110101 = 0x3B5

52.4.2.18 MIXER_CM_COEFF_CB

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0084, Reset Value = 0x3B5D_B0E1

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_COEFF_01	[29:20]	RW	Specifies scaled color space conversion Coefficient (C_{01}) [29] = Sign-bit [28:20] = Fractional bit Default value of C_{01}: – 0.0742785 in decimal Recommended value of C_{01}: – 0.148 in decimal and 0x3b5 in hexadecimal	0x3b4
REG_COEFF_11	[19:10]	RW	Specifies scaled color space conversion Coefficient (C_{11}) [19] = Sign-bit [18:10] = Fractional bit Default value of C_{11}: – 0.1455078125 in decimal Recommended value of C_{11}: – 0.291 in decimal and 0x36c in hexadecimal	0x36b
REG_COEFF_21	[9:0]	RW	Specifies scaled color space conversion Coefficient (C_{21}) [9] = Sign-bit [8:0] = Fractional bit Default and Recommended value of C_{21}: 0.439 in decimal	0xe1

52.4.2.19 MIXER_CM_COEFF_CR

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0088, Reset Value = 0x0E1D_13DC

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_COEFF_02	[29:20]	RW	Specifies scaled color space conversion coefficient (C_{02}). [29] = Sign-bit [28:20] =Fractional bit Default and recommended value of C_{02}: 0.439 in decimal	0xe1
REG_COEFF_12	[19:10]	RW	Specifies scaled color space conversion coefficient (C_{12}). [19] =Sign-bit [18:10] =Fractional bit Default and recommended value of C_{12}: – 0.368 in decimal	0x344
REG_COEFF_22	[9:0]	RW	Specifies scaled color space conversion Coefficient (C_{22}). [9] = Sign-bit [8:0] = Fractional bit Default and recommended value of C_{22}: – 0.071 in decimal	0x3dc

52.4.2.20 MIXER_TVOUT_CFG

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0100, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:2]	–	Reserved (Read it as zero. Do not modify this bit.)	0
REG_INTERLACED	[1]	RW	Specifies frame packet interlaced mode 0 = Progressive mode 1 = Interlaced mode This bit is valid only when the value of REG_STEREO_SCOPIC is 1.	0
REG_STEREO_SCOPIC	[0]	RW	Specifies stereo scopic 0 = 2D Format 1 = Stereo Scopic (Frame Packing)	0

NOTE: Mixer supports only frame packing in stereoscopic mode.

52.4.2.21 MIXER_3D_ACTIVE_VIDEO

- Base Address: 0x12C1_0000
- Address = Base Address + 0x0104, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
REG_HACTIVE_VIDEO	[31:16]	RW	Specifies HACTIVE video pixel count	0x0
REG_VACTIVE_VIDEO	[15:0]	RW	Specifies VACTIVE video line count	0x0

Figure 52-4 illustrates the 3D frame packing.

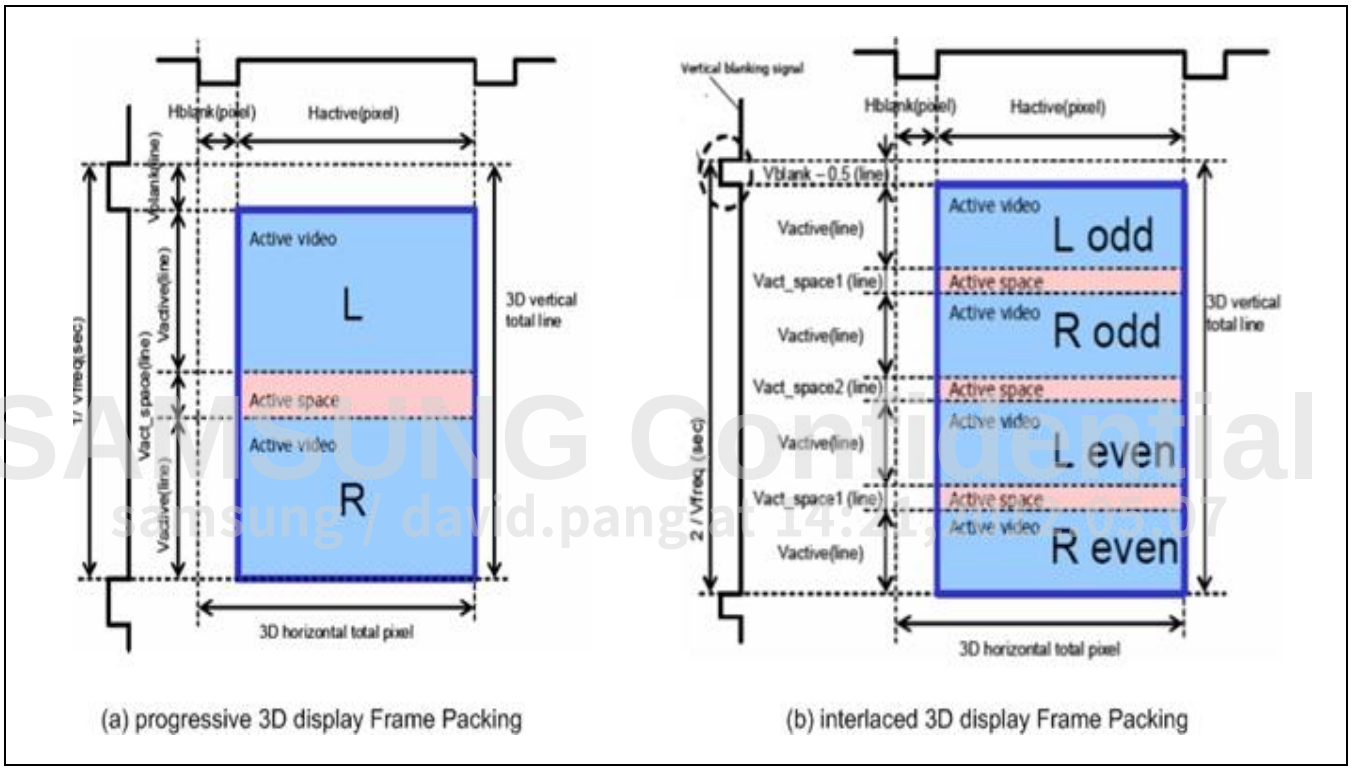


Figure 52-4 3D Frame Packing

52.5 Layers

Mixer blends all the image sources such as video layer, graphic layer, and background layer and transfers the blended pixel data to TVNEC/HDMI. You should set the priority value of layers to select the order of blending operation. You can completely order and blend the video and the two graphic layers. The background layer is always the lowest layer. The priority order for layers is given by:

- Background → (Video ↔ Graphic0 ↔ Graphic1)

Register setting enables or disables the video layer and graphic layer. The blending factors for the different layers are:

- Background layer- This is the lowest layer and has no blending factor.
- Video layer- This layer has a blending factor of 1 that applies to all pixels in the video layer. The REG_ALPHA_VID bit of MIXER_VIDEO_CFG [7:0] register determines the blending factor. The bit setting on the register enables or disables the video blending.
- Graphic0 layer- This layer supports pixel blending and window blending. It applies pixel blending factors pixel-by-pixel, although the window blending factor is applied on all pixels in the graphic layer. It should apply these two blending factors simultaneously to a pixel.
- Graphic1 layer- This layer supports pixel blending and window blending. It applies pixel blending factors pixel-by-pixel, although the window blending factor is applied on all pixels in the graphic layer. It should apply these two blending factors simultaneously to a pixel.

Figure 52-5 illustrates the mixer blending.

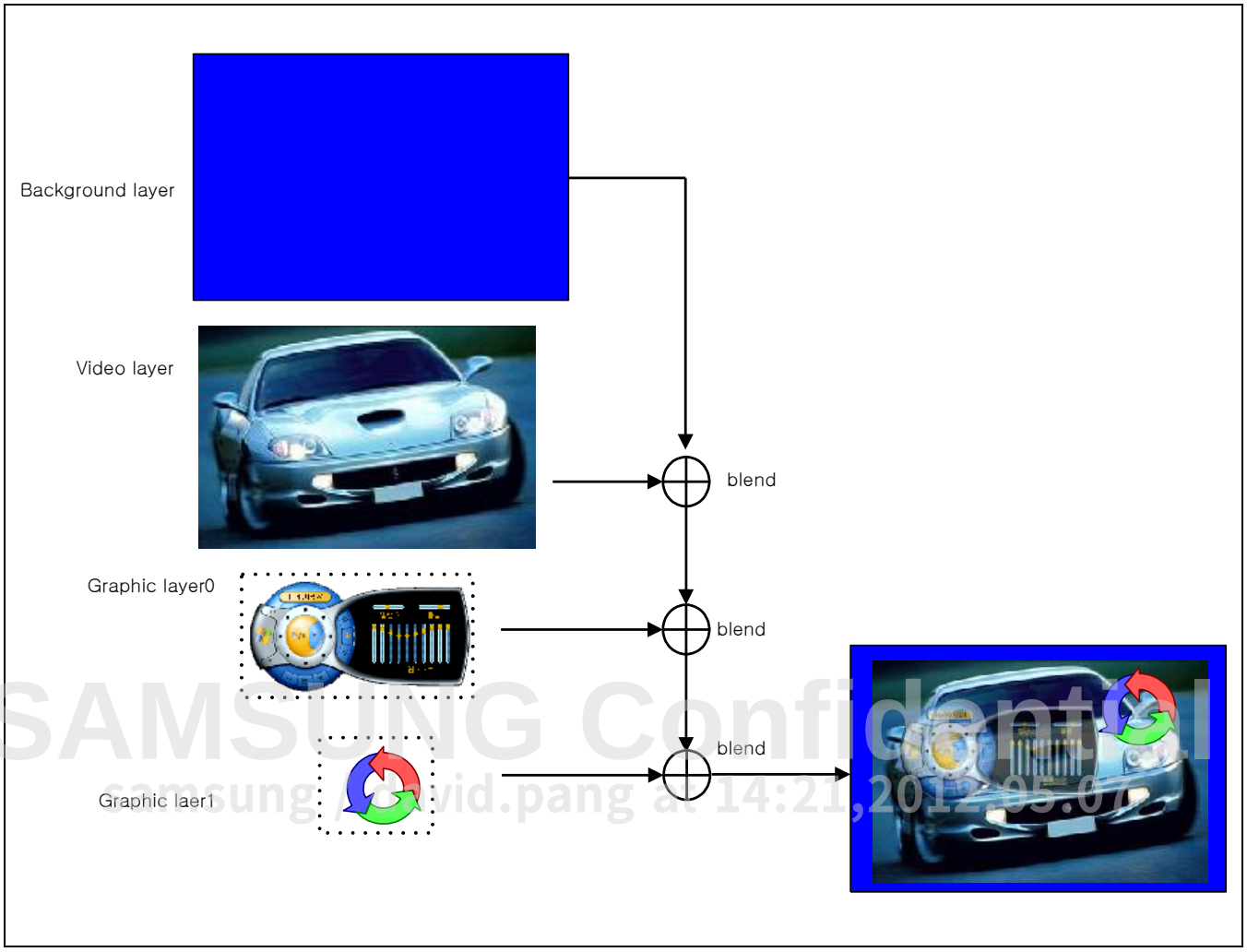


Figure 52-5 Mixer Blending

52.5.1 Video Layer

Video data is directly transferred from the video processor to mixer in YCbCr[888] in 4:4:4 formats. The display region of video data is smaller than screen size when video processor scales the source image in letterbox mode. In this case, you can see the background layer in the blank region.

52.5.2 Graphic Layer

ARM or Graphic Accelerator generates the graphic source data in the external memory and it transfers graphic source data to the mixer using AXI access. Mixer supports the graphic formats.

They are:

- 16 bpp RGB[565]
- 16 bpp ARGB[1555]
- 16 bpp ARGB[4444]
- 32 bpp ARGB[8888]

In 16 and 32 bpp direct modes, the value of a pixel data directly indicates the RGB, but the bit width for R, G, and B are different for each mode. For example, for a 16 bpp direct ARGB[4444] mode, it assigns the RGB component to 16-bit length.

Figure 52-6 illustrates an example for 16 bpp ARGB.

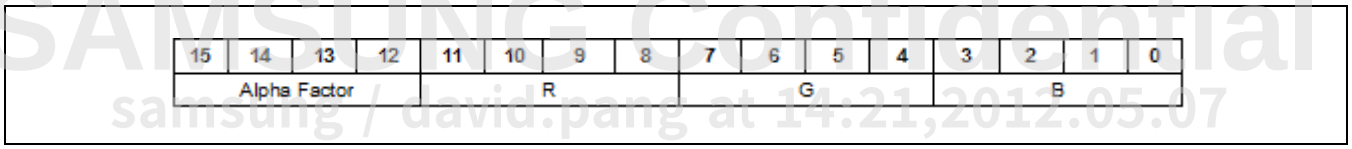


Figure 52-6 16 bpp ARGB Example

It processes the internal data path with YCbCr[888] format. Therefore, it converts the RGB format to YCbCr during color matrix conversion.

If bit per pixel (bpp) of color format is less than 8 bits, then you should use the value only after expanding. *Figure 52-7* illustrates the ARGB [1555] example of expanding.

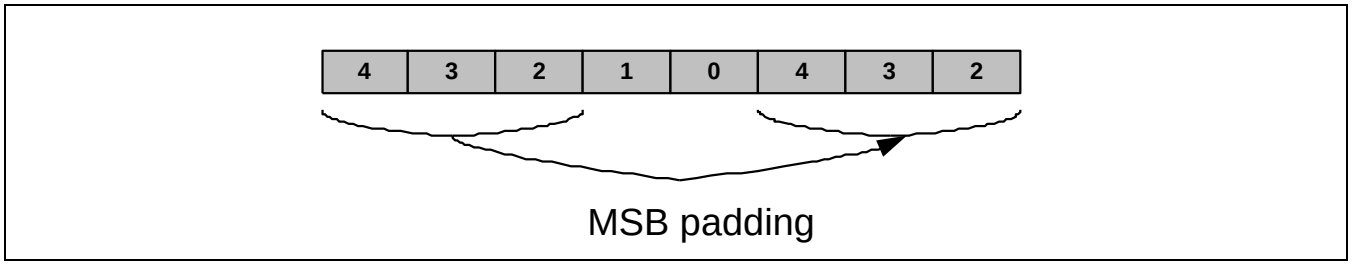


Figure 52-7 Example of Expanding

Mixer supports up to two graphic layers and one video layer. You can enable or disable each layer and can configure the priority among the layers. There is a blending factor between different layers. The graphic layers have different color formats associated with them.

When specifying (X, Y) co-ordinates, and width/height of a graphic layer, the graphic layers should be located in their appropriate display regions. The supported display modes are:

- 720 × 480/ 720 × 576 in SD display mode
- 1280 × 720p/1920 ×1080i/ p in HD display mode

Mixer does not support the clipping operation for the pixels that are displayed out of screen.

52.5.3 Blank Pixel

Blank pixel data in graphic layer is a pixel data that is transparent to the lower layer. You can define a blank pixel data in the register (MIXER_GRAPHICn_BLANK) and if the graphic data is same as the blank pixel value, a lower image is seen instead of the blank pixel.

52.5.4 Source Data in Memory

As the graphic data comes from the external memory through the bus, the memory format for the source data is dependent on the bus system endian. In little endian system, it stores the lower 8 bits in the lower address.

It is possible to align the source data format of the mixer in little endian or big endian format. These data formats are applicable to the data of graphic layer.

Mixer supports many graphic formats. *Figure 52-8* illustrates the graphic data format in memory. It shows the pixels in a display visible through human eyes. It also shows the supported data source formats.

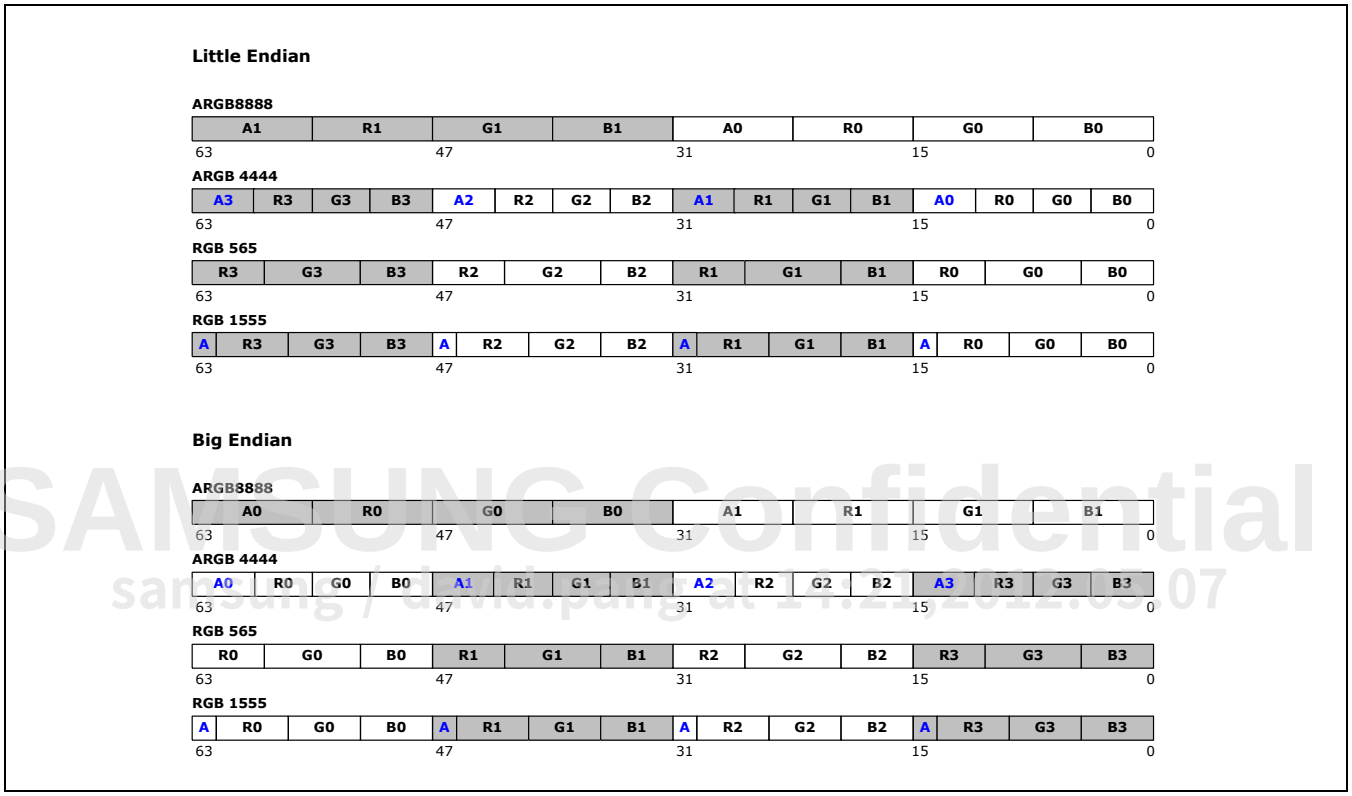


Figure 52-8 Graphic Data Format in Memory

Pixels with lower X coordinates are located from the left. These pixel data are represented by different digital data depending on the graphic format setting. For example, one graphic pixel is represented by 16-bit digital data in 16 BPP mode. If these pixels are processed in the Mixer, the word format to represent these pixels is different depending on the endian format.

In the big endian mode, the pixel with lower X coordinate is positioned to the MSB parts in the 64-bit register of Mixer. However, in the little endian mode, the pixel with lower X coordinate is positioned to the LSB parts in the 64-bit register.

52.5.5 Background Layer

If there is no video or graphic, then the background color is seen in the display region. You should use YCbCr[888] format to set the background color in the register.

53 High-Definition Multimedia Interface

53.1 Overview

The High-Definition Multimedia Interface (HDMI) 1.4 (3D feature), 1.4 Tx Subsystem Version 1.0 consists of HDMI Tx Core with I2S/SPDIF input interface, Consumer Electronics Control (CEC) block, and High-bandwidth Digital Content Protection (HDCP) Key Block.

53.2 Features

The features of HDMI are:

- Complies with HDMI 1.4 (3D feature), HDCP 1.1, and DVI 1.0
- The video formats that HDMI supports are:
  - 480p 59.94 Hz/60 Hz, 576p @ 50 Hz
  - 720p @ 50 Hz/59.94 Hz/60 Hz
  - 1080i @ 50 Hz/59.94 Hz/60 Hz
  - 1080p @ 50 Hz/59.94 Hz/60 Hz
- Supports other various formats up to 148.5 MHz Pixel Clock
- Supports Color Format: 4:4:4 RGB/YCbCr
- Supports 8-bit precision per color only
- Supports CEC function
- Contains an Integrated HDCP Encryption Engine for video/ audio content protection
- Does not include DDC. There is a dedicated Inter-Integrated Circuit (I2C) for DDC in Exynos 4412 SCP peripheral Bus

53.3 Block Diagram of HDMI

Figure 53-1 illustrates the block diagram of HDMI.

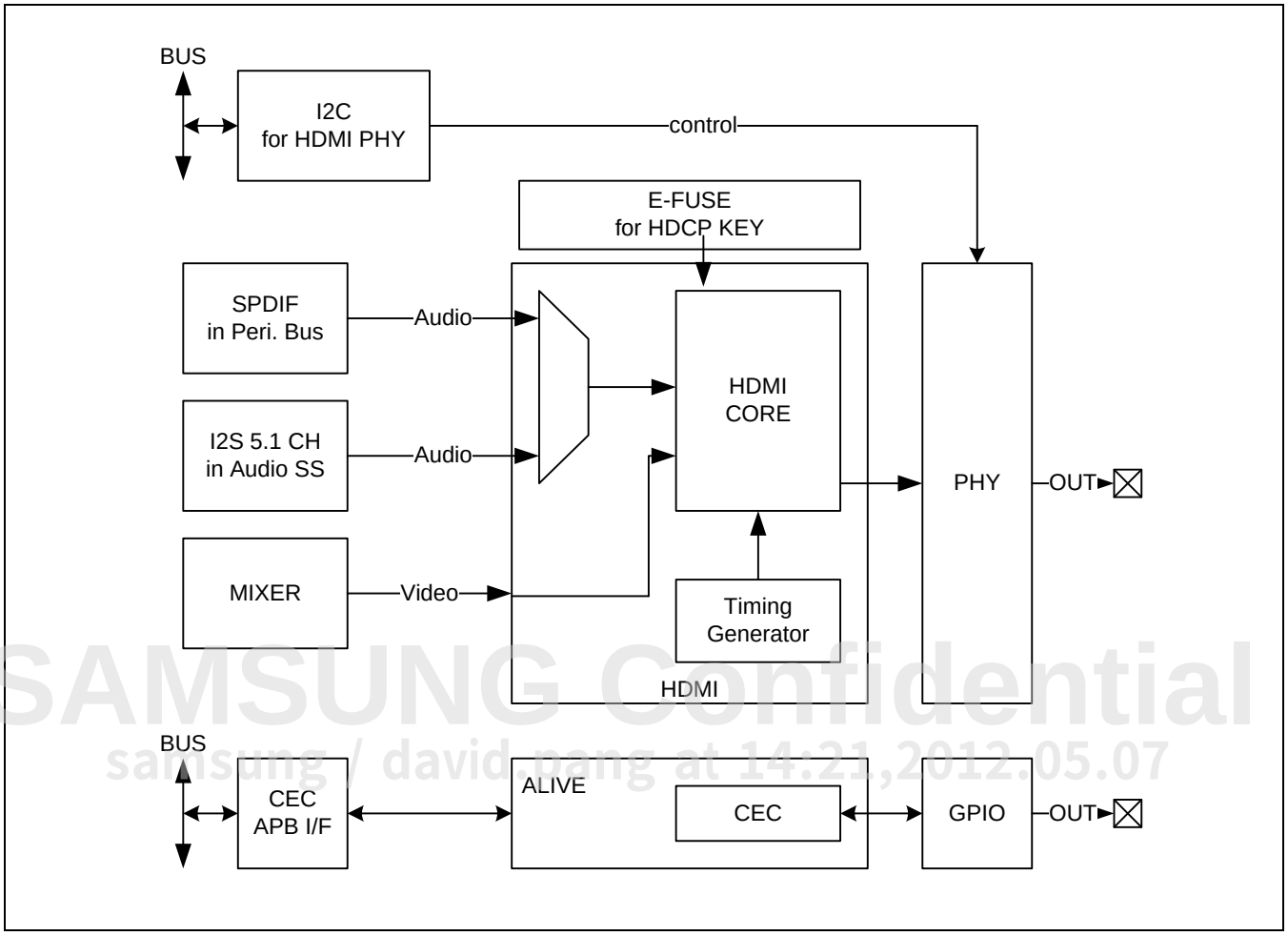


Figure 53-1 Block Diagram of HDMI

HDMI Tx V1.4 consists of several blocks. Each block has a unique function. For example, the Mixer specifies the source image of HDMI. It transmits image data, which can be either in RGB888 or YUV444. Before it works, you must set the pixel clock properly. The ratio of pixel clock depends on the output resolution. You can configure HDMI_PHY.

SPDIF in peripheral bus and I2S 5.1 channel in Audio sub-system feed audio data in HDMI Tx V1.4. Refer to SPDIF and I2S datasheets for more information.

HDMI Tx V1.4 in Exynos 4412 SCP supports embedded HDCP key system. Exynos 4412 SCP does not allow access to HDCP key.

A dedicated I2C is used to configure HDMI PHY. In addition, HDMI PHY generates pixel and Transition Minimized Differential Signaling (TMDS) clock through I2C.

The CEC block is separate from HDMI Tx, and is used by wake-up source in Exynos 4412 SCP. It belongs to the ALIVE block and communicates with external CEC through bi-directional General Purpose Input Output (GPIO).

53.4 Block Diagram of HDMI SUB-system in Exynos 4412 SCP

Figure 53-2 illustrates the block diagram of HDMI sub-system in Exynos 4412 SCP.

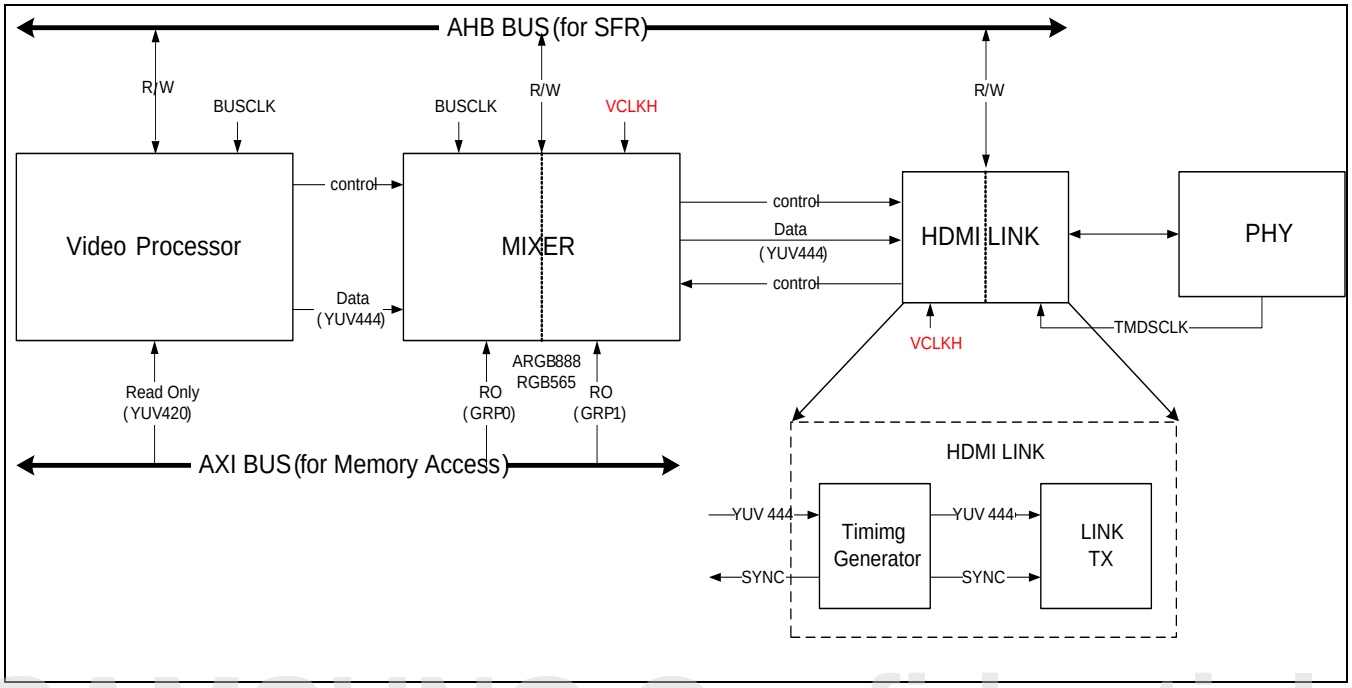


Figure 53-2 Block Diagram of HDMI SUB-system in Exynos 4412 SCP

53.5 Block Diagram of HDCP Key Management

Figure 53-3 illustrates the HDCP key management.

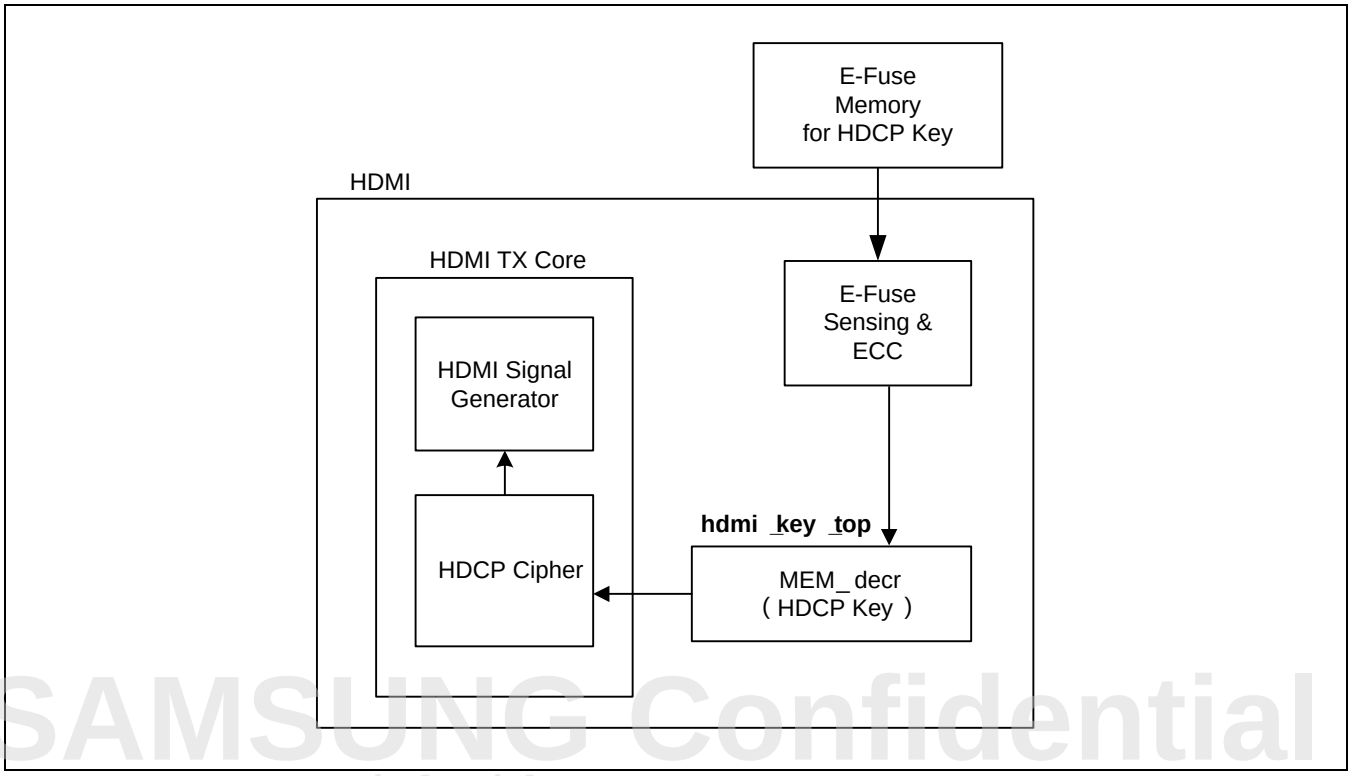


Figure 53-3 Block Diagram of HDCP Key Management

Exynos 4412 SCP supports embedded HDCP key system. The HDCP key value is fused during fabrication, based on customers' request. Exynos 4412 SCP strictly prohibits access to HDCP key value from any method. After Exynos 4412 SCP boot up, it loads the HDCP key by using SFR from E-FUSE memory (HDCP_E_FUSE_CTRL, 0x12D6_000, [0] bit).

Contact the Customer Service (CS) team when you want to use the HDCP system.

53.6 Video Input Timing GUIDE for HDMI Timing Generator

Table 54-1, *Table 54-2* describes HDMI link timing generator configuration guide.

Table 53-1 HDMI LINK 2D Timing Generator Configuration Guide

–	720×480p	720×576p	1280×720p	1920x1080i	1920×1080p
TG_H_FSZ_L (0x1458_0018)	0x5a	0x60	0x72	0x98	0x98
TG_H_FSZ_H (0x1458_001C)	0x03	0x03	0x06	0x08	0x08
TG_HACT_ST_L (0x1458_0020)	0x8a	0x90	0x72	0x18	0x18
TG_HACT_ST_H (0x1458_0024)	0x00	0x00	0x01	0x01	0x01
TG_HACT_SZ_L (0x1458_0028)	0xd0	0xd0	0x00	0x80	0x80
TG_HACT_SZ_H (0x1458_002C)	0x02	0x02	0x05	0x07	0x07
TG_V_FSZ_L (0x1458_0030)	0x0d	0x71	0xee	0x65	0x65
TG_V_FSZ_H (0x1458_0034)	0x02	0x02	0x02	0x04	0x04
TG_VSYNC_L (0x1458_0038)	0x01	0x01	0x01	0x01	0x01
TG_VSYNC_H (0x1458_003C)	0x00	0x00	0x00	0x00	0x00
TG_VSYNC2_L (0x1458_0040)	Reset value	Reset value	Reset value	0x33	Reset value
TG_VSYNC2_H (0x1458_0044)	Reset value	Reset value	Reset value	0x02	Reset value
TG_VACT_ST_L (0x1458_0048)	0x2d	0x31	0x1e	0x16	0x2d
TG_VACT_ST_H (0x1458_004C)	0x00	0x00	0x00	0x00	0x00
TG_VACT_SZ_L (0x1458_0050)	0xe0	0xe0	0xd0	0x1c	0x38
TG_VACT_SZ_H (0x1458_0054)	0x01	0x01	0x02	0x02	0x04
TG_FIELD_CHG_L (0x1458_0058)	Reset value	Reset value	Reset value	0x33	Reset value
TG_FIELD_CHG_H (0x1458_005C)	Reset value	Reset value	Reset value	0x02	Reset value
TG_VACT_ST2_L (0x1458_0060)	Reset value	Reset value	Reset value	0x49	Reset value

–	720×480p	720×576p	1280×720p	1920x1080i	1920×1080p
TG_VACT_ST2_H (0x1458_0064)	Reset value	Reset value	Reset value	0x02	Reset value
TG_VACT_ST3_L (0x1458_0068)	Reset value	Reset value	Reset value	Reset value	Reset value
TG_VACT_ST3_H (0x1458_006C)	Reset value	Reset value	Reset value	Reset value	Reset value
TG_VACT_ST4_L (0x1458_0070)	Reset value	Reset value	Reset value	Reset value	Reset value
TG_VACT_ST4_H (0x1458_0074)	Reset value	Reset value	Reset value	Reset value	Reset value
TG_VSYNC_TOP_HDMI_L (0x1458_0078)	0x01	0x01	0x01	0x01	0x01
TG_VSYNC_TOP_HDMI_H (0x1458_007C)	0x00	0x00	0x00	0x00	0x00
TG_VSYNC_BOT_HDMI_L (0x1458_0080)	Reset value	Reset value	Reset value	0x33	Reset value
TG_VSYNC_BOT_HDMI_H (0x1458_0084)	Reset value	Reset value	Reset value	0x02	Reset value
TG_FIELD_TOP_HDMI_L (0x1458_0088)	0x01	0x01	0x01	0x01	0x01
TG_FIELD_TOP_HDMI_H (0x1458_008C)	0x00	0x00	0x00	0x00	0x00
TG_FIELD_BOT_HDMI_L (0x1458_0090)	Reset value	Reset value	Reset value	0x33	Reset value
TG_FIELD_BOT_HDMI_H (0x1458_0094)	Reset value	Reset value	Reset value	0x02	Reset value
TG_3D_FP (0x1458_0094)	Reset value	Reset value	Reset value	Reset value	Reset value

Table 53-2 HDMI LINK 3D Timing Generator Configuration Guide

–	1280×720p	1280×720p	1920x1080p	1920x1080p
–	59.94/60Hz	50Hz	59.94/60Hz	24Hz
–	TB/SB(H)	TB	TB/SB(H)	TB/SB(H)
TG_H_FSZ_L (0x1458_0018)	0x72	0xBC	0x98	0xBE
TG_H_FSZ_H (0x1458_001C)	0x06	0x07	0x08	0x0A
TG_HACT_ST_L (0x1458_0020)	0x72	0xBC	0x18	0x3E
TG_HACT_ST_H (0x1458_0024)	0x01	0x02	0x01	0x03
TG_HACT_SZ_L (0x1458_0028)	0x00	0x00	0x80	0x80
TG_HACT_SZ_H (0x1458_002C)	0x05	0x05	0x07	0x07
TG_V_FSZ_L (0x1458_0030)	0xee	0xEE	0x65	0x65
TG_V_FSZ_H (0x1458_0034)	0x02	0x02	0x04	0x04
TG_VSYNC_L (0x1458_0038)	0x01	0x01	0x01	0x01
TG_VSYNC_H (0x1458_003C)	0x00	0x00	0x00	0x00
TG_VSYNC2_L (0x1458_0040)	Reset value	Reset value	Reset value	Reset value
TG_VSYNC2_H (0x1458_0044)	Reset value	Reset value	Reset value	Reset value
TG_VACT_ST_L (0x1458_0048)	0x1E	0x1E	0x2d	0x2D
TG_VACT_ST_H (0x1458_004C)	0x00	0x00	0x00	0x00
TG_VACT_SZ_L (0x1458_0050)	0xd0	0xD0	0x38	0x38
TG_VACT_SZ_H (0x1458_0054)	0x02	0x02	0x04	0x04
TG_FIELD_CHG_L (0x1458_0058)	Reset value	Reset value	Reset value	Reset value
TG_FIELD_CHG_H (0x1458_005C)	Reset value	Reset value	Reset value	Reset value
TG_VACT_ST2_L (0x1458_0060)	Reset value	Reset value	Reset value	Reset value

–	1280×720p	1280×720p	1920x1080p	1920x1080p
–	59.94/60Hz	50Hz	59.94/60Hz	24Hz
–	TB/SB(H)	TB	TB/SB(H)	TB/SB(H)
TG_VACT_ST2_H (0x1458_0064)	Reset value	Reset value	Reset value	Reset value
TG_VACT_ST3_L (0x1458_0068)	Reset value	Reset value	Reset value	Reset value
TG_VACT_ST3_H (0x1458_006C)	Reset value	Reset value	Reset value	Reset value
TG_VACT_ST4_L (0x1458_0070)	Reset value	Reset value	Reset value	Reset value
TG_VACT_ST4_H (0x1458_0074)	Reset value	Reset value	Reset value	Reset value
TG_VSYNC_TOP_HDMI_L (0x1458_0078)	Reset value	Reset value	Reset value	Reset value
TG_VSYNC_TOP_HDMI_H (0x1458_007C)	Reset value	Reset value	Reset value	Reset value
TG_VSYNC_BOT_HDMI_L (0x1458_0080)	Reset value	Reset value	Reset value	Reset value
TG_VSYNC_BOT_HDMI_H (0x1458_0084)	Reset value	Reset value	Reset value	Reset value
TG_FIELD_TOP_HDMI_L (0x1458_0088)	Reset value	Reset value	Reset value	Reset value
TG_FIELD_TOP_HDMI_H (0x1458_008C)	Reset value	Reset value	Reset value	Reset value
TG_FIELD_BOT_HDMI_L (0x1458_0090)	Reset value	Reset value	Reset value	Reset value
TG_FIELD_BOT_HDMI_H (0x1458_0094)	Reset value	Reset value	Reset value	Reset value
TG_3D_FP (0x1458_0094)	Reset value	Reset value	Reset value	Reset value

53.7 HDMI PHY Configuration

You should configure HDMI PHY by using dedicated I2C, which is only used in TX mode. The address of HDMI PHY is 0x70..

NOTE: If User doesn't use HDMI, PHY should be power down mode. User should set Reg1D of I2C Register for PHY to 0x1F. Refer to 1.7.1 Selected i2C Register Control for more information.

Figure 53-4 illustrates the sequence of I2C data.

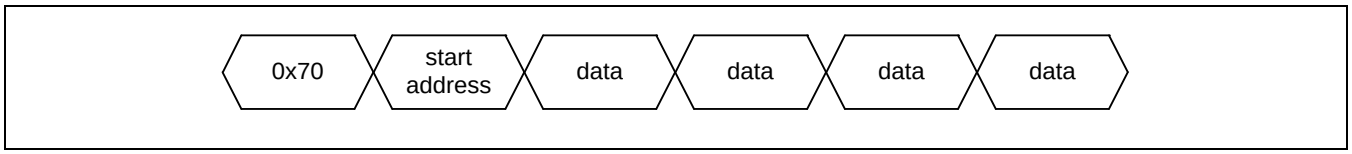


Figure 53-4 Sequence of I2C Data

Due to the security policy, the value of configuration is only opened, as described in *Table 53-3*.

Table 53-3 describes HDMI PHY configuration for 24 MHz OSC_In.

Table 53-3 HDMI PHY Configuration Table for 24 MHz OSC_In

Addr	27 MHz (Pixel Clock Ratio)	27.027 MHz	74.176 MHz	74.25 MHz	148.5 MHz
	8b	8b	8b	8b	8b
0x01	11h	91h	91h	91h	91h
0x02	2Dh	2Dh	1Fh	1Fh	1Fh
0x03	75h	72h	10h	10h	00h
0x04	40h	40h	40h	40h	40h
0x05	01h	64h	5Bh	40h	40h
0x06	00h	12h	EFh	F8h	F8h
0x07	08h	08h	08h	08h	08h
0x08	82h	43h	81h	81h	81h
0x09	00h	20h	20h	20h	20h
0x0A	0Eh	0Eh	B9h	BAh	BAh
0x0B	D9h	D9h	D8h	D8h	D8h
0x0C	45h	45h	45h	45h	45h
0x0D	A0h	A0h	A0h	A0h	A0h
0x0E	ACh	ACh	ACh	ACh	ACh
0x0F	80h	80h	80h	80h	80h
0x10	08h	08h	08h	08h	08h
0x11	80h	80h	80h	80h	80h
0x12	11h	11h	11h	11h	11h

Addr	27 MHz (Pixel Clock Ratio)	27.027 MHz	74.176 MHz	74.25 MHz	148.5 MHz
	8b	8b	8b	8b	8b
0x13	84h	84h	84h	84h	84h
0x14	02h	02h	02h	02h	02h
0x15	22h	22h	22h	22h	22h
0x16	44h	44h	44h	44h	44h
0x17	86h	86h	86h	86h	86h
0x18	54h	54h	54h	54h	54h
0x19	E4h	E3h	A6h	A5h	4Bh
0x1A	24h	24h	24h	24h	25h
0x1B	00h	00h	01h	01h	03h
0x1C	00h	00h	00h	00h	00h
0x1D	00h	00h	00h	00h	00h
0x1E	01h	01h	01h	01h	01h
0x1F	80h	80h	80h	80h	80h

Address 0x1f specifies the PHY_START control. If you configure PHY, the address 0x1f should be 0x0.

NOTE: It is necessary to adjust PHY configuration according to the PCB board environment.

53.7.1 Selected i2C Register Control

Name	Code	Description
REF_SEL (Reg01 bit<6>)	0	REF_OSC
	1	INT_CLK
	Comments : Internal Reference Clock Selection	
CLK_SEL[1:0] (Reg01 bit<5:4>)	0X	REF_OSC or INT_CLK
	10	External Pixel Clock
	11	External TMDS Clock
	Comments : HDMI TX PHY Internal PLL Input Clock Selection	
REF_CKO_SEL (Reg09 bit<7>)	0	REF_OSC
	1	Internal Reference Clock
	Comments : REF_CKO Selection	
TX_AMP_LVL[4:0] (Reg10 bit<3:0>, Reg0F bit<7>)	00000	Min Value
	11111	Max Value
	Comments : TMDS Data Amplitude Control	
TX_LVL_CH0[1:0] TX_LVL_CH1[1:0] TX_LVL_CH2[1:0] (Reg04 bit<7:6> Reg13 bit<1:0> Reg17 bit<1:0>)	00	Min Value
	11	Max Value
	Comments : TMDS Data Amplitude Fine Control for Each Channel	
TX_EMP_LVL[3:0] (Reg10 bit<7:4>)	00000	Min Value
	11111	Max Value
	Comments : TMDS Data Pre-Emphasis Level Control	
TX_CLK_LVL[4:0] (Reg17 bit<7:3>)	00000	Min Value
	11111	Max Value
	Comments : TMDS Clock Amplitude Control	
TX_RES[1:0] (Reg0F bit<5:4>)	00	Max Resistance
	11	Min Resistance
	Comments : TMDS Data Source Termination Resistor Control	

Name	Code	Description
I2C_PDEN NOTE: (Reg1D bit<7>)	0	Disable
	1	Enable
	Comments : If I2C_PDEN = 1, power down of each building blocks of PHY can be controlled I2C Reg1D bit<6:4>, bit<2:0>	
PLL PD NOTE: TX_CLKSER_PD TX_CLKDRV_PD TX_DRV_PD TX_SER_PD TX_CLK_PD (Reg1D bit<6:4>,bit<2:0>)	0	Set Normal Status
	1	Set Power Down Status
	Comments : Bit<6> : PLL PD (PLL & Bias Block Power Down) Bit<5> : TX_CLKSER_PD (Clock Serializer Power Down) Bit<4> : TX_CLKDRV_PD (TMDS Clock Driver Power Down) Bit<2> : TX_DRV_PD (TMDS Data Driver Power Down) Bit<1> : TX_SER_PD (TMDS Data Serializer Power Down) Bit<0> : TX_CLK_PD (TX Internal Clock Buffer / Divider Power Down)	
TESTEN (Reg1E bit<7>)	0	Normal Operation Mode
	1	PHY Test Mode
	Comments : PHY Test Mode Enable	
TEST[6:0] (Reg1E bit<6:0>)	Comments : PHY Test Mode Control Signal For more detail information, contact with S.LSI DIPD Team	
TX_LPEN[1:0] (Reg11 bit<1:0>)	01	CH0 Selection
	10	CH1 Selection
	11	CH2 Selection
	Comments : TX Channel Selection for BIST Loopback Test Mode	
RX_LPEN[1:0] (Reg11 bit<3:2>)	00	CH0 Selection
	01	CH1 Selection
	1X	CH2 Selection
	Comments : RX Channel Selection for BIST Loopback Test Mode	
MODE SET DONE (Reg1F bit<7>)	Comments : An indicator of I2C setting state. Refer to the Figure 9.	

NOTE: To reduce power consumption in power down mode, Set Reg1D bit[7], bit[6], bit[5], bit[4], bit[2], bit[1], and bit[0] should be set to 1.

53.8 Block Diagram of Clock Strategy for HDMI Tx

Figure 53-5 illustrates the block diagram of clock scheme in Exynos 4412 SCP.

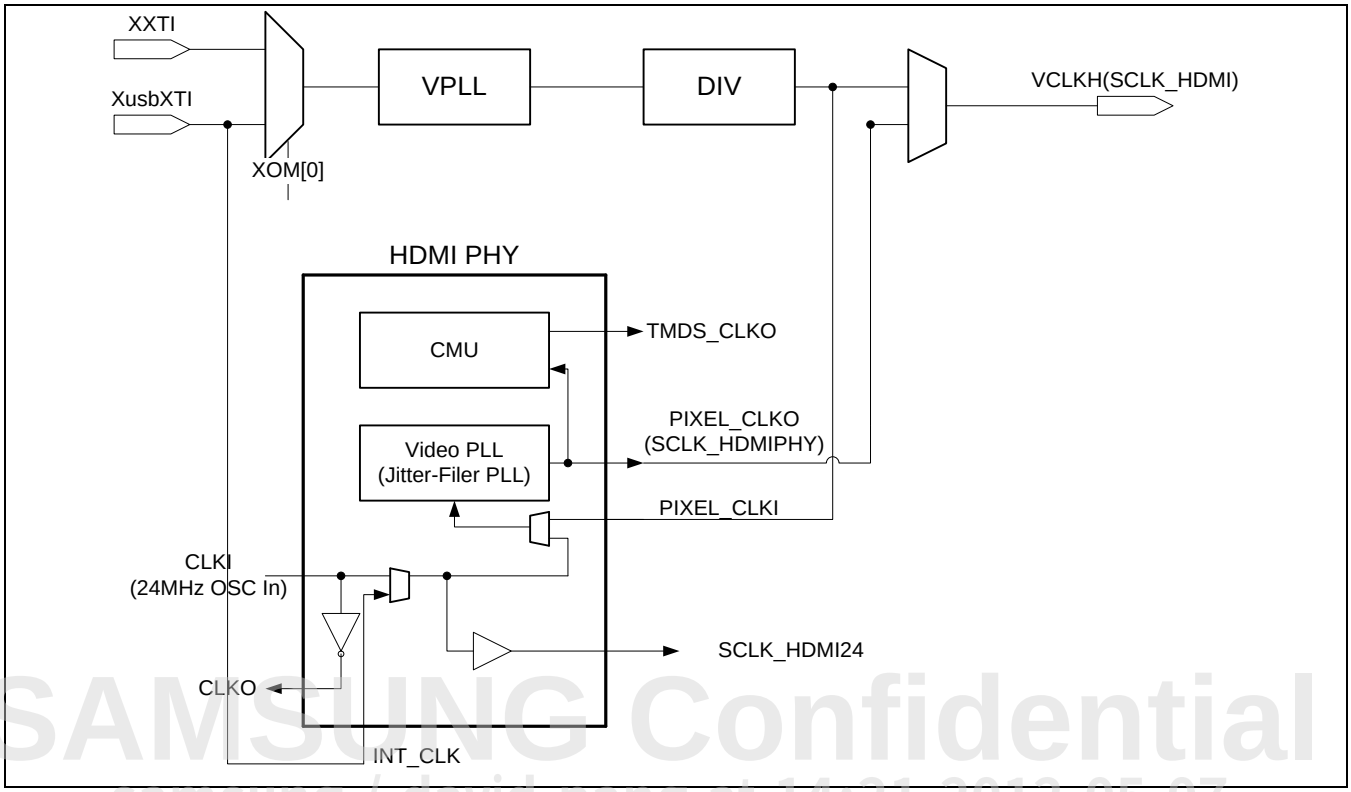


Figure 53-5 Block Diagram of HDMI Tx Clock Scheme in Exynos 4412 SCP

The HDMI link part uses pixel and TMDS clock. It feeds from HDMI PHY. You should configure it before use. VCLKH (MIXER pixel clock, HDMI pixel clock) are synchronous. Thus, it feeds same clock through VCLKH. Refer to *Table 53-4* for more information on pixel frequency.

Table 53-4 lists frequency summary in use (VCLKH usage frequency).

Table 53-4 Frequency Summary in Use (VCLKH Usage Frequency)

Vertical Freq.	Format	Pixel Freq.
59.94 Hz	480p	27 MHz (or 54 MHz)
	720p	74.175 MHz
	720p (3D – SB.H, TB)	74.250 MHz
	1080I	74.175 MHz
	1080p (3D – SB.H, TB)	148.500 MHz
60.00 Hz	480p	27 MHz (or 54 MHz)
	720p	74.250 MHz
	720p (3D – SB.H, TB)	74.250 MHz
	1080I	74.250 MHz
	1080p (3D – SB.H, TB)	148.500 MHz
50.00 Hz	576p	27 MHz (or 54 MHz)
	720p	74.250 MHz
	720p (3D-TB)	74.250 MHz
	1080I	74.250 MHz
29.97 Hz	1080p	74.175 MHz
30 Hz	1080p	74.250 MHz
25 Hz	1080p	74.250 MHz
24 Hz	1080p (3D – SB.H, TB)	74.250 MHz

Where:

- 3D-SB.H: Stereoscopy Side-by-side (Half)
- 3D-TB: Stereoscopy Top and Bottom

53.9 SPDIF (Auxiliary Information)

This section describes Frame Format.

53.9.1 Frame Format

A frame consists of two sub-frames. The transmission rate of frames corresponds exactly to the source sampling frequency. In the 2 channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive sub-frames.

Usually, sub-frames that are related to channel 1 (left or "A" channel in stereophonic operation and primary channel in monophonic operation) use preamble M. However, the preamble is changed to preamble B after every 192 frames. This unit is composed of 192 frames. It defines the block structure used to organize the channel status information.

On the other hand, sub-frames of channel 2 (right or "B" in stereophonic operation and secondary channel in monophonic operation) always use preamble W. In single channel operation mode and broadcasting studio environment, the frame format is identical to 2 channel mode. The data is only carried in channel 1. In the sub-frames allocated to channel 2, time slot 28 (validity flag) is set to logical "1" (invalid).

Figure 53-6 illustrates the frame format.

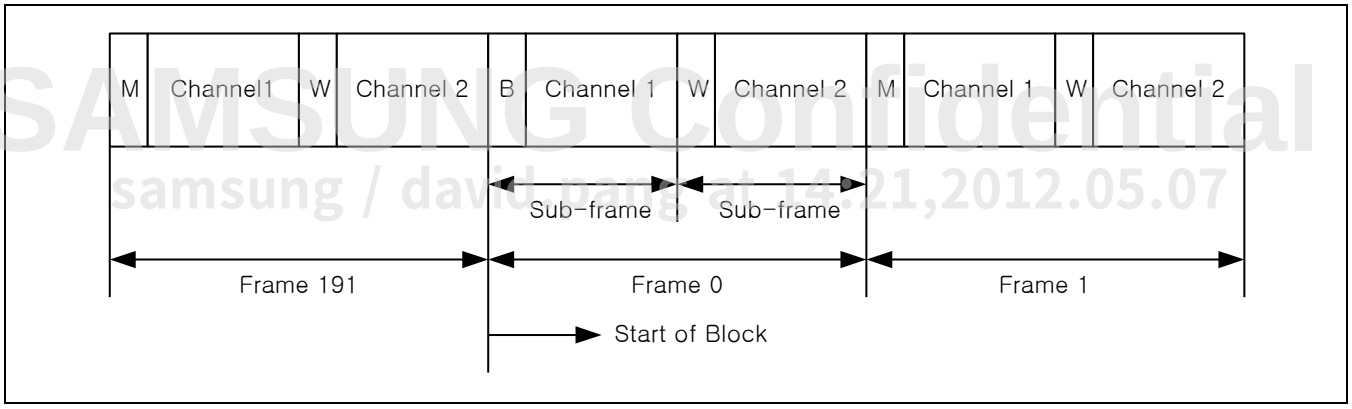


Figure 53-6 Frame Format

53.9.1.1 Sub-Frame Format (IEC 60958)

Each sub-frame is divided into 32 time slots numbered from 0 to 31. Time slots from 0 to 3 carry one of the three permitted preambles. These slots affect the synchronization of sub-frames, frames, and blocks. Time slots from 4 to 27 carry the audio sample word in linear 2's complement representation.

The most significant bit (MSB) is carried by time slot 27. When a 24-bit coding range is used, the least significant bit (LSB) is in the time slot 4.

When a 20-bit coding range is sufficient, the LSB is in the time slot 8. Time slots from 4 to 7 may be used for other applications. Under these circumstances, the bits in the time slots 4 to 7 are designated auxiliary sample bits.

When the source provides fewer bits than what the interface allows (24 or 20), the unused LSBs are set to a logical "0". By this procedure, the equipment using different numbers of bits can be connected together.

- Time slot 28 carries the validity flag associated with audio sample word. This flag is set to logical "0" when audio sample is reliable.
- Time slot 29 carries one bit of user data associated with audio channel. It is transmitted in the same sub-frame. The default value of user bit is logical "0".
- Time slot 30 carries one bit of channel status words associated with audio channel. It is transmitted in the same sub-frame.
- Time slot 31 carries a parity bit such that time slots from 4 to 31 (inclusive) will carry an even number of ones and an even number of zeroes.

Figure 53-7 illustrates the sub-frame format.

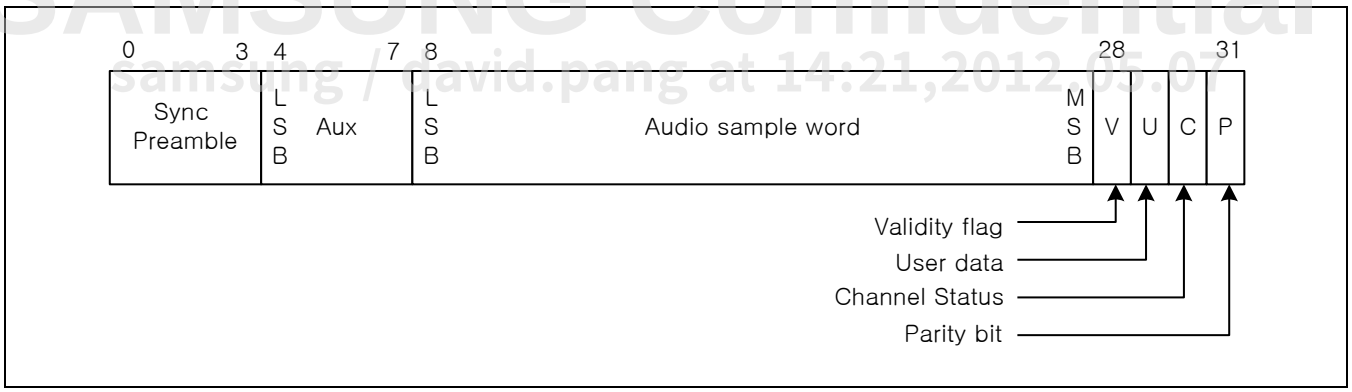


Figure 53-7 Sub-Frame Format

53.9.1.2 Channel Status Block (IEC-60958-3)

Channel Status Block specifies the aggregation of Channel Status bit in each sub-frame, as illustrated in *Figure 53-8*. As one frame consists of 192 frames, one channel status block can be obtained for one channel.

This block holds the information of stream being transmitted such as application, stream type, sampling frequency, word length, and so on.

Figure 53-8 illustrates the channel status block extract from SPDIF stream.

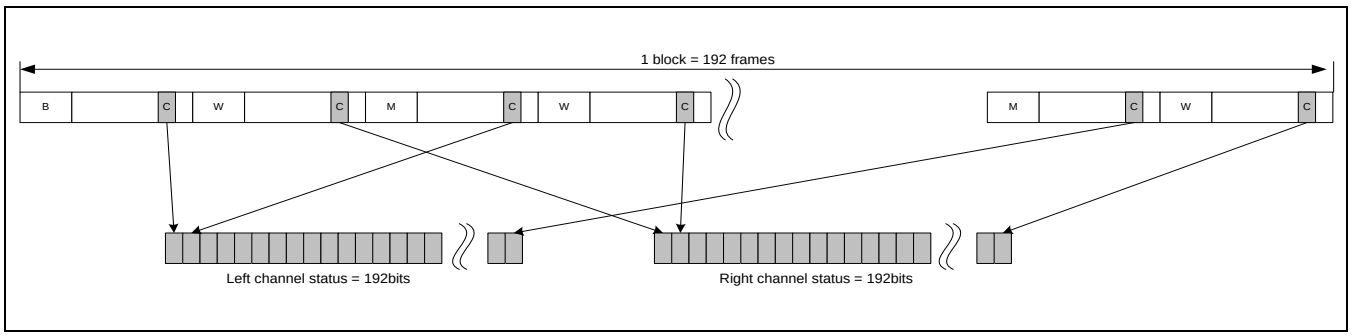


Figure 53-8 Channel Status Block Extract from SPDIF Stream

Figure 53-9 illustrates the channel status block

Byte									
0		a = "0"	b = "0"	c	d			Mode = "0 0"	
	bit	0	1	2	3	4	5	6	7
1		Category code							
	bit	8	9	10	11	12	13	14	15
2		Source number				Channel number			
	bit	16	17	18	19	20	21	22	23
3		Sampling frequency				Clock accuracy			
	bit	24	25	26	27	28	29	30	31
4		Word length				Original sampling frequency			
	bit	32	33	34	35	36	37	38	39
5									
	bit	40	41	42	43	44	45	46	47
6									
	bit	48	49	50	51	52	53	54	55
7									
	bit	56	57	58	59	60	61	62	63
8									
	bit	64	65	66	67	68	69	70	71
9									
	bit	72	73	74	75	76	77	78	79
10									
	bit	80	81	82	83	84	85	86	87
11									
	bit	88	89	90	91	92	93	94	95
12									
	bit	96	97	98	99	100	101	102	103
13									
	bit	104	105	106	107	108	109	110	111
14									
	bit	112	113	114	115	116	117	118	119
15									
	bit	120	121	122	123	124	125	126	127
16									
	bit	128	129	130	131	132	133	134	135
17									
	bit	136	137	138	139	140	141	142	143
18									
	bit	144	145	146	147	148	149	150	151
19									
	bit	152	153	154	155	156	157	158	159
20									
	bit	160	161	162	163	164	165	166	167
21									
	bit	168	169	170	171	172	173	174	175
22									
	bit	176	177	178	179	180	181	182	183
23									
	bit	184	185	186	187	188	189	190	191
		a: use of channel status block. b: linear PCM identification.			c: copyright information. d: additional format information.				

Figure 53-9 Channel Status Block

53.9.1.3 Channel Coding

Time slots from 4 to 31 are encoded in bi-phase-mark to:

- Minimize the DC component on transmission line
- Facilitate clock recovery from the data stream
- Make the interface insensitive to polarity of connections

Each bit that needs to be transmitted is represented by a symbol comprising two consecutive binary states.

The first state of a symbol is always different from the second state of previous symbol. The second state of the symbol is identical to the first when the bit to be transmitted is logical "0" and different from the first when the bit is logical "1".

Figure 53-10 illustrates the channel coding.

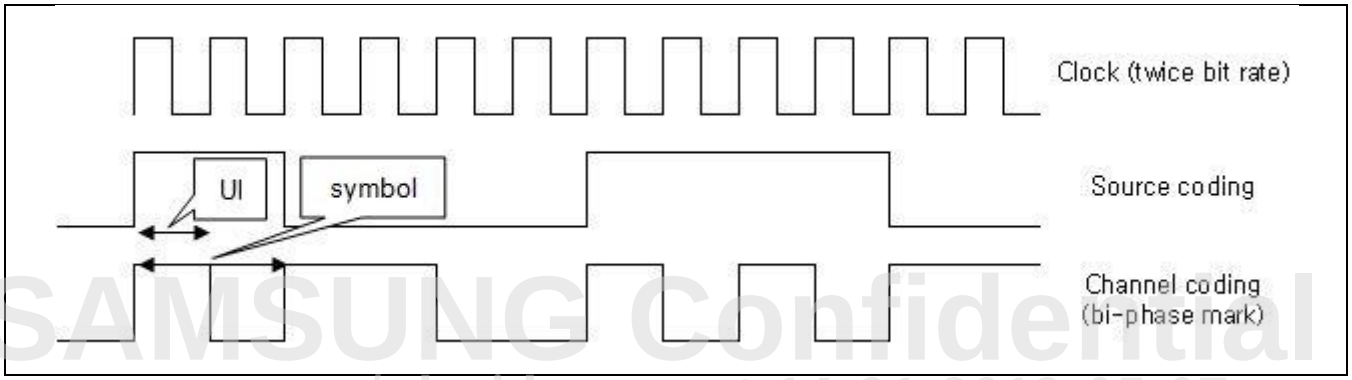


Figure 53-10 Channel Coding

53.9.1.4 Preamble

Preambles are specific patterns that provide synchronization and identification of sub-frames and blocks. A set of three preambles is used. These preambles are transmitted in the time allocated to four time slots (time slots 0 to 3) and are represented by eight successive states.

The first state of the preamble is always different from the second state of the previous symbol. Similar to the bi-phase code, these preambles are DC independent and provide clock recovery. They differ in at least two states from any valid bi-phase sequence.

53.9.1.5 Non-Linear PCM Encoded Source (IEC 61937)

The non-linear PCM encoded audio bit stream is transferred by using the basic 16-bit data area of the IEC 60958 sub frames, that is, in time slots from 12 to 27. Each IEC 60958 frame transfers 32-bit of non-PCM data in the consumer application mode.

When SPDIF bit stream conveys linear PCM audio. The frequency of the symbol is 64 times the PCM sampling frequency (32 time slots per PCM sample time two channels).

When interface conveys non-linear PCM encoded audio bit stream, the frequency of the symbol is 64 times the sampling rate of encoded audio within that bit stream.

If the interface that contains audio with low sampling frequency conveys a non-linear PCM encoded audio bit stream then the frequency of the symbol is 128 times the sampling rate of encoded audio within that bit stream.

Each data burst contains a burst-preamble consisting of four 16-bit words (Pa, Pb, Pc, and Pd); followed by the burst-payload that contains data of an encoded audio frame.

The burst-preamble consists of four mandatory fields. They are:

- Pa and Pb represent a synchronization word.
- Pc provides information about the type of data and some information/ control for receiver.
- Pd provides the length of burst-payload. It is limited to 216 (= 65,535) bits.

The four preamble words are contained in two sequential SPDIF frames.

The frame that begins with data-burst contains preamble word Pa in sub-frame 1 and Pb in sub-frame 2.

The next frame contains Pc in sub-frame 1 and Pd in sub-frame 2. When placed into a SPDIF sub-frame, the MSB of a 16-bit burst-preamble is placed into time slot 27 and the LSB is placed into time slot 12.

Figure 53-11 illustrates the non-linear PCM format.

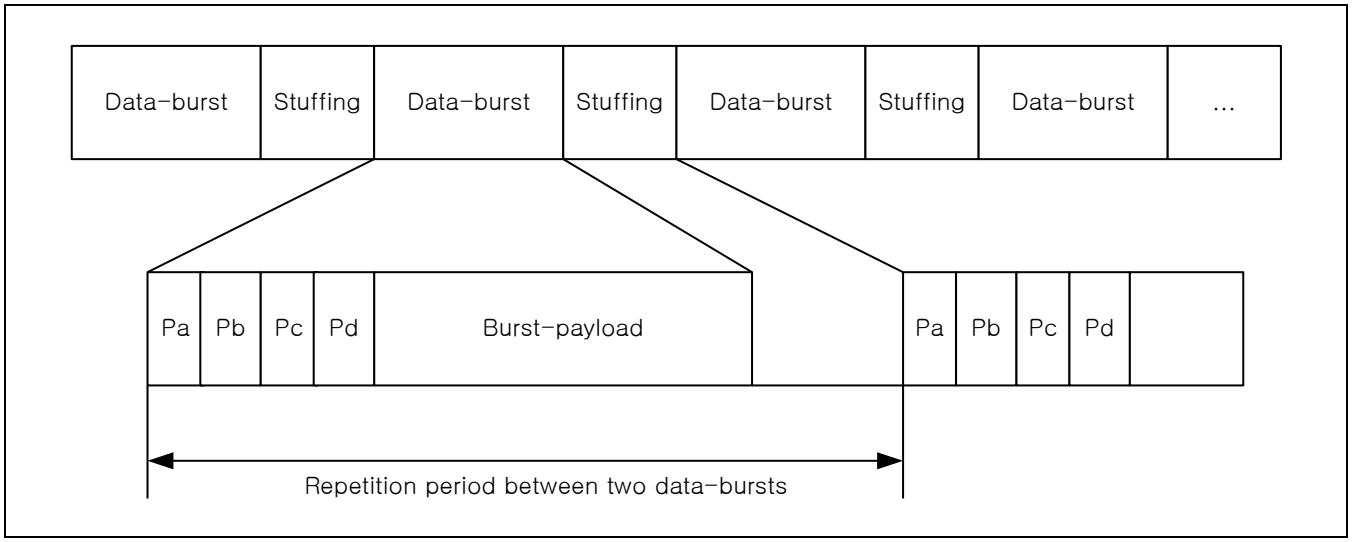


Figure 53-11 **Non-Linear PCM Format**

53.10 Register Description

53.10.1 Register Map Summary

- Base Address: 0x12D0_0000

Register	Offset	Description	Reset Value
Control Registers			
INTC_CON_0	0x0000	Specifies interrupt control register.	0x00
INTC_FLAG_0	0x0004	Specifies interrupt flag register.	0x00
HDCP_KEY_LOAD	0x0008	Specifies HDCP key status.	0x00
HPD_STATUS	0x000C	Specifies value of HPD signal.	0x00
INTC_CON_1	0x0010	Interrupt Control Register 1	0x00
INTC_FLAG_1	0x0014	Interrupt Flag Register 1	0x00
PHY_STATUS_0	0x0020	PHY Status Register 0	0x00
PHY_STATUS_AFC	0x0024	PHY AFC Status Register	0x00
PHY_STATUS_PLL	0x0028	PHY PLL Status Register	0x00
PHY_CON_0	0x0030	PHY Control Register	0x00
HPD_CTRL	0x0040	HPD Signal Control Register	0x02
HPD_ST	0x0044	HPD Status Register	0x00
HPD_TH_x	0x0050	HPD Filter Threshold Value Register	0x00
AUDIO_CLKSEL	0x0070	Selects the audio system clock.	0x00
HDMI_PHY_RSTOUT	0x0074	Specifies the HDMI PHY reset out.	0x00
HDMI_PHY_PLL	0x0078	Specifies the HDMI PHY PLL monitor.	0x00
HDMI_PHY_AFC	0x007C	Specifies the HDMI PHY CMU monitor.	0x00
HDMI_CORE_RSTOUT	0x0080	Specifies the HDMI TX core software reset.	0x01

- Base Address: 0x12D1_0000

Register	Offset	Description	Reset Value
HDMI Core Registers			
Control Registers			
HDMI_CON_0	0x0000	Specifies the HDMI system control register 0.	0x00
HDMI_CON_1	0x0004	Specifies the HDMI system control register 1.	0x00
HDMI_CON_2	0x0008	Specifies the HDMI system control register 2.	0x00
STATUS	0x0010	Specifies the HDMI system status register.	0x00
STATUS_EN	0x0020	Specifies the HDMI system status enable register.	0x00
HPD	0x0030	Specifies the HPD control register.	0x00
MODE_SEL	0x0040	Selects the HDMI/DVI mode.	0x00
ENC_EN	0x0044	Specifies the HDCP encryption enable register.	0x00

Register	Offset	Description	Reset Value
Video Related Registers			
HDMI_YMAX	0x0060	Specifies the maximum Y (or R, G, B) pixel value.	0xeb
HDMI_YMIN	0x0064	Specifies the minimum Y (or R, G, B) pixel value.	0x10
HDMI_CMAX	0x0068	Specifies the maximum Cb/Cr pixel value.	0xf0
HDMI_CMIN	0x006C	Specifies the minimum Cb/Cr pixel value.	0x10
H_BLANK_0	0x00A0	Specifies the horizontal blanking 0 setting.	0x00
H_BLANK_1	0x00A4	Specifies the horizontal blanking 1 setting.	0x00
V2_BLANK_0	0x00B0	Specifies the vertical 2 blanking 0 setting.	0x00
V2_BLANK_1	0x00B4	Specifies the vertical2 blanking 1 setting.	0x00
V1_BLANK_0	0x00B8	Specifies the vertical1 blanking 0 setting.	0x00
V1_BLANK_1	0x00BC	Specifies the vertical1 blanking 1 setting.	0x00
V_LINE_0	0x00C0	Specifies the horizontal line 0 setting.	0x00
V_LINE_1	0x00C4	Specifies the horizontal line 1 setting.	0x00
H_LINE_0	0x00C8	Specifies the vertical line 0 setting.	0x00
H_LINE_1	0x00CC	Specifies the vertical line 1 setting.	0x00
HSYNC_POL	0x00E0	Specifies the Horizontal sync polarity control register	0x00
VSYNC_POL	0x00E4	Specifies the vertical sync polarity control register.	0x00
INT_PRO_MODE	0x00E8	Specifies the Interlace/ Progressive control register.	0x00
V_BLANK_F0_0	0x0110	Specifies vertical blanking setting for bottom field 0.	0xff
V_BLANK_F0_1	0x0114	Specifies the vertical blanking setting for bottom field 1.	0x1f
V_BLANK_F1_0	0x0118	Specifies the vertical blanking setting for bottom field 0.	0xff
V_BLANK_F1_1	0x011C	Specifies the vertical blanking setting for bottom field 1.	0x1f
H_SYNC_START_0	0x0120	Specifies horizontal sync generation setting.	0x00
H_SYNC_START_1	0x0124	Specifies horizontal sync generation setting.	0x00
H_SYNC_END_0	0x0128	Specifies horizontal sync generation setting.	0x00
H_SYNC_END_1	0x012C	Specifies horizontal sync generation setting.	0x00
V_SYNC_LINE_BEF_2_0	0x0130	Specifies vertical sync generation for top field or frame	0xff
V_SYNC_LINE_BEF_2_1	0x0134	Specifies vertical sync generation for top field or frame	0x1f
V_SYNC_LINE_BEF_1_0	0x0138	Specifies vertical sync generation for top field or	0xff

Register	Offset	Description	Reset Value
		frame	
V_SYNC_LINE_BEF_1_1	0x013C	Specifies vertical sync generation for top field or frame	0x1f
V_SYNC_LINE_AFT_2_0	0x0140	Specifies vertical sync generation for bottom field-vertical position	0xff
V_SYNC_LINE_AFT_2_1	0x0144	Specifies vertical sync generation for bottom field-vertical position	0x1f
V_SYNC_LINE_AFT_1_0	0x0148	Specifies vertical sync generation for bottom field-vertical position	0xff
V_SYNC_LINE_AFT_1_1	0x014C	Specifies vertical sync generation for bottom field-vertical position	0x1f
V_SYNC_LINE_AFT_PXL_2_0	0x0150	Specifies vertical sync generation for bottom field-horizontal position	0xff
V_SYNC_LINE_AFT_PXL_2_1	0x0154	Specifies vertical sync generation for bottom field-horizontal position	0x1f
V_SYNC_LINE_AFT_PXL_1_0	0x0158	Specifies vertical sync generation for bottom field-horizontal position	0xff
V_SYNC_LINE_AFT_PXL_1_1	0x015C	Specifies vertical sync generation for bottom field-horizontal position	0x1f
V_BLANK_F2_0	0x0160	Specifies vertical blanking setting for third field	0xff
V_BLANK_F2_1	0x0164	Specifies vertical blanking setting for third field	0x1f
V_BLANK_F3_0	0x0168	Specifies vertical blanking setting for third field	0xff
V_BLANK_F3_1	0x016C	Specifies vertical blanking setting for third field	0x1f
V_BLANK_F4_0	0x0170	Specifies vertical blanking setting for fourth field	0xff
V_BLANK_F4_1	0x0174	Specifies vertical blanking setting for fourth field	0x1f
V_BLANK_F5_0	0x0178	Specifies vertical blanking setting for fourth field	0xff
V_BLANK_F5_1	0x017C	Specifies vertical blanking setting for fourth field	0x1f
V_SYNC_LINE_AFT_3_0	0x0180	Specifies vertical sync generation for third field-vertical position	0xff
V_SYNC_LINE_AFT_3_1	0x0184	Specifies vertical sync generation for third field-vertical position	0x1f
V_SYNC_LINE_AFT_4_0	0x0188	Specifies vertical sync generation for third field-vertical position	0xff
V_SYNC_LINE_AFT_4_1	0x018C	Specifies vertical sync generation for third field-vertical position	0x1f
V_SYNC_LINE_AFT_5_0	0x0190	Specifies vertical sync generation for fourth field-vertical position	0xff
V_SYNC_LINE_AFT_5_1	0x0194	Specifies vertical sync generation for fourth field-vertical position	0x1f
V_SYNC_LINE_AFT_6_0	0x0198	Specifies vertical sync generation for fourth field-vertical position	0xff

Register	Offset	Description	Reset Value
V_SYNC_LINE_AFT_6_1	0x019C	Specifies vertical sync generation for fourth field-vertical position	0x1f
V_SYNC_LINE_AFT_PXL_3_0	0x01A0	Specifies vertical sync generation for third field-horizontal position	0xff
V_SYNC_LINE_AFT_PXL_3_1	0x01A4	Specifies vertical sync generation for third field-horizontal position	0x1f
V_SYNC_LINE_AFT_PXL_4_0	0x01A8	Specifies vertical sync generation for third field-horizontal position	0xff
V_SYNC_LINE_AFT_PXL_4_1	0x01AC	Specifies vertical sync generation for third field-horizontal position	0x1f
V_SYNC_LINE_AFT_PXL_5_0	0x01B0	Specifies vertical sync generation for fourth field-horizontal position	0xff
V_SYNC_LINE_AFT_PXL_5_1	0x01B4	Specifies vertical sync generation for fourth field-horizontal position	0x1f
V_SYNC_LINE_AFT_PXL_6_0	0x01B8	Specifies vertical sync generation for fourth field-horizontal position	0xff
V_SYNC_LINE_AFT_PXL_6_1	0x01BC	Specifies vertical sync generation for fourth field-horizontal position	0x1f
VACT_SPACE1_0	0x01C0	Specifies first vertical active space start line	0xff
VACT_SPACE1_1	0x01C4	Specifies first vertical active space end line	0x1f
VACT_SPACE2_0	0x01C8	Specifies first vertical active space start line	0xff
VACT_SPACE2_1	0x01CC	Specifies first vertical active space end line	0x1f
VACT_SPACE3_0	0x01D0	Specifies second vertical active space start line	0xff
VACT_SPACE3_1	0x01D4	Specifies second vertical active space end line	0x1f
VACT_SPACE4_0	0x01D8	Specifies second vertical active space start line	0xff
VACT_SPACE4_1	0x01DC	Specifies second vertical active space end line	0x1f
VACT_SPACE5_0	0x01E0	Specifies third vertical active space start line	0xff
VACT_SPACE5_1	0x01E4	Specifies third vertical active space end line	0x1f
VACT_SPACE6_0	0x01E8	Specifies third vertical active space start line	0xff
VACT_SPACE6_1	0x01EC	Specifies third vertical active space end line	0x1f
GCP_CON	0x0200	Specifies General Control Packet (GCP) control register	0x04
GCP_BYTE1	0x0210	Specifies GCP body	0x00
GCP_BYTE2	0x0214	Specifies GCP body	0x00
GCP_BYTE3	0x0218	Specifies GCP body	0x00
Audio Related Registers			
ASP_CON	0x0300	Specifies the Audio Sample Packet (ASP) control register.	0x00
ASP_SP_FLAT	0x0304	Specifies the ASP sp_flat bit control.	0x00
ASP_CHCFG0	0x0310	Specifies the ASP audio channel configuration0.	0x08

Register	Offset	Description	Reset Value
ASP_CHCFG1	0x0314	Specifies the ASP audio channel configuration1.	0x1a
ASP_CHCFG2	0x0318	Specifies the ASP audio channel configuration2.	0x2c
ASP_CHCFG3	0x031C	Specifies the ASP audio channel configuration3.	0x3e
ACR_CON	0x0400	Specifies the ACR packet control register.	0x00
ACR_MCTS0	0x0410	Specifies the measured CTS0 value.	0x01
ACR_MCTS1	0x0414	Specifies the measured CTS1 value.	0x00
ACR_MCTS2	0x0418	Specifies the measured CTS2 value.	0x00
ACR_N0	0x0430	Specifies the N0 value for ACR packet.	0xe8
ACR_N1	0x0434	Specifies the N1 value for ACR packet.	0x03
ACR_N2	0x0438	Specifies the N2 value for ACR packet.	0x00
Packet Related Registers			
ACP_CON	0x0500	Specifies the ACP packet control register.	0x00
ACP_TYPE	0x0514	Specifies the ACP packet header.	0x00
ACP_DATA00 to ACP_DATA16	0x0520 to 0x0560	Specifies the ACP packet body.	0x00
ISRC_CON	0x0600	Specifies the ACR packet control register.	0x00
ISRC1_HEADER1	0x0614	Specifies the ISCR1 packet header.	0x00
ISRC1_DATA00 to ISRC1_DATA15	0x0620 to 0x065C	Specifies the ISRC1 packet body.	0x00
ISRC2_DATA00 to ISRC2_DATA15	0x06A0 to 0x06DC	Specifies the ISRC2 packet body.	0x00
AVI_CON	0x0700	Specifies the AVI packet control register.	0x00
AVI_HEADER0	0x0710	Specifies the AVI packet Header0	0x00
AVI_HEADER1	0x0714	Specifies the AVI packet Header1	0x00
AVI_HEADER2	0x0718	Specifies the AVI packet Header2	0x00
AVI_CHECK_SUM	0x071C	Specifies the AVI packet checksum.	0x00
AVI_BYTE01 to AVI_BYTE13	0x0720 to 0x0750	Specifies the AVI packet body.	0x00
AUI_CON	0x0800	Specifies the AUI packet control register.	0x00
AUI_HEADER0	0x0810	Specifies AUI packet header0.	0x00
AUI_HEADER1	0x0814	Specifies AUI packet header1.	0x00
AUI_HEADER2	0x0818	Specifies AUI packet header2.	0x00
AUI_CHECK_SUM	0x081C	Specifies the AUI packet checksum.	0x00
AUI_BYTE1 to	0x0820 to	Specifies the AUI packet body.	0x00

Register	Offset	Description	Reset Value
AUI_BYTE12	0x084C		
MPG_CON	0x0900	Specifies the ACR packet control register.	0x00
MPG_CHECK_SUM	0x091C	Specifies the MPG packet checksum.	0x00
MPG_DATA1 to MPG_DATA6	0x0920 to 0x0934	Specifies the MPG packet body.	0x00
SPD_CON	0x0A00	Specifies the SPD packet control register.	0x00
SPD_HEADER0	0x0A10	Specifies the SPD packet header0.	0x00
SPD_HEADER1	0x0A14	Specifies the SPD packet header1.	0x00
SPD_HEADER2	0x0A18	Specifies the SPD packet heade2r.	0x00
SPD_DATA00 to SPD_DATA27	0x0A20 to 0x0A8C	Specifies the SPD packet body.	0x00
GAMUT_CON	0x0B00	Specifies GAMUT packet control register	0x00
GAMUT_HEADER0	0x0B10	Specifies GAMUT packet header0	0x00
GAMUT_HEADER1	0x0B14	Specifies GAMUT packet header1	0x00
GAMUT_HEADER2	0x0B18	Specifies GAMUT packet header2	0x00
GAMUT_METADATA00 to GAMUT_METADATA27	0x0B20 to 0x0B8C	Specifies GAMUT packet body	0x00
VSI_CON	0x0C00	Specifies VSI packet control register	0x00
VSI_HEADER0	0x0C10	Specifies VSI packet header0	0x00
VSI_HEADER1	0x0C14	Specifies VSI packet header1	0x00
VSI_HEADER2	0x0C18	Specifies VSI packet header2	0x00
VSI_DATA00 to VSI_DATA27	0x0C20 to 0x0C8C	Specifies VSI packet body	0x00
VIDEO_PATTERN_GEN	0x0D04	Specifies video pattern generation register	0x00
An_Seed_Sel	0x0E48	Specifies An seed selection register.	0xFF
An_Seed_0	0x0E58	Specifies An seed0 value register	0x00
An_Seed_1	0x0E5C	Specifies An seed1 value register	0x00
An_Seed_2	0x0E60	Specifies An seed2 value register	0x00
An_Seed_3	0x0E64	Specifies An seed3 value register	0x00
HDCP Related Registers			
HDCP_SHA1_00 to HDCP_SHA1_19	0x7000 to 0x704C	Specifies the SHA-1 value from repeater.	0x00
HDCP_KSV_LIST_0 to HDCP_KSV_LIST_4	0x7050 to 0x7060	Specifies the KSV list from repeater.	0x00

Register	Offset	Description	Reset Value
HDCP_KSV_LIST_CON	0x7064	Controls the KSV list.	0x01
HDCP_SHA_RESULT	0x7070	Specifies the SHA-1 checking result register.	0x00
HDCP_CTRL1	0x7080	Specifies the HDCP control register1.	0x00
HDCP_CTRL2	0x7084	Specifies the HDCP control register2.	0x00
HDCP_CHECK_RESULT	0x7090	Verifies result of Ri and Pj values.	0x00
HDCP_BKSV_0 to HDCP_BKSV_4	0x70A0 to 0x70B0	Specifies the KSV of Rx.	0x00
HDCP_AKSV_0 to HDCP_AKSV_4	0x70C0 to 0x70D0	Specifies the KSV of Tx.	0x00
HDCP_An_0 to HDCP_An_7	0x70E0 to 0x70FC	Specifies the An value.	0x00
HDCP_BCAPS	0x7100	Specifies the BCAPS from Rx.	0x00
HDCP_BSTATUS_0	0x7110	Specifies the BSTATUS0 from Rx.	0x00
HDCP_BSTATUS_1	0x7114	Specifies the BSTATUS1 from Rx.	0x00
HDCP_Ri_0	0x7140	Specifies the Ri0 value of Tx.	0x00
HDCP_Ri_1	0x7144	Specifies the Ri1 value of Tx.	0x00
HDCP_I2C_INT	0x7180	Specifies the I2C interrupt flag.	0x00
HDCP_AN_INT	0x7190	Specifies the An value ready interrupt flag.	0x00
HDCP_WATCGDOG_INT	0x71A0	Specifies the watchdog interrupt flag.	0x00
HDCP_Ri_INT	0x71B0	Specifies the Ri value update interrupt flag.	0x00
HDCP_Ri_Compare_0	0x71D0	Specifies the HDCP Ri interrupt frame number index register 0.	0x80
HDCP_Ri_Compare_1	0x71D4	Specifies the HDCP Ri interrupt frame number index register 1.	0x7f
HDCP_Frame_Count	0x71E0	Specifies the current value of frame count index in the hardware.	0x00
RGB_ROUND_EN	0xD500	Specifies round enable for 8/10-bit R/G/B in video_receiver	0x00
VACT_SPACE_R_0	0xD504	Specifies vertical active space R0	0x00
VACT_SPACE_R_1	0xD508	Specifies vertical active space R1	0x00
VACT_SPACE_G_0	0xD50C	Specifies vertical active space G0	0x00
VACT_SPACE_G_1	0xD510	Specifies vertical active space G1	0x00
VACT_SPACE_B_0	0xD514	Specifies vertical active space B0	0x00
VACT_SPACE_B_1	0xD518	Specifies vertical active space B1	0x00
BLUE_SCREEN_R_0	0xD520	Specifies R pixel values for blue screen[3:0]	0x00
BLUE_SCREEN_R_1	0xD524	Specifies R pixel values for blue screen[11:4]	0x00

Register	Offset	Description	Reset Value
BLUE_SCREEN_G_0	0xD528	Specifies G pixel values for blue screen[3:0]	0x00
BLUE_SCREEN_G_1	0xD52C	Specifies G pixel values for blue screen[11:4]	0x00
BLUE_SCREEN_B_0	0xD530	Specifies B pixel values for blue screen[3:0]	0x00
BLUE_SCREEN_B_1	0xD534	Specifies B pixel values for blue screen[11:4]	0x00

- Base Address: 0x12D3_0000

Register	Offset	Description	Reset Value
SPDIF Registers			
SPDIFIN_CLK_CTRL	0x0000	Specifies the SPDIFIN clock control register.	0x02
SPDIFIN_OP_CTRL	0x0004	Specifies the SPDIFIN operation control register 1.	0x00
SPDIFIN_IRQ_MASK	0x0008	Specifies the SPDIFIN interrupt request mask register.	0x00
SPDIFIN_IRQ_STATUS	0x000C	Specifies the SPDIFIN interrupt request status register.	0x00
SPDIFIN_CONFIG_1	0x0010	Specifies the SPDIFIN configuration register 1.	0x02
SPDIFIN_CONFIG_2	0x0014	Specifies the SPDIFIN configuration register 2.	0x00
SPDIFIN_USER_VALUE_1	0x0020	Specifies the SPDIFIN user value register 1.	0x00
SPDIFIN_USER_VALUE_2	0x0024	Specifies the SPDIFIN user value register 2.	0x00
SPDIFIN_USER_VALUE_3	0x0028	Specifies the SPDIFIN user value register 3.	0x00
SPDIFIN_USER_VALUE_4	0x002C	Specifies the SPDIFIN user value register 4.	0x00
SPDIFIN_CH_STATUS_0_1	0x0030	Specifies the SPDIFIN channel status register 0-1.	0x00
SPDIFIN_CH_STATUS_0_2	0x0034	Specifies the SPDIFIN channel status register 0-2.	0x00
SPDIFIN_CH_STATUS_0_3	0x0038	Specifies the SPDIFIN channel status register 0-3.	0x00
SPDIFIN_CH_STATUS_0_4	0x003C	Specifies the SPDIFIN channel status register 0-4.	0x00
SPDIFIN_CH_STATUS_1	0x0040	Specifies the SPDIFIN channel status register 1.	0x00
SPDIFIN_FRAME_PERIOD_1	0x0048	Specifies the SPDIFIN frame period register 1.	0x00
SPDIFIN_FRAME_PERIOD_2	0x004C	Specifies the SPDIFIN frame period register 2.	0x00
SPDIFIN_Pc_INFO_1	0x0050	Specifies the SPDIFIN PC info register 1.	0x00
SPDIFIN_Pc_INFO_2	0x0054	Specifies the SPDIFIN PC info register 2.	0x00
SPDIFIN_Pd_INFO_1	0x0058	Specifies the SPDIFIN PD info register 1.	0x00
SPDIFIN_Pd_INFO_2	0x005C	Specifies the SPDIFIN PD info register 2.	0x00
SPDIFIN_DATA_BUF_0_1	0x0060	Specifies the SPDIFIN data buffer register 0_1.	0x00
SPDIFIN_DATA_BUF_0_2	0x0064	Specifies the SPDIFIN data buffer register 0_2.	0x00

Register	Offset	Description	Reset Value
SPDIFIN_DATA_BUF_0_3	0x0068	Specifies the SPDIFIN data buffer register 0_3.	0x00
SPDIFIN_USER_BUF_0	0x006C	Specifies the SPDIFIN user buffer register 0.	0x00
SPDIFIN_DATA_BUF_1_1	0x0070	Specifies the SPDIFIN data buffer register 1_1.	0x00
SPDIFIN_DATA_BUF_1_2	0x0074	Specifies the SPDIFIN data buffer register 1_2.	0x00
SPDIFIN_DATA_BUF_1_3	0x0078	Specifies the SPDIFIN data buffer register 1_3.	0x00
SPDIFIN_USER_BUF_1	0x007C	Specifies the SPDIFIN user buffer register 1.	0x00

- Base Address: 0x12D4_0000

Register	Offset	Description	Reset Value
I2S Registers			
I2S_CLK_CON	0x0000	Specifies the I2S clock enable register.	0x00
I2S_CON_1	0x0004	Specifies the I2S control register 1.	0x00
I2S_CON_2	0x0008	Specifies the I2S control register 2.	0x16
I2S_PIN_SEL_0	0x000C	Specifies the I2S input pin selection register 0.	0x77
I2S_PIN_SEL_1	0x0010	Specifies the I2S input pin selection register 1.	0x77
I2S_PIN_SEL_2	0x0014	Specifies the I2S input pin selection register 2.	0x77
I2S_PIN_SEL_3	0x0018	Specifies the I2S input pin selection register 3.	0x07
I2S_DSD_CON	0x001C	Specifies the I2S DSD control register.	0x02
I2S_IN_MUX_CON	0x0020	Specifies the I2S In/mux control register.	0x60
I2S_CH_ST_CON	0x0024	Specifies the I2S channel status control register.	0x00
I2S_CH_ST_0	0x0028	Specifies the I2S channel status block 0.	0x00
I2S_CH_ST_1	0x002C	Specifies the I2S channel status block 1.	0x00
I2S_CH_ST_2	0x0030	Specifies the I2S channel status block 2.	0x00
I2S_CH_ST_3	0x0034	Specifies the I2S channel status block 3.	0x00
I2S_CH_ST_4	0x0038	Specifies the I2S channel status block 4.	0x00
I2S_CH_ST_SH_0	0x003C	Specifies the I2S channel status block shadow register 0.	0x00
I2S_CH_ST_SH_1	0x0040	Specifies the I2S channel status block shadow register 1.	0x00
I2S_CH_ST_SH_2	0x0044	Specifies the I2S channel status block shadow register 2.	0x00
I2S_CH_ST_SH_3	0x0048	Specifies the I2S channel status block shadow register 3.	0x00
I2S_CH_ST_SH_4	0x004C	Specifies the I2S channel status block shadow register 4.	0x00
I2S_VD_DATA	0x0050	Specifies the I2S audio sample validity register.	0x00
I2S_MUX_CH	0x0054	Specifies the I2S channel enable register.	0x03
I2S_MUX_CUV	0x0058	Specifies the I2S CUV enable register.	0x03

Register	Offset	Description	Reset Value
I2S_CH0_L_0	0x0064	Specifies the I2S PCM output data register.	0x00
I2S_CH0_L_1	0x0068	Specifies the I2S PCM output data register.	0x00
I2S_CH0_L_2	0x006C	Specifies the I2S PCM output data register.	0x00
I2S_CH0_L_3	0x0070	Specifies the I2S PCM output data register.	0x00
I2S_CH0_R_0	0x0074	Specifies the I2S PCM output data register.	0x00
I2S_CH0_R_1	0x0078	Specifies the I2S PCM output data register.	0x00
I2S_CH0_R_2	0x007C	Specifies the I2S PCM output data register.	0x00
I2S_CH0_R_3	0x0080	Specifies the I2S PCM output data register.	0x00
I2S_CH1_L_0	0x0084	Specifies the I2S PCM output data register.	0x00
I2S_CH1_L_1	0x0088	Specifies the I2S PCM output data register.	0x00
I2S_CH1_L_2	0x008C	Specifies the I2S PCM output data register.	0x00
I2S_CH1_L_3	0x0090	Specifies the I2S PCM output data register.	0x00
I2S_CH1_R_0	0x0094	Specifies the I2S PCM output data register.	0x00
I2S_CH1_R_1	0x0098	Specifies the I2S PCM output data register.	0x00
I2S_CH1_R_2	0x009C	Specifies the I2S PCM output data register.	0x00
I2S_CH1_R_3	0x00A0	Specifies the I2S PCM output data register.	0x00
I2S_CH2_L_0	0x00A4	Specifies the I2S PCM output data register.	0x00
I2S_CH2_L_1	0x00A8	Specifies the I2S PCM output data register.	0x00
I2S_CH2_L_2	0x00AC	Specifies the I2S PCM output data register.	0x00
I2S_CH2_L_3	0x00B0	Specifies the I2S PCM output data register.	0x00
I2S_CH2_R_0	0x00B4	Specifies the I2S PCM output data register.	0x00
I2S_CH2_R_1	0x00B8	Specifies the I2S PCM output data register.	0x00
I2S_CH2_R_2	0x00BC	Specifies the I2S PCM output data register.	0x00
I2S_Ch2_R_3	0x00C0	Specifies the I2S PCM output data register.	0x00
I2S_CH3_L_0	0x00C4	Specifies the I2S PCM output data register.	0x00
I2S_CH3_L_1	0x00C8	Specifies the I2S PCM output data register.	0x00
I2S_CH3_L_2	0x00CC	Specifies the I2S PCM output data register.	0x00
I2S_CH3_R_0	0x00D0	Specifies the I2S PCM output data register.	0x00
I2S_CH3_R_1	0x00D4	Specifies the I2S PCM output data register.	0x00
I2S_CH3_R_2	0x00D8	Specifies the I2S PCM output data register.	0x00
I2S_CUV_L_R	0x00DC	Specifies the I2S CUV output data register.	0x00

- Base Address: 0x12D5_0000

Register	Offset	Description	Reset Value
Timing Generator Registers			
TG Configure/Status Registers			
TG_CMD	0x0000	Specifies the command register.	0x00
TG_H_FSZ_L	0x0018	Specifies the horizontal full size.	0x72
TG_H_FSZ_H	0x001C	Specifies the horizontal full size.	0x06
TG_HACT_ST_L	0x0020	Specifies the horizontal active start.	0x05
TG_HACT_ST_H	0x0024	Specifies the horizontal active start.	0x01
TG_HACT_SZ_L	0x0028	Specifies the horizontal active size.	0x00
TG_HACT_SZ_H	0x002C	Specifies the horizontal active size.	0x05
TG_V_FSZ_L	0x0030	Specifies the vertical full line size.	0xEE
TG_V_FSZ_H	0x0034	Specifies the vertical full line size.	0x02
TG_VSYNC_L	0x0038	Specifies the vertical sync position.	0x01
TG_VSYNC_H	0x003C	Specifies the vertical sync position.	0x00
TG_VSYNC2_L	0x0040	Specifies the vertical sync position for bottom field.	0x33
TG_VSYNC2_H	0x0044	Specifies the vertical sync position for bottom field.	0x02
TG_VACT_ST_L	0x0048	Specifies the vertical sync active start position.	0x1a
TG_VACT_ST_H	0x004C	Specifies the vertical sync active start position.	0x00
TG_VACT_SZ_L	0x0050	Specifies the vertical active size.	0xd0
TG_VACT_SZ_H	0x0054	Specifies the vertical active size.	0x02
TG_FIELD_CHG_L	0x0058	Specifies the HDMI field change position.	0x33
TG_FIELD_CHG_H	0x005C	Specifies the HDMI field change position.	0x02
TG_VACT_ST2_L	0x0060	Specifies the HDMI vertical active start position for bottom field.	0x48
TG_VACT_ST2_H	0x0064	Specifies the HDMI vertical active start position for bottom field.	0x02
TG_VACT_ST3_L	0x0068	Specifies the HDMI vertical active start position for third bottom field.	0x7B
TG_VACT_ST3_H	0x006C	Specifies the HDMI vertical active start position for third bottom field.	0x04
TG_VACT_ST4_L	0x0070	Specifies the HDMI vertical active start position for fourth bottom field.	0xAE
TG_VACT_ST4_H	0x0074	Specifies the HDMI vertical active start position for fourth bottom field.	0x06
TG_VSYNC_TOP_HDMI_L	0x0078	Specifies the HDMI VSYNC position for top field.	0x01
TG_VSYNC_TOP_HDMI_H	0x007C	Specifies the HDMI VSYNC position for top field.	0x00
TG_VSYNC_BOT_HDMI_L	0x0080	Specifies the HDMI VSYNC position for bottom	0x33

Register	Offset	Description	Reset Value
		field.	
TG_VSYNC_BOT_HDMI_H	0x0084	Specifies the HDMI VSYNC position for bottom field.	0x02
TG_FIELD_TOP_HDMI_L	0x0088	Specifies the HDMI top field start position.	0x01
TG_FIELD_TOP_HDMI_H	0x008C	Specifies the HDMI top field start position.	0x00
TG_FIELD_BOT_HDMI_L	0x0090	Specifies the HDMI bottom field start position.	0X33
TG_FIELD_BOT_HDMI_H	0x0094	Specifies the HDMI bottom field start position.	0x02
TG_3D	0x00F0	Specifies the stereoscopy timing enable	0x00

- Base Address: 0x12D6_0000

Register	Offset	Description	Reset Value
HDCP eFUSE Registers			
HDCP_E_FUSE_CTRL	0x0000	E_FUSE control register	0x00
HDCP_E_FUSE_STATUS	0x0004	E_FUSE status register	0x00
EFUSE_ADDR_WIDTH	0x0008	Specifies the address width	0x14
EFUSE_SIGDEV_ASSERT	0x000C	Specifies the SIGDEV asserting position	0x00
EFUSE_SIGDEV_DE_ASSERT	0x0010	Specifies the SIGDEV de-asserting position	0x08
EFUSE_PRCHG_ASSERT	0x0014	Specifies the PRCHG asserting position	0x00
EFUSE_PRCHG_DE_ASSERT	0x0018	Specifies the PRCHG de-asserting position	0x0C
EFUSE_FSET_ASSERT	0x001C	Specifies the FSET asserting position	0x04
EFUSE_FSET_DE_ASSERT	0x0020	Specifies the FSET de-asserting position	0x10
EFUSE_SENSING	0x0024	Specifies the sensing width	0x14
EFUSE_SCK_ASSERT	0x0028	Specifies the SCK asserting position	0x04
EFUSE_SCK_DE_ASSERT	0x002C	Specifies the SCK de-asserting position	0x0C
EFUSE_SDOUT_OFFSET	0x0030	Specifies the SDOUT offset	0x10
EFUSE_READ_OFFSET	0x0034	Specifies the READ Offset	0x14

- Base Address: 0x100B_0000

Register	Offset	Description	Reset Value
CEC Configure Registers			
CEC_TX_STATUS_0	0x0000	CEC Tx status register 0.	0x00
CEC_TX_STATUS_1	0x0004	CEC Tx status register 1. Number of blocks transferred.	0x00
CEC_RX_STATUS_0	0x0008	CEC Rx status register 0.	0x00
CEC_RX_STATUS_1	0x000C	CEC Rx status register 1. Number of blocks received.	0x00
CEC_INTR_MASK	0x0010	CEC interrupt mask register	0x00

Register	Offset	Description	Reset Value
CEC_INTR_CLEAR	0x0014	CEC interrupt clear register	0x00
CEC_LOGIC_ADDR	0x0020	HDMI Tx logical address register	0x0F
CEC_DIVISOR_0	0x0030	Clock divisor for 0.05 ms period count ([7:0] of 32-bit)	0x00
CEC_DIVISOR_1	0x0034	Clock divisor for 0.05 ms period count ([15:8] of 32-bit)	0x00
CEC_DIVISOR_2	0x0038	Clock divisor for 0.05 ms period count ([23:16] of 32-bit)	0x00
CEC_DIVISOR_3	0x003C	Clock divisor for 0.05 ms period count ([31:24] of 32-bit)	0x00
CEC_TX_CTRL	0x0040	CEC Tx control register	0x10
CEC_TX_BYTE_NUM	0x0044	Number of blocks in a message to be transferred	0x00
CEC_TX_STATUS_2	0x0060	CEC Tx status register 2	0x00
CEC_TX_STATUS_3	0x0064	CEC Tx status register 3	0x00
CEC_TX_BUFFER_X	0x0080 to 0x00BC	Byte #0 to #15 of CEC message to be transferred. (#0 is transferred 1st)	0x00
CEC_RX_CTRL	0x00C0	CEC Rx control register	0x00
CEC_RX_STATUS_2	0x00E0	CEC Rx status register 2	0x00
CEC_RX_STATUS_3	0x00E4	CEC Rx status register 3	0x00
CEC_RX_BUFFER_x	0x0100 to 0x013C	Byte #0 to #15 of CEC message received. (#0 is received 1st)	0x00
CEC_FILTER_CTRL	0x0180	CEC Filter control register	0x81
CEC_FILTER_TH	0x0184	CEC Filter Threshold register	0x03

53.10.2 Control Registers

53.10.2.1 INTC_CON_0

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
IntrPol	[7]	RW	Specifies the interrupt polarity. 0 = Active high 1 = Active low	0
IntrEnGlobal	[6]	RW	0 = Disables all interrupts 1 = Enables or disables interrupts by INTC_CON[5:0]	0
IntrEnI2S	[5]	RW	Enables I2S interrupt. 0 = Disables I2S interrupt 1 = Enables I2S interrupt	0
IntrEnCEC	[4]	RW	Enables CEC interrupt. 0 = Disables CEC interrupt 1 = Enables CEC interrupt	0
IntrEnHPDplug	[3]	RW	Enables HPD plugged interrupt. 0 = Disables HPD plugged interrupt 1 = Enables HPD plugged interrupt	0
IntrEnHPDunplug	[2]	RW	Enables HPD unplugged interrupt. 0 = Disables HPD unplugged interrupt 1 = Enables HPD unplugged interrupt	0
IntrEnSPDIF	[1]	RW	Enables SPDIF interrupt. 0 = Disables SPDIF interrupt 1 = Enables SPDIF interrupt	0
IntrEnHDCP	[0]	RW	Enables HDCP interrupt. 0 = Disables HDCP interrupt 1 = Enables HDCP interrupt	0

53.10.2.2 INTC_FLAG_0

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	2b00
IntrI2S	[5]	RW	Specifies the I2S interrupt flag (read only). 0 = Interrupt does not occur 1 = Interrupt occurs	0
IntrCEC	[4]	RW	Specifies the CEC interrupt flag (read only). 0 = Interrupt does not occur 1 = Interrupt occurs	0
IntrHPDplug	[3]	RW	Specifies the HPD plugged interrupt flag. It clears when you write 1. 0 = Interrupt does not occur 1 = HPD plugged interrupt occurs	0
IntrHPDunplug	[2]	RW	Specifies the HPD unplugged interrupt flag. It clears when you write 1. 0 = Interrupt does not occur 1 = HPD unplugged interrupt occurs	0
IntrSPDIF	[1]	RW	Specifies the SPDIF interrupt flag (read only). 0 = Interrupt does not occur 1 = Interrupt occurs	0
IntrHDCP	[0]	RW	Specifies the HDCP interrupt flag (read only). 0 = Interrupt does not occur 1 = Interrupt occurs	0

53.10.2.3 HDCP_KEY_LOAD

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0008, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	R	Reserved	7b0000000
HDCP_KEY_LOAD_DONE	[0]	R	Loads the HDCP key from e-fuse. 0 = Not available 1 = Completes loading HDCP key from e-fuse	0

53.10.2.4 HPD_STATUS

- Base Address: 0x12D0_0000
- Address = Base Address + 0x000C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	R	Reserved	0x00
HPD_Value	[0]	R	Specifies the value of HPD signal. 0 = Unplugged 1 = Plugged	0

53.10.2.5 INTC_CON_1

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0010, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	0x00
IntEnSinkDetect	[1]	RW	SINK_DET interrupt enable. Triggers when SINK_DET signal from goes PHY high. INTC_CON_1[6] should also be enabled. 0 = Disables 1 = Enables	0
IntEnSinknotDetect	[0]	RW	SINK_NOT_DET interrupt enable. Triggers when SINK_DET signal from PHY goes low. INTC_CON_1[6] should also be enabled. 0 = Disables 1 = Enables	0

53.10.2.6 INTC_FLAG_1

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0014, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	0x00
IntSinkDetect	[1]	RW	SINK_DET interrupt. Triggers when SINK_DET signal from PHY goes high. INTC_CON_1[6] should also be enabled. It clears when you Write 1. 0 = Does not occur 1 = SINK_DET positive edge occurs.	0
IntSinknotDetect	[0]	RW	SINK_NOT_DET interrupt. Triggers when SINK_DET signal from PHY goes low. INTC_CON_1[6] should also be enabled. It clears when you Write 1. 0 = Does not occur 1 = SINK_DET negative edge occurs.	0

53.10.2.7 PHY_STATUS_0

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0020, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	R	Reserved	0x00
Sink_Detect	[1]	R	SINK_DET signal from PHY. 0 = Does not detect Sink 1 = Detects Sink	0
Phy_Ready	[0]	R	PHY_READY signal from PHY. 0 = PHY not ready 1 = PHY ready	0

53.10.2.8 PHY_STATUS_AFC

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0024, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AFC_Code	[7:4]	R	AFC_CODE signal from PHY. AFC_CODE is AFC (automatic frequency calibration) code that is used by the CMU to converge to the target frequency. To lock CMU, PLL that generated TMDS clock and the pixel clock generator should be locked.	0x00
RSVD	[3:0]	R	Reserved	0

53.10.2.9 PHY_STATUS_PLL

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0028, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	R	Reserved	0
PLL_Lock	[0]	R	PLL_LOCK signal from PHY. 0 = Does not lock PLL 1 = Locks PLL	0

53.10.2.10 PHY_CON_0

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0030, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0
PHY_Pwr_Off	[0]	RW	PHY power off signal. It propagates value of this bit to o_phy_pwroff port of hdmi_14tx_ss. When it connects to PHY properly, you can power-off PHY. Refer to PHY datasheet for more information.	0

53.10.2.11 HPD_CTRL

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0040, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0
HPD_Deglitch_En	[0]	RW	Enables deglitch logic to wait for a stable HPD signal. The duration of stable signal is determined by HPD_TH0 to HPD_TH3 registers.	0

53.10.2.12 HPD_ST

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0044, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0
HPD_Deglitched	[0]	RW	Current HPD signal status after deglitch logic.	0

53.10.2.13 HPD_TH_X (0 to 3)

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0050, + 0x0054, + 0x0058, + 0x005C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HPD_TH_x	[7:0]	RW	A 32-bit HPD filter threshold value. When filter is enabled, it filters out signals stable for less than HPD_Th cycles. Least significant byte first. For example, • HPD_Th[7:0] ← HPD_TH_0[7:0] • HPD_Th[31:24] ← HPD_TH_3[7:0]	0

53.10.2.14 AUDIO_CLKSEL

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0070, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0x00
AUDIO_CLK	[0]	RW	Specifies the clock selection of Audio system (Must be higher than 512 × fs). 0 = PCLK 1 = SPDIF clock	0

53.10.2.15 HDMI_PHY_RSTOUT

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0074, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0x00
RSTOUT	[0]	RW	Specifies the HDMI PHY Software Reset out (active high). 0 = Normal 1 = Reset	0

53.10.2.16 HDMI_PHY_PLL

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0078, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
PLL_LOCK	[7]	R	Specifies the HDMI PHY PLL Locking.	0x0
RSVD	[6:0]	R	Reserved	0x0

53.10.2.17 HDMI_PHY_AFC

- Base Address: 0x12D0_0000
- Address = Base Address + 0x007C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	R	Reserved	0x0
AFC_CODE	[3:0]	R	Specifies the HDMI PHY AFC Code.	0x0

53.10.2.18 HDMI_CORE_RSTOUT

- Base Address: 0x12D0_0000
- Address = Base Address + 0x0080, Reset Value = 0x01

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0x00
RSTOUT	[0]	RW	Specifies the HDMI TX core software reset out (active low). 1 = Normal 0 = Reset	0x1

53.10.3 HDMI Core Registers (Control Registers)

53.10.3.1 HDMI_CON_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	0
Blue_Scr_En	[5]	RW	Enables blue screen mode. When set, the input video pixels are discarded and blue screen register values are transmitted for all video data period. 0 = Disables screen mode 1 = Enables screen mode	0
Encoding_Option	[4]	RW	Specifies 10-bit TMDS encoding bit order option. 0 = Reverses bit order during 10-bit encoding (to be set to 1 when connecting to TMDS PHY) 1 = Retains bit order as it is	0
RSVD	[3]	RW	Reserved	0
Asp_E	[2]	RW	Generates audio sample packet. This bit is valid only when SYSTEM_EN is set. 0 = Discards audio sample 1 = Generates audio sample packet after receiving audio sample	0
RSVD	[1]	RW	Reserved	0
System_En	[0]	RW	Enables HDMI system. 0 = Disables HDMI 1 = Enables HDMI	0

53.10.3.2 HDMI_CON_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	0
Pxl_Lmt_Ctrl	[6:5]	RW	Controls the pixel value limitation. • 2b00 = By-pass (Does not limit the pixel value) • 2b01 = RGB mode It limits video input pixels of every channel on the basis of YMAX and YMIN register values. • 2b10 = YCbCr mode Limits the value of I_VIDEO_G on the basis of YMAX and YMIN register values. Limits the values of I_VIDEO_B and I_VIDEO_R on the basis of CMAX and CMIN register values. • 2b11 = Reserved	3b00
RSVD	[4:2]	RW	Reserved	3b000
RSVD	[1:0]	RW	Reserved	0

53.10.3.3 HDMI_CON_2

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0008, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	3b00
Vid_Period_En	[5]	RW	Controls the video preamble. 0 = Applies Video preamble (HDMI mode) 1 = Does not apply Video preamble (DVI mode)	0
RSVD	[4:2]	RW	Reserved	3b000
Dvi_ Band_En	[1]	RW	In DIV mode, the leading guard band is not used. 0 = Applies Guard band (HDMI mode) 1 = Does not apply Guard band (DVI mode)	0
RSVD	[0]	RW	Reserved	0

53.10.3.4 STATUS

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0010, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Authen_Ack	[7]	RW	When HDCP is authenticated, this read-only bit occurs. It keeps the authentication signal without interruption. It is not cleared at all. This bit specifies just one delayed signal of authen_ack from HDCP block. It is not an interrupt source. 0 = Does not authenticate 1 = Authenticates	0
Aud_Fifo_Ovf	[6]	RW	When audio FIFO overflows, this bit is set. After it is set, host should clear it. 0 = Not full 1 = Full	0
RSVD	[5]	RW	Reserved	0
Update_Ri_Int	[4]	RW	Specifies the Ri interrupt status bit. If you write it as 1, it clears. 0 = Interrupt does not occur 1 = Interrupt occurs	0
RSVD	[3]	RW	Reserved	0
An_Write_Int	[2]	RW	Indicates that {An} random value is ready. If you write it as 1, it clears. 0 = Interrupt does not occur 1 = Interrupt occurs	0
Watchdog_Int	[1]	RW	Indicates that the second part of HDCP authentication protocol is initiated. CPU should set a watchdog timer to check 5 seconds interval. If it is written by 1, it is cleared. 0 = Interrupt does not occur 1 = Interrupt occurs	0
I2c_Init_Int	[0]	RW	Indicates that the first part of HDCP authentication protocol can start. If it is written by 1, it is cleared. 0 = Interrupt does not occur 1 = Interrupt occurs	0

53.10.3.5 STATUS_EN

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0020, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	0
Aud_Fido_Ovf_Ee	[6]	RW	Enables audio buffer overflow interrupt. When it is set to "1", it writes interrupt assertion on status registers. 0 = Disables interrupt 1 = Enables interrupt	0
RSVD	[5]	RW	Reserved	0
Update_Ri_Int_En	[4]	RW	Enables UPDATE_RI_INT interrupt. 0 = Disables interrupt 1 = Enables interrupt	0
RSVD	[3]	RW	Reserved	0
An_Write_Int_En	[2]	RW	Enables AN_WRITE_INT interrupt. 0 = Disables interrupt 1 = Enables interrupt	0
Watchdog_Int_En	[1]	RW	Enables WATCHDOG_INT interrupt. 0 = Disables interrupt 1 = Enables interrupt	0
I2c_Int_En	[0]	RW	Enables I2C_INT interrupt. 0 = Disables 1 = Enables	0

53.10.3.6 HPD

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0030, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
Sw_Hpd	[1]	RW	If HPD_SEL bit is set, it uses SW_HPD signal for HPD (HDMI/ DVI cable plugging). However, when this bit is set to low during HDMI transmission, status machines in HDCP core are reset. Note that it does not influence other HDCP register values. 0 = Low (unplugged) 1 = High (plugged)	0
Hpd_Sel	[0]	RW	It this bit is cleared, the I_HPD signal from the I/O port is used for HPD. If set, the SW_HPD signal is used for HPD. 0 = HPD signal 1 = SW_HPD internal HPD signal	0

NOTE: If you disable (not using HDCP) ENC_EN (0x12D1_0044) then S/W controls HPD. If you do not use S/W control, it is possible that HDMI core works abnormally.

53.10.3.7 MODE_SEL

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0040, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
Hdmi_Mode	[1]	RW	Selects a mode. 0 = Disables mode. 1 = Enables mode.	0
Dvi_Mode	[0]	RW	Selects a mode. 0 = Disables mode. 1 = Enables mode.	0

NOTE: DVI mode gets a higher priority than HDMI.

53.10.3.8 ENC_EN

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0044, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	7b0000000
Hdcp_Enc_En	[0]	RW	When this bit is set, it applies HDCP encryption. Before setting this bit, the HDCP authentication process should complete. 0 = Disables encryption 1 = Enables encryption	0

53.10.4 HDMI Core Registers (Video Related Registers)

53.10.4.1 HDMI_YMAX/HDMI_YMIN/HDMI_CMAX/HDMI_CMIN

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0060, Reset Value = 0xEB
- Address = Base Address + 0x0064, Reset Value = 0x10
- Address = Base Address + 0x0068, Reset Value = 0xF0
- Address = Base Address + 0x006C, Reset Value = 0x10

Name	Bit	Type	Description	Reset Value

HDMI_YMAX	[7:0]	RW	It uses these registers on the basis of PX_LMT_CTRL bits in HDMI_CON_1 register.
			For RGB mode
			if (i_video_x > HDMI_YMAX × 16)
			output = HDMI_YMAX x 16
			else if (i_video_x < HDMI_YMIN × 16)
			output = HDMI_YMIN × 16
			else output = i_video_x
			For YCbCr mode, it deals Y input in similar way as shown above.
			For Cb and Cr values,
			if (i_video_x > HDMI_CMAX × 16)
			output = HDMI_CMAX × 16
			else if (i_video_x < HDMI_CMIN × 16)
			output = HDMI_CMIN × 16
			else output = i_video_x
			NOTE: The value 16 in each line compensates the difference of bit width between input pixel and register value.
				0xEB

HDMI_YMIN				0x10
HDMI_CMAX				0xF0
HDMI_CMIN				0x10

53.10.4.2 H_BLANK_0/1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x00A0, + 0x00A4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[15:13]	RW	Reserved	6b000000
H_BLANK	[12:0]	RW	Specifies the clock cycles of horizontal blanking size. Refer to "Reference CEA-861D" for more details on H_BLANK.	0x000

60 Hz	720 × 480i	720 × 480p	1440 × 480p	1280 × 720p	1920 × 1080i	1920 × 1080p
H_BLANK	276 (114h)	138 (8Ah)	276 (114h)	370 (172h)	280 (118h)	280 (118h)

50 Hz	720 × 576i	720 × 576p	1440 × 576p	1280 × 720p	1920 × 1080i	1920 × 1080p
H_BLANK	288 (120h)	144 (90h)	288 (120h)	700 (2bch)	720 (2d0h)	720 (2d0h)

NOTE: 1440 × 480p and 1440 × 576p specifies pixel-doubling format of 720 × 480p and 720 × 576p.

53.10.4.3 V2_BLANK_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x00B0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
V2_BLANK	[7:0]	RW	V2_BLANK[7:0] of 13-bit. V1_BLANK + Active Lines. End Part. This value is same as V_LINE value for progressive mode. For interlace mode, use reference value in table. Refer to Reference CEA-861D for more information.	0

53.10.4.4 V2_BLANK_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x00B4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	3b000
V2_BLANK	[4:0]	RW	V2_BLANK[12:8] of 13-bit. V1_BLANK + Active Lines. End Part. This value is same as V_LINE value for progressive mode. For interlace mode, use reference value in table. Refer to Reference CEA-861D for more information.	0

53.10.4.5 V1_BLANK_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x00B8, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
V1_BLANK	[7:0]	RW	V1_BLANK[7:0] of 13-bit. Vertical Blanking Line Size. Front Part. Refer to Reference CEA-861D for more information.	0

53.10.4.6 V1_BLANK_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x00BC, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	3b000
V1_BLANK	[4:0]	RW	V1_BLANK[7:0] of 13-bit. Vertical Blanking Line Size. Front Part. Refer to Reference CEA-861D for more information.	0

53.10.4.7 V_LINE_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x00C0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
V_LINE	[7:0]	RW	V_LINE[7:0] of 13-bit. Vertical Line Length. Refer to Reference CEA-861D for more information.	0

53.10.4.8 V_LINE_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x00C4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x0
V_LINE	[4:0]	RW	V_LINE[12:8] of 13-bit. Vertical Line Length. Refer to Reference CEA-861D for more information.	0

53.10.4.9 H_LINE_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x00C8, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
H_LINE	[7:0]	RW	H_LINE[7:0] of 13-bit. Horizontal Line Length. Refer to Reference CEA-861D for more information.	0

53.10.4.10 H_LINE_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x00CC, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x0
H_LINE	[4:0]	RW	H_LINE[12:8] of 13-bit. Horizontal Line Length. Refer to Reference CEA-861D for more information.	0

53.10.4.11 HSYNC_POL

- Base Address: 0x12D1_0000
- Address = Base Address + 0x00E0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0x00
Hsync_Pol	[0]	RW	Set this bit for inverting the generated signal to meet the modes. In 720p and 1080i modes, do not invert the signal. Others need to be inverted. Refer to Reference CEA-861D for more information. 0 = Active high 1 = Active low	0

53.10.4.12 VSYNC_POL

- Base Address: 0x12D1_0000
- Address = Base Address + 0x00E4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0x00
V_Sync_Pol_Sel	[0]	RW	Starts point detection polarity selection bit. 720p or 1080i's sync shapes are different from 480p, 480i, and 576p's. They are inverted shapes. 0 = Active high 1 = Active low	0

50/60 Hz	720×480i	720×576i	720×480p	720×576p	1280×720p	1920×1080i	1920×1080p
VSYNC_POL	1	1	1	1	0	0	0

53.10.4.13 INT_PRO_MODE

- Base Address: 0x12D1_0000
- Address = Base Address + 0x00E8, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	7b0000000
INT_PRO_MODE	[0]	RW	Selects the interlaced or progressive mode. Refer to "Reference CEA-861D" for more information. 0 = Progressive mode 1 = Interlaced mode	0

53.10.4.14 V_BLANK_F0_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0110, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_blank_f0	[7:0]	RW	v_blank_f0[7:0] of 13-bit. The start position of bottom field active region. This value is same as V_LINE value for Interlace mode. For progressive mode, This value is not used. Refer to Reference CEA-861D for more information.	0

53.10.4.15 V_BLANK_F0_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0114, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_blank_f0	[4:0]	RW	v_blank_f0[12:8] of 13-bit. The start position of bottom field's active region. This value is same as V_LINE value for Interlace mode. For progressive mode, This value is not used. Refer to Reference CEA-861D for more information.	0

53.10.4.16 V_BLANK_F1_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0118, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_blank_f1	[7:0]	RW	v_blank_f1[7:0] of 13-bit. In interlace mode, v_blank length of even field and odd field is different. This register specifies end position of active region of bottom field. Refer to Reference CEA-861D for more information.	0

53.10.4.17 V_BLANK_F1_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x011C, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_blank_f1	[4:0]	RW	v_blank_f1[12:8] of 13-bit. In the interlace mode, v_blank length of even field and odd field is different. This register specifies end position of active region of bottom field. Refer to Reference CEA-861D for more information.	0

53.10.4.18 H_SYNC_START_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0120, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Hsync_Start	[7:0]	RW	Hsync_Start[7:0] of 11-bit. Sets the start point of H sync. Refer to the Reference CEA-861D for more information.	0

53.10.4.19 H_SYNC_START_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0124, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x00
Hsync_Start	[2:0]	RW	Hsync_Start[10:8] of 11-bit. Sets start point of H sync. Refer to Reference CEA-861D for more information.	0

53.10.4.20 H_SYNC_END_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0128, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Hsync_End	[7:0]	RW	Hsync_End[7:0] of 11-bit. Sets end point of H sync. Refer to Reference CEA-861D for more information.	0

53.10.4.21 H_SYNC_END_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x012C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x00
Hsync_End	[2:0]	RW	Hsync_End[10:8] of 11-bit. Sets end point of H sync. Refer to Reference CEA-861D for more information.	0

53.10.4.22 V_SYNC_LINE_BEF_2_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0130, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_bef_2	[7:0]	RW	v_sync_line_bef_2[7:0] of 13-bit. Top field (or frame) V sync end line number. Refer to Reference CEA-861D for more information.	0xFF

53.10.4.23 V_SYNC_LINE_BEF_2_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0134, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_bef_2	[4:0]	RW	v_sync_line_bef_2[12:8] of 13-bit. Top field (or frame) V sync end line number. Refer to Reference CEA-861D for more information.	0x1F

53.10.4.24 V_SYNC_LINE_BEF_1_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0138, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_bef_1	[7:0]	RW	Top field (or frame) V sync starts line number. Refer to Reference CEA-861D for more information.	0xFF

53.10.4.25 V_SYNC_LINE_BEF_1_1

- Base Address: 0x12D1_0000
- Address = Base Address +0x013C, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_bef_1	[4:0]	RW	v_sync_line_bef_1[12:8] of 13-bit. Top field (or frame) V sync starts line number. Refer to Reference CEA-861D for more information.	0x1F

53.10.4.26 V_SYNC_LINE_AFT_2_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0140, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_2	[7:0]	RW	v_sync_line_aft_2[7:0] of 13-bit. Bottom field V sync end line number. Refer to Reference CEA-861D for more information.	0xFF

53.10.4.27 V_SYNC_LINE_AFT_2_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0144, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_aft_2	[4:0]	RW	v_sync_line_aft_2[12:8] of 13-bit. Bottom field V sync end line number. Refer to Reference CEA-861D for more information.	0x1F

53.10.4.28 V_SYNC_LINE_AFT_1_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0148, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_1	[7:0]	RW	v_sync_line_aft_1[7:0] of 13-bit. Bottom field V sync start line number. Refer to Reference CEA-861D for more information.	0xFF

53.10.4.29 V_SYNC_LINE_AFT_1_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x014C, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_aft_1	[4:0]	RW	v_sync_line_aft_1[12:8] of 13-bit. Bottom field V sync start line number. Refer to Reference CEA-861D for more information.	0x1F

53.10.4.30 V_SYNC_LINE_AFT_PXL_2_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0150, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_pxl_2	[7:0]	RW	v_sync_line_aft_pxl_2[7:0] of 13-bit. Bottom field V sync end transition point. Refer to Reference CEA-861D for more information.	0xFF

53.10.4.31 V_SYNC_LINE_AFT_PXL_2_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0154, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_aft_pxl_2	[4:0]	RW	v_sync_line_aft_pxl_2[12:8] of 13-bit. Bottom field V sync end transition point. Refer to Reference CEA-861D for more information.	0x1F

53.10.4.32 V_SYNC_LINE_AFT_PXL_1_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0158, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_pxl_1	[7:0]	RW	v_sync_line_aft_pxl_1[7:0] of 13-bit. Bottom field V sync start transition point. Refer to Reference CEA-861D for more information.	0xFF

53.10.4.33 V_SYNC_LINE_AFT_PXL_1_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x015C, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_aft_pxl_1	[4:0]	RW	v_sync_line_aft_pxl_1[12:8] of 13-bit. Bottom field V sync start transition point. Refer to Reference CEA-861D for more information.	0x1F

53.10.4.34 V_BLANK_F2_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0160, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_blank_f2	[7:0]	RW	v_blank_f2[7:0] of 13-bit. The start position of third field's active region. For 2D mode, This value is not used.	0xFF

53.10.4.35 V_BLANK_F2_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0164, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_blank_f2	[4:0]	RW	v_blank_f2[12:8] of 13-bit. The start position of third field's active region. For 2D mode, This value is not used.	0x1F

53.10.4.36 V_BLANK_F3_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0168, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_blank_f3	[7:0]	RW	v_blank_f3[7:0] of 13-bit. The end position of third field's active region. For 2D mode, This value is not used.	0xFF

53.10.4.37 V_BLANK_F3_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x016C, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_blank_f3	[4:0]	RW	v_blank_f3[12:8] of 13-bit. The end position of third field's active region. For 2D mode, This value is not used.	0x1F

53.10.4.38 V_BLANK_F4_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0170, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_blank_f4	[7:0]	RW	v_blank_f4[7:0] of 13-bit. The start position of fourth field's active region. For 2D mode, This value is not used.	0xFF

53.10.4.39 V_BLANK_F4_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0174, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_blank_f4	[4:0]	RW	v_blank_f4[12:8] of 13-bit. The start position of fourth field's active region. For 2D mode, This value is not used.	0x1F

53.10.4.40 V_BLANK_F5_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0178, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_blank_f5	[7:0]	RW	v_blank_f5[7:0] of 13-bit. The end position of fourth field's active region. For 2D mode, This value is not used.	0xFF

53.10.4.41 V_BLANK_F5_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x017C, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_blank_f5	[4:0]	RW	v_blank_f5[12:8] of 13-bit. The end position of fourth field's active region. For 2D mode, This value is not used.	0x1F

53.10.4.42 V_SYNC_LINE_AFT_3_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0180, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_3	[7:0]	RW	v_sync_line_aft_3[7:0] of 13-bit. Third field V sync starts line number.	0xFF

53.10.4.43 V_SYNC_LINE_AFT_3_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0184, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_aft_3	[4:0]	RW	v_sync_line_aft_3[12:8] of 13-bit. Third field V sync starts line number.	0x1F

53.10.4.44 V_SYNC_LINE_AFT_4_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0188, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_4	[7:0]	RW	v_sync_line_aft_4[7:0] of 13-bit. Third field V sync end line number.	0xFF

53.10.4.45 V_SYNC_LINE_AFT_4_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x018C, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_aft_4	[4:0]	RW	v_sync_line_aft_4[12:8] of 13-bit. Third field V sync end line number.	0x1F

53.10.4.46 V_SYNC_LINE_AFT_5_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x190, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_5	[7:0]	RW	v_sync_line_aft_5[7:0] of 13-bit. Fourth field V sync starts line number.	0xFF

53.10.4.47 V_SYNC_LINE_AFT_5_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0194, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_aft_5	[4:0]	RW	v_sync_line_aft_5[12:8] of 13-bit. Fourth field V sync starts line number.	0x1F

53.10.4.48 V_SYNC_LINE_AFT_6_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0198, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_6	[7:0]	RW	v_sync_line_aft_6[7:0] of 13-bit. Fourth field V sync end line number.	0xFF

53.10.4.49 V_SYNC_LINE_AFT_6_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x019C, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_aft_6	[4:0]	RW	v_sync_line_aft_6[12:8] of 13-bit. Fourth field V sync end line number.	0x1F

53.10.4.50 V_SYNC_LINE_AFT_PXL_3_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x1A0, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_pxl_3	[7:0]	RW	v_sync_line_aft_pxl_3[7:0] of 13-bit. Third field V sync start transition point.	0xFF

53.10.4.51 V_SYNC_LINE_AFT_PXL_3_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01A4, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_aft_pxl_3	[4:0]	RW	v_sync_line_aft_pxl_3[12:8] of 13-bit. Third field V sync start transition point.	0x1F

53.10.4.52 V_SYNC_LINE_AFT_PXL_4_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01A8, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_pxl_4	[7:0]	RW	v_sync_line_aft_pxl_4[7:0] of 13-bit. Third field V sync end transition point.	0xFF

53.10.4.53 V_SYNC_LINE_AFT_PXL_4_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01AC, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_aft_pxl_4	[4:0]	RW	v_sync_line_aft_pxl_4[12:8] of 13-bit. Third field V sync end transition point.	0x1F

53.10.4.54 V_SYNC_LINE_AFT_PXL_5_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01B0, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_pxl_5	[7:0]	RW	v_sync_line_aft_pxl_5[7:0] of 13-bit. Fourth field V sync start transition point.	0xFF

53.10.4.55 V_SYNC_LINE_AFT_PXL_5_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01B4, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_aft_pxl_5	[4:0]	RW	v_sync_line_aft_pxl_5[12:8] of 13-bit. Fourth field V sync start transition point.	0x1F

53.10.4.56 V_SYNC_LINE_AFT_PXL_6_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01B8, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_pxl_6	[7:0]	RW	v_sync_line_aft_pxl_6[7:0] of 13-bit. Fourth field V sync end transition point.	0xFF

53.10.4.57 V_SYNC_LINE_AFT_PXL_6_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01BC, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
v_sync_line_aft_pxl_6	[4:0]	RW	v_sync_line_aft_pxl_6[12:8] of 13-bit. Fourth field V sync end transition point.	0x1F

53.10.4.58 VACT_SPACE_1_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0xC0, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
vact_space1	[7:0]	RW	vact_space1[7:0] of 13-bit. First active space start line number	0xFF

53.10.4.59 VACT_SPACE_1_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0xC4, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
vact_space1	[4:0]	RW	vact_space1[12:8] of 13-bit. First active space start line number	0x1F

53.10.4.60 VACT_SPACE_2_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0xC8, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
vact_space2	[7:0]	RW	vact_space2[7:0] of 13-bit. First active space end line number	0xFF

53.10.4.61 VACT_SPACE_2_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0xCC, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
vact_space2	[4:0]	RW	vact_space2[12:8] of 13-bit. First active space end line number	0x1F

53.10.4.62 VACT_SPACE_3_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01D0, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
vact_space3	[7:0]	RW	vact_space3[7:0] of 13-bit. second active space start line number	0xFF

53.10.4.63 VACT_SPACE_3_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01D4, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
vact_space3	[4:0]	RW	vact_space3[12:8] of 13-bit. Second active space start line number	0x1F

53.10.4.64 VACT_SPACE_4_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01D8, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
vact_space4	[7:0]	RW	vact_space4[7:0] of 13-bit. Third active space end line number	0xFF

53.10.4.65 VACT_SPACE_4_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01DC, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
vact_space4	[4:0]	RW	vact_space4[12:8] of 13-bit. Third active space end line number	0x1F

53.10.4.66 VACT_SPACE_5_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01E0, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
vact_space5	[7:0]	RW	vact_space5[7:0] of 13-bit. Third active space start line number	0xFF

53.10.4.67 VACT_SPACE_5_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01E4, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
vact_space5	[4:0]	RW	vact_space5[12:8] of 13-bit. Third active space start line number	0x1F

53.10.4.68 VACT_SPACE_6_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01E8, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
vact_space6	[7:0]	RW	vact_space6[7:0] of 13-bit. Third active space end line number	0xFF

53.10.4.69 VACT_SPACE_6_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x01EC, Reset Value = 0x1F

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x00
vact_space6	[4:0]	RW	vact_space6[12:8] of 13-bit. Third active space end line number	0x1F

53.10.4.70 GCP_CON

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0200, Reset Value = 0x04

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	5b00000
ENABLE_1st_VSYNC	[3]	RW	For interlace mode, enable this bit to transfer GCP packet on first VSYNC in a frame. 0 = Does not transfer GCP packet 1 = Transfers GCP packet Alternatively, for progressive mode, GCP packet is transferred regardless of this bit. In progressive mode it transfers GCP packet every vsync if GCP_CON is 2b1x.	1b0
ENABLE_2nd_VSYNC	[2]	RW	For interlace mode, enable this bit to transfer GCP packet on second VSYNC in a frame. 0 = Does not transfer GCP packet 1 = Transfers GCP packet	1b1
GCP_CON	[1:0]	RW	00 = Does not transmit 01 = Transmits once 1x = Transmits every vsync Transmits GCP packet within 384 cycles after active vsync.	2b00

53.10.4.71 GCP_BYTE1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0210, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GCP_BYTE1	[7:0]	RW	Specifies first data byte of GCP packet. It is either 0x10 (Clear AVMUTE) or 0x01 (Set AVMUTE). Refer to Table 5-17 of HDMI specification for more information.	0x00

53.10.4.72 GCP_BYTE2

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0214, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
PP	[7:4]	RW	Specifies Packing Phase (PP). This bit is read only.	0x0
CD	[3:0]	RW	Specifies Color Depth (CD).	0x0

53.10.4.73 GCP_BYTE3

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0218, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	7b0000000
GCP_BYTE3	[0]	RW	Specifies default state.	0

53.10.5 HDMI Core Registers (Audio Related Registers)

53.10.5.1 ASP_CON

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0300, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
DST_Double	[7]	RW	Specifies the DST double	0
Aud_Type	[6:5]	RW	Specifies the packet type instead of audio type 00 = Audio Sample Packet 01 = One-bit audio packet 10 = HBR packet 11 = DST packet	2b00
Aud_Mode	[4]	RW	Selects the two-channel or multi-channel mode You can also use this bit for layout bit in ASP header. 0 = Two-channel mode 1 = Multi-channel mode Set this bit to transmit HBR packets.	0
SP_Pre	[3:0]	RW	Controls sub-packet usage for multi-channel mode only. When using two channel modes, it does not use this register value. [0]: AUDIO0 control 0 = Disables control 1 = Enables control [1]: AUDIO1 control 0 = Disables control 1 = Enables control [2]: AUDIO2 control 0 = Disables control 1 = Enables control [3]: AUDIO3 control 0 = Disables control 1 = Enables control	4b0000

53.10.5.2 ASP_SP_FLAT

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0304, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	4b0000
SP_Flat	[3:0]	RW	Specifies the sp_flat or sample_invalid value for ASP header Refer to the HDMI specification version 1.4 (5.3.4 and 5.3.9) for more information on SP_Flat.	0x0

53.10.5.3 ASP_CHCFG0/1/2/3

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0310, Reset Value = 0x08
- Address = Base Address + 0x0314, Reset Value = 0x1A
- Address = Base Address + 0x0318, Reset Value = 0x2C
- Address = Base Address + 0x031C, Reset Value = 0x3E

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	RW	Reserved	2b00
Spk3R_Sel	[29:27]	RW	Selects the audio channel for subpacket 3 right channel data in multi-channel mode. 000 = Uses i_pcm0L for sub packet 3 left channel 001 = Uses i_pcm0R for sub packet 3 right channel 010 = Uses i_pcm1L for sub packet 3 left channel 011 = Uses i_pcm1R for sub packet 3 right channel 100 = Uses i_pcm2L for sub packet 3 left channel 101 = Uses i_pcm2R for sub packet 3 right channel 110 = Uses i_pcm3L for sub packet 3 left channel 111 = Uses i_pcm3R for sub packet 3 right channel	3b111
Spk3L_Sel	[26:24]	RW	Selects the audio channel for subpacket 3 left channel data in multi-channel mode. The meaning is same as SPK3R_SEL.	3b110
RSVD	[23:22]	RW	Reserved	2b00
Spk2R_Sel	[21:19]	RW	Selects the audio channel for subpacket 2 right channel data in multi-channel mode. The meaning is same as SPK2R_SEL.	3b101
Spk2L_Sel	[18:16]	RW	Selects the audio channel selection for subpacket 2 left channel data in multi-channel mode. The meaning is same as SPK2R_SEL.	3b100
RSVD	[15:14]	RW	Reserved	2b00
SPK1R_SEL	[13:11]	RW	Selects the audio channel for subpacket 1 right channel data in multi-channel mode. The meaning is same as SPK1R_SEL.	3b011
Spk1L_Sel	[10:8]	RW	Selects the audio channel for subpacket 1 left channel data in multi-channel mode. The meaning is same as SPK1R_SEL.	3b010
RSVD	[7:6]	RW	Reserved	2b00
Spk0R_Sel	[5:3]	RW	Selects the audio channel for subpacket 0 right channel data in multi-channel mode. The meaning is same as SPK0R_SEL.	3b001
Spk0L_Sel	[2:0]	RW	Selects the audio channel selection for subpacket 0 left channel data in multi-channel mode. The meaning is same as SPK0R_SEL.	3b000

53.10.5.4 ACR_CON

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0400, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	3b000
RSVD	[4:3]	R	Reserved	2b00
ACR_Tx_Mode	[2:0]	RW	000 = Do not Tx (Transfer) ACR packet. 100 = Measures CTS mode. Makes ACR packet with CTS value by counting TMDS clock for Fs x 128/N duration. In this case, the 7 LSBs of N value (ACR_N register) should be all zero.	3b000

53.10.5.5 ACR_MCTS0/1/2

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0410, Reset Value = 0x01
- Address = Base Address + 0x0414, Reset Value = 0x00
- Address = Base Address + 0x0418, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[23:20]	RW	Reserved	0x0
ACR_MCTS	[19:0]	RW	Specifies the TMDS clock cycles for N[19:7] number of audio sample inputs. Only valid when measured CTS mode is set on ACR_CON register.	0x00001

53.10.5.6 ACR_N0/1/2

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0430, Reset Value = 0xE8
- Address = Base Address + 0x0434, Reset Value = 0x03
- Address = Base Address + 0x0438, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[23:20]	RW	Reserved	0
ACR_N	[19:0]	RW	Specifies the N value in ACR packet	0x003E8

53.10.6 HDMI Core Registers (Packet Related Registers)

53.10.6.1 ACP_CON

- Base Address: 0x12D1_0000
- Address = Base Address + 0x5000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ACP_FR_RATE	[7:3]	RW	Transmits the ACP packet once per every ACP_FR_RATE + 1 frames (or fields).	5b00000
RSVD	[2]	RW	Reserved	0
ACP_TX_CON	[1:0]	RW	00 = Does not transmit 01 = Transmits once 1x = Transmits every sync with ACP_FR_RATE	2b00

53.10.6.2 ACP_TYPE

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0514, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ACP_TYPE	[7:0]	RW	Specifies the HB1 of ACP packet header Refer to Table 5-18 in HDMI v1.4 specification for more information.	0x00

53.10.6.3 ACP_DATA00 to ACP_DATA16

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0520,+ 0x0524, + 0x0528, + 0x052C,
  + 0x0530, + 0x0534, + 0x0538, + 0x053C,
  + 0x0540, + 0x0544, + 0x0548, + 0x054C,
  + 0x0550, + 0x0554, + 0x0558, + 0x055C, + 0x0560, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ACP_DATA00 to ACP_DATA16	[7:0]	RW	Specifies the ACP packet body data registers (PB0 to PB16 of ACP packet body). Refer to Section 9.3 in HDMI v1.4 specification for more information.	0x00

53.10.6.4 ISRC_CON

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0600, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ISRC_FR_RATE	[7:3]	RW	Transmits ISRC1 (with or without ISRC2) packet once every ISRC_FR_RATE + 1 frames (or fields).	5b00000
ISRC2_EN	[2]	RW	Transmits ISRC2 packet with ISRC1 packet.	0
ISRC_TX_CON	[1:0]	RW	00 = Does not transmit 01 = Transmits once 1x = Transmits every vsync with ISRC_FR_RATE	2b00

53.10.6.5 ISRC1_HEADER1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0614, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ISRC_Cont	[7]	RW	Refer to Table 5-20 in HDMI v1.4 specification.	0
ISRC_Valid	[6]	RW	Refer to Table 5-20 in HDMI v1.4 specification.	0
RSVD	[5:3]	RW	Reserved	3b000
ISRC status	[2:0]	RW	Refer to Table 5-20 in HDMI v1.4 specification.	3b000

53.10.6.6 ISRC1_DATA00 to ISRC1_DATA15

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0620, + 0x0624, + 0x0628, + 0x062C,
  + 0x0630, + 0x0634, + 0x0638, + 0x063C,
  + 0x0640, + 0x0644, + 0x0648, + 0x064C,
  + 0x0650, + 0x0654, + 0x0658, + 0x065C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ISRC1_DATA00 to ISRC1_DATA15	[7:0]	RW	Specifies the ISRC2 packet body data (PB0 to15 of ISRC2 packet body). Refer to Table 5-21 in HDMI v1.4 specification for more information.	0x00

53.10.6.7 ISRC2_DATA00 to ISRC2_DATA15

- Base Address: 0x12D1_0000
- Address = Base Address + 0x06A0, + 0x06A4, + 0x06A8, + 0x06AC,
  + 0x06B0, + 0x06B4, + 0x06B8, + 0x06BC,
  + 0x06C0, + 0x06C4, + 0x06C8, + 0x06CC,
  + 0x06D0, + 0x06D4, + 0x06D8, + 0x06DC, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ISRC2_DATA00 to ISRC2_DATA15	[7:0]	RW	Specifies the ISRC2 packet body data (PB0 to 15 of ISRC2 packet body). Refer to Table 5-23 in HDMI v1.4 specification for more information.	0x00

53.10.6.8 AVI_CON

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0700, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
AVI_TX_CON	[1:0]	RW	00 = Does not transmit 01 = Transmits once 1x = Transmits every Vsync	2b00

53.10.6.9 AVI_HEADER0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0710, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AVI_HEADER0	[7:0]	RW	HB0 byte of AVI packet header	0x00

53.10.6.10 AVI_HEADER1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0714, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AVI_HEADER1	[7:0]	RW	HB1 byte of AVI packet header	0x00

53.10.6.11 AVI_HEADER2

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0718, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AVI_HEADER2	[7:0]	RW	HB2 byte of AVI packet header	0x00

53.10.6.12 AVI_CHECK_SUM

- Base Address: 0x12D1_0000
- Address = Base Address + 0x071C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AVI_CHECK_SUM	[7:0]	RW	Specifies the AVI InfoFrame checksum byte (PB0 byte of AVI packet body).	0x00

53.10.6.13 AVI_BYTE01 to AVI_BYTE13

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0720, + 0x0724, + 0x0728, + 0x072C,
  + 0x0730, + 0x0734, + 0x0738, + 0x073C,
  + 0x0740, + 0x0744, + 0x0748, + 0x074C, + 0x0750, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AVI_DATA01 to AVI_DATA13	[7:0]	RW	Specifies the AVI Info frame packet data registers (PB1 to PB13 bytes of AVI packet body).	0x00

53.10.6.14 AUI_CON

- Base Address: 0x12D1_0000
- Address = Base Address + 0x800, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
AUI_TX_CON	[1:0]	RW	00 = Does not transmit 01 = Transmits once 1x = Transmits every Vsync	2b00

53.10.6.15 AUI_HEADER0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0810, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AUI_HEADER0	[7:0]	RW	HB0 byte of AUI packet header	0x00

53.10.6.16 AUI_HEADER1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0814, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AUI_HEADER1	[7:0]	RW	HB1 byte of AUI packet header	0x00

53.10.6.17 AUI_HEADER2

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0818, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AUI_HEADER2	[7:0]	RW	HB2 byte of AUI packet header	0x00

53.10.6.18 AUI_CHECK_SUM

- Base Address: 0x12D1_0000
- Address = Base Address + 0x081C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AUI_CHECK_SUM	[7:0]	RW	Specifies the AUI checksum data (PB0 byte of AUI packet body)	0x00

53.10.6.19 AUI_BYTE1 to AUI_BYTE12

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0820, + 0x0824, + 0x0828, + 0x082C,
+ 0x0830, + 0x0834, + 0x0838, + 0x083C,
+ 0x0840, + 0x0844, + 0x0848, + 0x084C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AUI_BYTE1 to AUI_BYTE12	[7:0]	RW	AUI Infoframe packet body (PB1 to PB12 bytes of AUI packet body)	0x00

53.10.6.20 MPG_CON

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0900, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
MPG_TX_CON	[1:0]	RW	00 = Does not transmit 01 = Transmits once 1x = Transmits every Vsync	2b00

53.10.6.21 MPG_CHECK_SUM

- Base Address: 0x12D1_0000
- Address = Base Address + 0x091C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
MPG_CHECK_SUM	[7:0]	RW	Specifies the MPG info frame checksum register (PB0 byte of MPG packet body).	0x00

53.10.6.22 MPG_DATA1 to MPG_DATA6

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0920, + 0x0924, + 0x0928, + 0x092C, + 0x0930, + 0x0934, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
MPG_DTAT1 to MPG_DATA6	[7:0]	RW	Specifies the MPG Info frame packet data (PB1 to PB6 bytes of MPG packet body).	All Zeros

Source Product Descriptor Info frame (General Packet Generation)

These registers can be used for Source Product Descriptor (SPD) packet transmission. Furthermore, they consist of full configurable header and packet body registers (3-bytes header register and 28-bytes packet body registers), so that they can be used for transmission of any type of packet.

53.10.6.23 SPD_CON

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0A00, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
SPD_TX_CON	[1:0]	RW	00 = Does not transmit 01 = Transmits once 1x = Transmits every Vsync	2b00

53.10.6.24 SPD_HEADER0/1/2

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0A10, + 0x0A14, + 0x0A18, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
SPD_HEADER0	[7:0]	RW	Specifies the HB0 byte of SPD packet header	0x00
SPD_HEADER1			Specifies the HB1 byte of SPD packet header	
SPD_HEADER2			Specifies the HB2 byte of SPD packet header	

53.10.6.25 SPD_DATA00 to SPD_DATA27

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0A20, + 0x0A24, + 0x0A28, + 0x0A2C,
  + 0x0A30, + 0x0A34, + 0x0A38, + 0x0A3C,
  + 0x0A40, + 0x0A44, + 0x0A48, + 0x0A4C,
  + 0x0A50, + 0x0A54, + 0x0A58, + 0x0A5C,
  + 0x0A60, + 0x0A64, + 0x0A68, + 0x0A6C,
  + 0x0A70, + 0x0A74, + 0x0A78, + 0x0A7C,
  + 0x0A80, + 0x0A84, + 0x0A88, + 0x0A8C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
SPD_DATA00 to SPD_DATA27	[7:0]	RW	Specifies the SPD packet data registers (PB0 to PB27 bytes).	0x00

53.10.6.26 GAMUT_CON

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0B00, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b00000
GAMUT_CON	[1:0]	RW	00 = Does not transmit 01 = Transmits once 1x = Transmits every Vsync	2b00

53.10.6.27 GAMUT_HEADER0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0B10, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HB0	[7:0]	RW	Specifies the HB0 value in Table 5-30 Refer to HDMI v1.4 specification for more information.	0x00

53.10.6.28 GAMUT_HEADER1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0B14, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Next_Field	[7]	RW	Indicates the effectiveness of GBD carried in this packet on the next video field	0
GBD_profile	[6:4]	RW	Specifies the transmission profile number (only profile 0 is supported)	3b000
Affected_Gamut_Seq_Num	[3:0]	RW	Indicates which video fields are relevant for this metadata	4b0000

53.10.6.29 GAMUT_HEADER2

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0B18, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
No_Crnt_GBD	[7]	RW	Indicates that there is no gamut metadata available for currently transmitted video.	0
RSVD	[6]	RW	Reserved	0
Packet_Seq	[5:4]	RW	Indicates whether this packet is first, intermediate, last or the only packet in a gamut metadata packet sequence.	2b00
Current_Gamut _Seq_Num	[3:0]	RW	Indicates the gamut number of the currently transmitted video stream.	4b0000

53.10.6.30 GAMUT_METADATA00 to GAMUT_METADATA27

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0B20, + 0x0B24, + 0x0B28, + 0x0B2C,
  + 0x0B30, + 0x0B34, + 0x0B38, + 0x0B3C,
  + 0x0B40, + 0x0B44, + 0x0B48, + 0x0B4C,
  + 0x0B50, + 0x0B54, + 0x0B58, + 0x0B5C,
  + 0x0B60, + 0x0B64, + 0x0B68, + 0x0B6C,
  + 0x0B70, + 0x0B74, + 0x0B78, + 0x0B7C,
  + 0x0B80, + 0x0B84, + 0x0B88, + 0x0B8C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GAMUT_METADATA0 to GAMUT_METADATA27	[7:0]	RW	Specifies the gamut metadata packet body for P0 transmission profile.	0x00

53.10.6.31 VSI_CON

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0C00, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	0x00
VSI_TX_CON	[1:0]	RW	00 = Do not transmit 01 = Transmit once 1x = Transmit every Vsync	0x0

53.10.6.32 VSI_HEADER0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0C10, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
VSI_HEADER0	[7:0]	RW	HB0 byte of VSI packet header	0x00

53.10.6.33 VSI_HEADER1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0C14, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
VSI_HEADER1	[7:0]	RW	HB1 byte of VSI packet header	0x00

53.10.6.34 VSI_HEADER2

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0C18, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
VSI_HEADER2	[7:0]	RW	HB2 byte of VSI packet header	0x00

53.10.6.35 VSI_DATA00 to VSI_DATA27

- Base Address: 0x12D1_0000
- Address = Base Address: 0x0C20, + 0x0C24, + 0x0C28, + 0x0C2C,
  + 0x0C30, + 0x0C34, + 0x0C38, + 0x0C3C,
  + 0x0C40, + 0x0C44, + 0x0C48, + 0x0C4C,
  + 0x0C50, + 0x0C54, + 0x0C58, + 0x0C5C,
  + 0x0C60, + 0x0C64, + 0x0C68, + 0x0C6C,
  + 0x0C70, + 0x0C74, + 0x0C78, + 0x0C7C,
  + 0x0C80, + 0x0C84, + 0x0C88, + 0x0C8C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
VSI_DATA00 to VSI_DATA27	[7:0]	RW	VSI packet data registers. (PB0 to PB27 bytes)	0x00

53.10.6.36 VIDEO_PATTERN_GEN

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0D04, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
Ext_Video_En	[1]	RW	0 = Ext off 1 = Ext en	0
Video Pattern Enable	[0]	RW	0 = Disables 1 = Uses video pattern internally generated	0

53.10.6.37 An_Seed_Sel

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0E48, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0x7F
An_Seed_Sel	[0]	RW	0 = Use An_Seed_0 to 3 registers as a seed. 1 = Use input R/G/B as a seed.	0x1

53.10.6.38 An_Seed_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0E58, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
An_Seed	[7:0]	RW	[23:16] of An seed value	0x0

53.10.6.39 An_Seed_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0E5C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	0x00
An_Seed	[3:0]	RW	[15:12] of An seed value	0x0

53.10.6.40 An_Seed_2

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0E60, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
An_Seed	[7:0]	RW	[11:4] of An seed value	0x0

53.10.6.41 An_Seed_3

- Base Address: 0x12D1_0000
- Address = Base Address + 0x0E64, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	0x00
An_Seed	[3:0]	RW	[3:0] of An seed value	0x0

53.10.7 HDMI Core Registers (HDCP Related Registers)

53.10.7.1 HDCP_SHA1_00 to HDCP_SHA_19

- Base Address: 0x12D1_0000
- Address = Base Address + 0x7000, + 0x7004, + 0x7008, + 0x700C,
  + 0x7010, + 0x7014, + 0x7018, + 0x701C,
  + 0x7020, + 0x7024, + 0x7028, + 0x702C,
  + 0x7030, + 0x7034, + 0x7038, + 0x703C,
  + 0x7040, + 0x7044, + 0x7048, + 0x704C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HDCP_SHA1	[159:0]	RW	Specifies the SHA-1 value of 160-bit HDCP repeater (LSB first). These registers are readable but they are not modified by HDCP H/W.	All zeros

53.10.7.2 HDCP_KSV_LIST_0 to HDCP_KSV_LIST_4

- Base Address: 0x12D1_0000
- Address = Base Address + 0x7050, + 0x7054, + 0x7058, + 0x705C, + 0x7060, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HDCP_KSV_LIST	[39:0]	RW	Specifies little endian addressing and one KSV value from KSV list of HDCP repeater. These registers are readable.	All zeros

53.10.7.3 HDCP_KSV_LIST_CON

- Base Address: 0x12D1_0000
- Address = Base Address + 0x7064, Reset Value = 0x01

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	4b0000
Hdcp_Ksv_Write_Done	[3]	RW	After writing KSV data into HDCP_KSV_LIST_X registers and then writing "1" to this register, the HW processes the written KSV value and clears this bit to "0". 0 = Does not write 1 = Writes KSV data into HDCP_KSV_LIST_X registers and then writes "1" to this register	0
Hdcp_Ksv_List_Empty	[2]	RW	When the number of KSV list is zero, set this value to enable the SHA-1 module calculate without KSV list. 0 = Not empty 1 = Empty	0
Hdcp_Ksv_End	[1]	RW	Indicates that current KSV value in HDCP_KSV_LIST_X registers is the last one. 0 = Not End 1 = End	0
Hdcp_Ksv_Read	[0]	RW	After writing KSV data in HDCP_KSV_LIST_X registers, the HDCP SHA-1 module keeps the KSV value in internal buffer. Set this flag to 1 to notify that you have read it. After setting the flag to "1", the SW clears to "0" at the same time when writing the HDCP_KSV_WRITE_DONE bit for next KSV list value. 0 = Not Read 1 = Read	1

53.10.7.4 HDCP_SHA_RESULT

- Base Address: 0x12D1_0000
- Address = Base Address + 0x7070, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
Hdcp_Sha_Valid_Ready	[1]	RW	Indicates that HW performs SHA comparison. S/W should clear it by writing 0. 0 = Not ready 1 = Ready	0
Hdcp_Sha_Valid	[0]	RW	Indicates that the SHA-1 comparison succeeds. S/W should clear it by writing 0. 0 = Valid 1 = Not valid	0

53.10.7.5 HDCP_CTRL1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x7080, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	4b0000
RSVD	[3]	RW	Reserved	0
Timeout	[2]	RW	Sets the bit if Rx is the repeater and its KSV list is not ready until five seconds. 0 = Does not timeout 1 = Timeout (KSV Ready bit in the HDCP_BCAPS register is not high until five seconds) and re-starts the 1st authentication. Refer to Figure 2-6 in the HDCP 1.3 specification for more information.	0
CP_Desired	[1]	RW	Enables HDCP 0 = Does not use CP Desired 1 = User CP Desired	0
RSVD	[0]	RW	Reserved	0

53.10.7.6 HDCP_CTRL2

- Base Address: 0x12D1_0000
- Address = Base Address +0x7084, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	7b0000000
Revocation_Set	[0]	RW	Specifies the KSV list that is on the revocation list Setting this bit fails the 2nd authentication. 0 = Does not set revocation 1 = Sets revocation	0

53.10.7.7 HDCP_CHECK_RESULT

- Base Address: 0x12D1_0000
- Address = Base Address + 0x7090, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
Ri_Match_Result	[1:0]	RW	Writes the result of comparison between Ri of Rx and Tx as the values: (Ri: Tx, Ri': Rx) SW should clear it after setting 10 or 11 before next Ri interrupt occurs. 0x = Don't care 10 = Ri ≠ Ri' 11 = Ri = Ri'	2b00

53.10.7.8 HDCP_BKSV0 to HDCP_BKSV4

- Base Address: 0x12D1_0000
- Address = Base Address + 0x70A0, + 0x70A4, + 0x70A8, + 0x70AC, + 0x70B0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HDCP_BKSV	[39:0]	RW	Specifies the key selection vector (KSV) value from receiver	All zeros

53.10.7.9 HDCP_AKSV0 to HDCP_AKSV4

- Base Address: 0x12D1_0000
- Address = Base Address + 0x70C0, + 0x70C4, + 0x70C8, + 0x70CC, + 0x70D0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HDCP_AKSV	[39:0]	RW	Specifies the KSV value of transmitter	All zeros

53.10.7.10 HDCP_An_0 to HDCP_An_7

- Base Address: 0x12D1_0000
- Address = Base Address + 0x70E0, + 0x70E4, + 0x70E8, + 0x70EC,
  + 0x70F0, + 0x70F4, + 0x70F8, + 0x70FC, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HDCP_An	[63:0]	RW	Specifies the 64-bit random number generated by Tx (An)	All zeros

53.10.7.11 HDCP_BCAPS

- Base Address: 0x12D1_0000
- Address = Base Address + 0x7100, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	0
Repeater	[6]	RW	Specifies the receiver that supports downstream connections 0 = Does not set Repeater 1 = Set Repeater	0
Ready	[5]	RW	Indicates KSV FIFO, SHA-1 is calculation ready 0 = Not Ready 1 = Ready	0
Fast	[4]	RW	Specifies the receiver devices that support 400 kHz transfer. 0 = Not Fast 1 = Fast	0
RSVD	[3:2]	RW	Reserved. Must be 0.	0
1.1_Features	[1]	RW	Supports EESS, advance cipher, and enhanced link verification. 0 = Does not set feature 1 = Sets feature	0
Fast_Reauthentication	[0]	RW	Specifies all HDMI receivers that are capable of reauthentication. 0 = Does not set 1 = Sets	0

53.10.7.12 HDCP_BSTATUS_0/1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x7110, + 0x7114, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[15:13]	RW	Reserved	3b000
Hdmi_Mode	[12]	RW	Specifies the HDMI mode When set, HDCP works in HDMI mode.	0
Max_Cascade_ Exceeded	[11]	RW	Specifies the topology error	0
Depth	[10:8]	RW	Specifies the cascade depth	3b000
Max_Devs_ Exceeded	[7]	RW	Indicates the topology error 0 = No Error 1 = Error	0
Device_Count	[6:0]	RW	Specifies the total number of downstream devices that are attached.	0

53.10.7.13 HDCP_Ri_0/1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x7140, + 0x7144, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HDCP_Ri	[15:0]	RW	Specifies the HDCP Ri value of transmitter	0x0000

53.10.7.14 HDCP_I2C_INT

- Base Address: 0x12D1_0000
- Address = Base Address + 0x7180, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	7b0000000
HDCP_I2C_INT	[0]	RW	Specifies the HDCP I2C interrupt status (active high) It indicates the start of I2C transaction when it is set. After active, S/W should clear it by writing 0. 0 = Does not occur 1 = Occurs	0

53.10.7.15 HDCP_AN_INT

- Base Address: 0x12D1_0000
- Address = Base Address + 0x7190, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	7b0000000
HDCP_AN_INT	[0]	RW	Specifies the HDCP An Interrupt status (active high) When A value is available, it is set. After active, S/W should clear it by writing 0. 0 = Does not occur 1 = Occurs	0

53.10.7.16 HDCP_WATCHDOG_INT

- Base Address: 0x12D1_0000
- Address = Base Address + 0x71A0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	7b0000000
HDCP_WATCHDOG_INT	[0]	RW	Specifies the HDCP watchdog interrupt status (active high) When the repeater bit value is set after first authentication success, this bit is set. After active, S/W should clear it by writing 0. 0 = Does not occur 1 = Occurs	0

53.10.7.17 HDCP_Ri_INT

- Base Address: 0x12D1_0000
- Address = Base Address + 0x71B0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	7b0000000
HDCP_Ri_INT	[0]	RW	When it updates Ri internally (at every 128 video frames), it is set to high. After set, S/W should clear it by writing 0. 0 = Does not occur 1 = Occurs	0

53.10.7.18 HDCP_Ri_Compare_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0x71D0, Reset Value = 0x80

Name	Bit	Type	Description	Reset Value
Enable	[7]	RW	Enables the interrupt for this frame number index.	1
Frame Number index	[6:0]	RW	When the frame count reaches "frame number index", a Ri link integrity check interrupt occurs.	7b0000000

53.10.7.19 HDCP_Ri_Compare_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0x71D4, Reset Value = 0x7F

Name	Bit	Type	Description	Reset Value
Enable	[7]	RW	Enables the interrupt for this frame number index.	0
Frame Number index	[6:0]	RW	When the frame count reaches "frame number index", a Ri Link integrity check interrupt occurs.	7b1111111

53.10.7.20 HDCP_Frame_Count

- Base Address: 0x12D1_0000
- Address = Base Address + 0x71E0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	R	Reserved	0
Frame Count	[6:0]	R	Specifies the current value of frame count index in hardware.	7b0000000

53.10.7.21 RGB_ROUND_EN

- Base Address: 0x12D1_0000
- Address = Base Address + 0xD500, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0x00
rgb_round_en	[0]	RW	RGB Rounding enable	0

53.10.7.22 VACT_SPACE_R_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0xD504, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
vact_space_r	[7:0]	RW	vact_space_r[7:0] of 12 bits. Constant pixel color in vact space.	0

53.10.7.23 VACT_SPACE_R_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0xD508, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	0x00
vact_space_r	[3:0]	RW	vact_space_r[11:8] of 12 bits. Constant pixel color in vact space.	0

53.10.7.24 VACT_SPACE_G_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0xD50C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
vact_space_g	[7:0]	RW	vact_space_g[7:0] of 12 bits. Constant pixel color in vact space.	0

53.10.7.25 VACT_SPACE_G_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0xD510, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	0x00
vact_space_g	[3:0]	RW	vact_space_g[11:8] of 12 bits. Constant pixel color in vact space.	0

53.10.7.26 VACT_SPACE_B_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0xD514, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
vact_space_b	[7:0]	RW	vact_space_b[7:0] of 12 bits. Constant pixel color in vact space.	0

53.10.7.27 VACT_SPACE_B_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0xD518, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	0x00
vact_space_b	[3:0]	RW	vact_space_b[11:8] of 12 bits. Constant pixel color in vact space.	0

53.10.7.28 BLUE_SCREEN_R_0

- Base Address: 0x12D1_0000
- Address = Base Address +, 0xD520, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
blue_screen_r	[7:0]	RW	blue_screen_r[7:0] of 12 bits.	0x0

53.10.7.29 BLUE_SCREEN_R_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0xD524, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	0x0
blue_screen_r	[3:0]	RW	blue_screen_r[11:8] of 12 bits.	0x0

53.10.7.30 BLUE_SCREEN_G_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0xD528, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
blue_screen_g	[7:0]	RW	blue_screen_g[7:0] of 12 bits.	0x0

53.10.7.31 BLUE_SCREEN_G_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0xD52C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	0x0
blue_screen_g	[3:0]	RW	blue_screen_g[11:8] of 12 bits.	0x0

53.10.7.32 BLUE_SCREEN_B_0

- Base Address: 0x12D1_0000
- Address = Base Address + 0xD530, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
blue_screen_b	[7:0]	RW	blue_screen_b[7:0] of 12 bits.	0x0

53.10.7.33 BLUE_SCREEN_B_1

- Base Address: 0x12D1_0000
- Address = Base Address + 0xD534, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	0x0
blue_screen_b	[3:0]	RW	blue_screen_b[11:8] of 12 bits.	0x0

53.10.8 SPDIF Registers

53.10.8.1 SPDIFIN_CLK_CTRL

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0000, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
ready_clk_down	[1]	RW	0 = Enables clock 1 = Ready for disabling clock (default)	1
power_on	[0]	RW	0 = Disables clock (default) 1 = Activates clock When this bit is reset, SPDIFIN stops checking input signal just before the next "subframe" of SPDIF signal format. It also waits for the "acknowledge" signal from HDMI for unresolved previous "request" towards HDMI. It then asserts "ready_clk_down" as high. To initialize internal states, assert software reset, that is, SPDIFIN_OP_CTRL. op_ctrl = 00b right after activating clock again.	0

The spdif_clk is gated by an external clock gating module for low-power. Disabling the clock should not cause stalling of HDMI data transfer. Therefore, the system processor requests disabling of the clock by setting the power_on register to low. The module acknowledges this request by setting the ready_clk_down register to high after a current transaction on the I2C bus, and HDMI finishes. The module must not commence a new bus transaction until the system processor sets the power_on register to high again.

Figure 53-12 illustrates the structure of power down circuit.

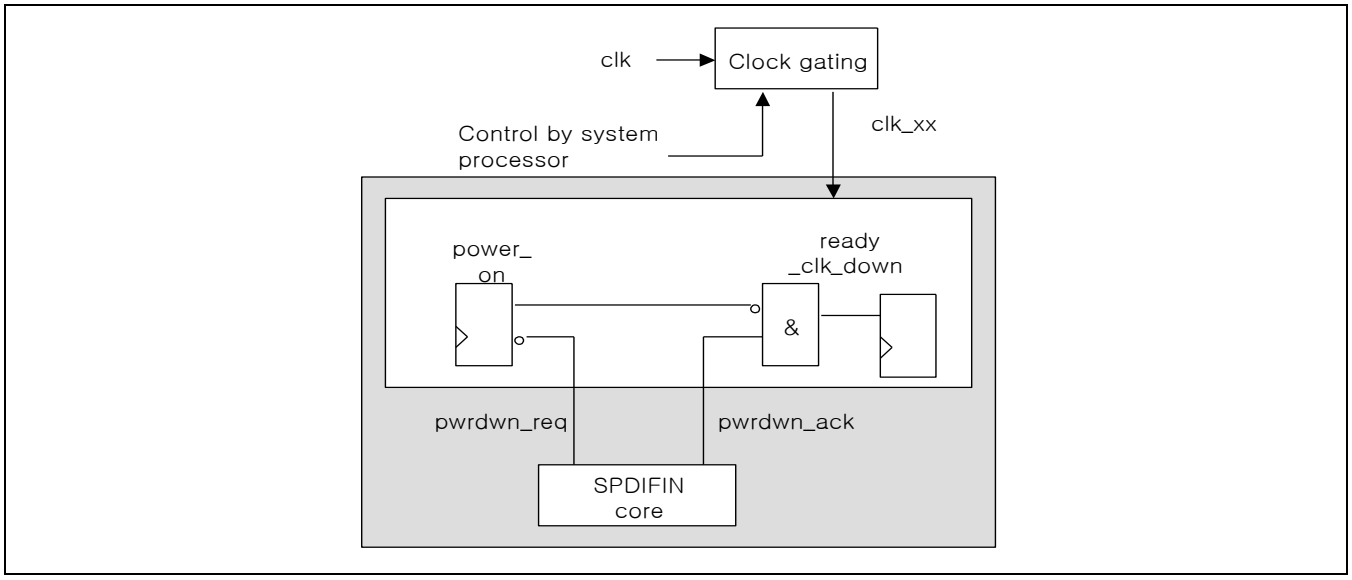


Figure 53-12 Structure of Power Down Circuit

The system processor switches off clk_xx when the ready_clk_down bit is 1 and the power_on bit is 0.

The system processor switches on the clock at any time. After it switches on clk_xx, the system processor sets the power_on bit to 1 that forces ready_clk_down bit to 0.

Once it switches off the reset clock of SPDIFIN, the power_on bit is set to zero and the ready_clk_down bit is set to one.

53.10.8.2 SPDIFIN_OP_CTRL

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
op_ctrl	[1:0]	RW	00b = Specifies the software reset 01b = Specifies the status checking mode (run) 11b = Specifies the status checking and HDMI operation modes (run with HDMI) Others = Undefined, do not use 00b = During a software reset, all state machines are set to idle or initial state and all internal registers are set to their initial values. Clears interrupt status registers; all other registers that are writable by system processor keep their values. 01b = This command should be asserted after SPDIFIN_CLK_CTRL.power_on is set. SPDIFIN starts the clock recovery. When recovery is done, SPDIFIN detects preambles of SPDIF signal format and stream data header, abnormal time signal input, abnormal signal input. It also reports this status through interrupts in SPDIFIN_IRQ_STATUS. 11b = Specifies the "01b" case operations, checks internal buffer overflow, and writes received data that can be audio sample word of PCM or payload of stream. Data will be transferred via HDMI. • Assert "op_ctrl" = 11b after SPDIFIN_IRQ_STATUS. It asserts ch_status_recovered_ir at least once for linear PCM data. • Assert "op_ctrl" = 11b after SPDIFIN_IRQ_STATUS. It asserts stream_header_detected_ir at least once for non-linear PCM stream data.	0

53.10.8.3 SPDIFIN_IRQ_MASK

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0008
- Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
buf_overflow_ir_en	[7]	RW	Specifies the mask bit for Interrupt 7	0
RSVD	[6]	RW	Reserved	0
RSVD	[5]	RW	Reserved	0
stream_header _detected_ir_en	[4]	RW	Specifies the mask bit for Interrupt 4	0
stream_header _not_detected_ir_en	[3]	RW	Specifies the mask bit for Interrupt 3	0
wrong_preamble _ir_en	[2]	RW	Specifies the mask bit for Interrupt 2	0
ch_status_recovered _ir_en	[1]	RW	Specifies the mask bit for Interrupt 1	0
wrong_signal_ir_en	[0]	RW	Specifies the mask bit for Interrupt 0	0

NOTE: For every bit:
0 = Disables interrupt generation
1 = Enables interrupt generation

53.10.8.4 SPDIFIN_IRQ_STATUS

- Base Address: 0x12D3_0000
- Address = Base Address + 0x000C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
buf_overflow_ir	[7]	RW	0 = No interrupt 1 = Internal buffer overflow SPDIFIN internal buffer (s) (SPDIFIN_DATA_BUF_x) overflows if HDMI fails to transfer data from buffer (s) to memory on time. • This interrupt is asserted only if SPDIFIN_OP_CTRL.op_ctrl is set to "011". • When this interrupt is not handled, SPDIFIN overwrites next subframe data to internal data buffer (SPDIFIN_DATA_BUF_x) and continues data transfer via HDMI.	0
RSVD	[6]	RW	Reserved	0
RSVD	[5]	RW	Reserved	0
stream_header_detected_ir	[4]	RW	0 = No interrupt 1 = Detects stream data header (Pa to Pd) • This interrupt is asserted if SPDIFIN_OP_CTRL.op_ctrl is equal to 001b or 011b. • Cases for interrupt Case1: Initially after power_on Case2: Next stream header at right time if receiving stream data with SPDIFIN_CONFIG.data_type set as "stream mode" Case3: Initially detects stream header if receiving stream data with SPDIFIN_CONFIG.data_type is set as "PCM mode"	0
stream_header_not_detected_ir	[3]	RW	0 = No interrupt 1 = Does not detect stream data header for 4096 repetition time • This interrupt will be asserted if SPDIFIN_OP_CTRL.op_ctrl is equal to 001b or 011b. • Cases for interrupt Case1: Initially after power_on Case2: SPDIFIN receives the stream but is unable to find the next stream header for 4096 times repetition since previous stream header Case3: Is unable to find stream header for 4096 times repetition since previous reset of repetition time counter after previous interrupt of "stream_header_not_detected_ir".	0
wrong_preamble_ir	[2]	RW	0 = No interrupt	0

Name	Bit	Type	Description	Reset Value
			1 = Detects preamble but indicates a problem with the detected time • This interrupt is asserted when SPDIFIN_OP_CTRL.op_ctrl is equal to 001b or 011b. • Meaningless until ch_status_recovered_ir is asserted initially after SPDIFIN_OP_CTRL.op_ctrl = 01b • Cases for interrupt Case1: Detects preamble in the middle of a subframe audio sample word time. Case2: Does not detect the next preamble at exact time after a subframe duration. Case3: Does not detect preamble B (or M or W) but detects other preamble at that time.	
ch_status_recovered_ir	[1]	RW	0 = No interrupt 1 = Recovers channel status Detects two consecutive B-preambles; thus recovers 192-bit wide channel status. • Only supports consumer mode. That is why it is able to reconstruct 36 bits. When you want to see the channel status bits through SPDIFIN_CH_STATUS_x, read two consecutive "ch_status_recovered_ir" and the register each time. If these two channel status values are the same, you can rely on that value.	0
wrong_signal_ir	[0]	RW	0 = No interrupt 1 = Clock recovery fails Cannot recover the clock from input due to tolerable range violation (unlock), no signal from outside, or non-bi-phase in non-preamble duration. • Meaningless until ch_status_recovered_ir is asserted initially after SPDIFIN_OP_CTRL.op_ctrl = 01b.	0

For every bit, the following holds: Reading returns interrupt request status. Writing "0" has no effect. Writing "1" clears the interrupt request.

1) Detection of stream header

   Waits for matching of Pa, Pb and 0xF872, 0x4E1F respectively.

   Waits for repetition time (from decoded PC value or user-defined PC in SPDIFIN_USER_VALUE.repetition_time_manual, based on SPDIFIN_CONFIG. PcPd_value_mode)

Check for matching of Pa, Pb on right time.

53.10.8.5 SPDIFIN_CONFIG_1

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0010, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	0
noise_filter_samples	[6]	RW	0 = Filtering with three consecutive samples 1 = Filtering with two consecutive samples Noise filtering is done for over-sampled SPDIF input signal. This operation will be done as: If "noise_filter_samples" is 0, three consecutive over-sampled signals are as high or low only if those samples are all high or all low respectively. If one or two samples are low or high for three over-sample duration, those noise-filtered signals will keep previous value. If "noise_filter_samples" is 1, two consecutive over-sampled signals are as high or low only if those samples are all high or low respectively. This setting can be used for reduced over-sampling ratio. It recommends over-sampling ratio of 10 (see also "clk_divisor").	0
RSVD	[5]	RW	Reserved (Must be "0")	0
PcPd_value_mode	[4]	RW	0 = Sets Automatically 1 = Sets Manually If "0" is used for automatic setting, Pc and Pd values are chosen by value of Pc and Pd from decoded stream header, as it reports in SPDIFIN_Px_INFO. If you set this register, the receiver will use SPDIFIN_USER_VALUE[31:16] and SPDIFIN_USER_VALUE[15:4] values as Pc and Pd respectively, instead of decoded data from stream header, as it reports in SPDIFIN_Px_INFO. If the (cf) burst payload length is automatically or manually set, it affects the data size to be written in memory through HDMI-by dumping the full sub-frame for last bit in burst payload length. For example, if burst payload length is 257-bit, that is, 16 sub-frames × 16-bit + 1-bit, then HDMI will write data in 17 consecutive sub-frames.	0
word_length_value_mode	[3]	RW	0 = Sets Automatically 1 = Sets Manually If "0" is used for automatic setting, the word length value will be chosen by value of channel status from decoded SPDIF format, as it reports in SPDIFIN_CH_STATUS_1.word_length. If user sets this register, the receiver will use SPDIFIN_USER_VALUE [3:0] value as word length	0

Name	Bit	Type	Description	Reset Value
			instead of decoded data from channel status, as it reports in SPDIFIN_CH_STATUS_1.word_length.	
U_V_C_P_report	[2]	RW	0 = Neglects "user_bit", "validity_bit", "channel status", and "parity_bit" of SPDIF format. 1 = Reports "user_bit", "validity_bit", "channel status", and "parity_bit" of SPDIF format The report will be delivered via HDMI for each sub-frame. Valid only if SPDIFIN_CONFIG. Data_align is set for 32-bit mode. See SPDIFIN_DATA_BUF_x for more information.	0
RSVD	[1]	RW	Reserved (Must be "1")	1
data_align	[0]	RW	0 = 16-bit mode 1 = 32-bit mode 16-bit: Only takes 16-bit from MSB in a sub-frame of SPDIF format, and then concatenates two consecutive 16-bit data in one 32-bit register of SPDIFIN_DATA_BUF_x. 32-bit: Only takes data from one sub-frame with zero padding to MSB part. For example, 0x00ffffff for 24-bit data. With stream mode, set "word_length_value_mode" as 1 and set SPDIFIN_USER_VALUE.word_length_manual as 3b000. • These two modes will be applied to both modes of SPDIFIN_CONFIG.data_type, that is, PCM or stream. See SPDIFIN_DATA_BUF_x for more information.	0

53.10.8.6 SPDIFIN_CONFIG_2

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0014, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	0x0
clk_divisor	[3:0]	RW	SPDIFIN_internal_clock = system_clock/(clk_divisor + 1) (SPDIFIN_internal_clock ≤ 135 MHz) SPDIFIN over-samples the SPDIF input signal with internal clock that is divided from system clock. It recommends over-sampling ratio from 8 to 10, thus the calculation holds: Recommended SPDIFIN_internal_clock = Sampling Frequency of SPDIF Input Signal × 64-bit × 10 times over-sampling For example, 48 kHz × 64-bit × 10 times over-sampling = 31 MHz.	0x0

53.10.8.7 SPDIFIN_USER_VALUE_1

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0020, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
repetition_time_manual_low	[7:4]	RW	Specifies repetition time[3:0] Repetition_time_manual register has 12-bit value. This register is low by 4-bit. It counts one block of stream data and is valid only if SPDIFIN_CONFIG. PcPd_value_mode is set for manual mode. Unit: frames (1 frame = 2 sub-frames) of SPDIF format. The value should be actual repetition time minus one. For example, when you want to manually set the repetition time as 1536, you should write 1535.	0x0
word_length_manual	[3:0]	RW	Specifies the word length Uses it as size for transferring data to memory through HDMI; valid only when SPDIFIN_CONFIG.word_length_value_mode is set for manual mode. See SPDIFIN_DATA_BUF_x for more information. [0] is 1 [0] is 0 [3:1] 101: 24-bit 20-bit 001: 23-bit 19-bit 010: 22-bit 18-bit 011: 21-bit 17-bit 100: 20-bit 16-bit	0x0

53.10.8.8 SPDIFIN_USER_VALUE_2

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0024, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
repetition_time _manual_high	[7:0]	RW	Specifies the repetition time[11:4] Repetition_time_manual register has 12-bit value. This register is high by 8-bits. Counts one block of stream data; valid only if SPDIFIN_CONFIG. PcPd_value_mode is set for manual mode. Unit: frames (1 frame = 2 sub-frames) of SPDIF format The value should be actual repetition time minus one. For example, when you want to manually set the repetition time as 1536, you should write 1535.	0x00

53.10.8.9 SPDIFIN_USER_VALUE_3

- Base Address: 0x12D3_0000
- Address = Base Address +0x0028, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
burst_payload _length_manual_low	[7:0]	RW	Specifies the burst_payload_length_manual[7:0] Burst_payload_length register has 16-bits value. This register is low by 8-bit. Valid only if SPDIFIN_CONFIG. PcPd_value_mode is set for manual mode. Unit: bits	0

53.10.8.10 SPDIFIN_USER_VALUE_4

- Base Address: 0x12D3_0000
- Address = Base Address + 0x002C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
burst_payload _length_manual_high	[7:0]	RW	Specifies the burst_payload_length_manual[15:8] Burst_payload_length register has 16-bit value. This register is high by 8-bit. Valid only if SPDIFIN_CONFIG. PcPd_value_mode is set for manual mode. Unit: bits.	0

Channel Status registers
it updates channel status register for every 192 frames (1 block) of SPDIF format only for consumer mode.

53.10.8.11 SPDIFIN_CH_STATUS_0_1

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0030, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
channel_status_mode	[7:6]	R	00 = Mode 0 others = Reserved	2b00
emphasis	[5:3]	R	000 = Does not indicate Emphasis 100 = Indicates Emphasis-CD type	3b000
copyright_assertion	[2]	R	0 = Copyright 1 = No copyright	0
audio_sample_word	[1]	R	0 = Linear PCM 1 = Non-linear PCM	0
channel_status_block	[0]	R	0 = Consumer format 1 = Professional format	0

This register is updated every 192 frames (1block) of SPDIF format.
SPDIFIN_CH_STATUS_0_1 [7:0] is the matched internal register SPDIFIN_CH_STATUS_0 [7:0].

53.10.8.12 SPDIFIN_CH_STATUS_0_2

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0034, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
category_code	[7:0]	R	Equipment type: [8:15] CD player: 1000_0000 DAT player: 1100_000L DCC player: 1100_001L Mini disc: 1001_001L (L: information about generation status of the material)	0x00

53.10.8.13 SPDIFIN_CH_STATUS_0_3

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0038, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
channel_number	[7:4]	R	Specifies the channel number (Bit[20] is LSB)	0x0
source_number	[3:0]	R	Specifies the source number (Bit[16] is LSB)	0x0

53.10.8.14 SPDIFIN_CH_STATUS_0_4

- Base Address: 0x12D3_0000
- Address = Base Address + 0x003C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	R	Reserved	2b00
clock_accuracy	[5:4]	R	Specifies the clock accuracy 00 = level II, ± 1000 ppm 01 = level I, ± 50 ppm 10 = level III, variable pitch shifted	2b00
sampling_frequency	[3:0]	R	Specifies the sampling frequency 0100 = 22.05 kHz 0000 = 44.1 kHz 1000 = 88.2 kHz 1100 = 176.4 kHz 0110 = 24 kHz 0010 = 48 kHz 1010 = 96 kHz 1110 = 192 kHz 0011 = 32 kHz	0x0

53.10.8.15 SPDIFIN_CH_STATUS_1

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0040, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	R	Reserved	0x0
word_length	[3:1]	R	Specifies the word length (field_size = 1), (field_size = 0) 000 not indicated not indicated 101 = 24-bit 20-bit 100 = 23-bit 19-bit 010 = 22-bit 18-bit 110 = 21-bit 17-bit 001 = 20-bit 16-bit	3b000
field_size	[0]	R	Specifies the field size 0 = Maximum length 20-bit 1 = Maximum length 24-bit	0

53.10.8.16 SPDIFIN_FRAME_PERIOD_1

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0048, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
frame_cnt_low	[7:0]	R	Specifies the frame count value[7:0] Frame_cnt register has 16-bit value. This is low by 8-bit. The period of a frame (two sub-frames) and register is updated every two sub-frames. It will be measured by "SPDIFIN_internal_clk" made with SPDIFIN_CONFIG.clk_divisor. Unit: SPDIF_internal_clk cycles The value it recommends for locking incoming signals: Over 0x220 (8.5 times × 64-bit)	0x00

53.10.8.17 SPDIFIN_FRAME_PERIOD_2

- Base Address: 0x12D3_0000
- Address = Base Address + 0x004C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
frame_cnt_high	[7:0]	R	Specifies the frame count value[15:8] Frame_cnt register has 16-bit value. This is high by 8-bits. The period of a frame (two sub-frames) and register is updated every two sub-frames. It is measured by "SPDIFIN_internal_clk" made with SPDIFIN_CONFIG.clk_divisor. Unit: SPDIF_internal_clk cycles The value it recommends for locking incoming signals: Over 0x220 (8.5 times × 64-bit)	0x0

53.10.8.18 SPDIFIN_Pc_INFO_1

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0050, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
error_flag	[7]	R	0 = Valid burst payload 1 = Burst payload may contain errors	0
RSVD	[6:5]	R	Reserved	2b00
compressed_data_type	[4:0]	R	0d = Null data 1d = Dolby AC-3 2d = Reserved 3d = Pause 4d = MPEG-1 layer 1 5d = MPEG-1 layer 2 or 3 or MPEG-2 w/o extension 6d = MPEG-2 w/ extension 7d = Reserved 8d = MPEG-2 layer 1 low sampling freq. 9d = MPEG-2 layer 2 or 3 low sampling freq. 10d = Reserved 11d, 12d, 13d = DTS 14d to 31d = Reserved	5b00000

53.10.8.19 SPDIFIN_Pc_INFO_2

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0054, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
bit_stream_number	[7:5]	R	Specifies the bit stream number	3b000
data_type_dependent_ info	[4:0]	R	Specifies the data type dependent information	5b00000

53.10.8.20 SPDIFIN_Pd_INFO_1

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0058, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
burst_payload_length_ low	[7:0]	R	Specifies the length of burst pay load[7:0] (Unit: bits)	0x00

53.10.8.21 SPDIFIN_Pd_INFO_2

- Base Address: 0x12D3_0000
- Address = Base Address + 0x005C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
burst_payload_length_ high	[7:0]	R	Specifies the length of burst pay load[15:8] (Unit: bits)	0x00

53.10.8.22 SPDIFIN_DATA_BUF_0_1/2/3

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0060, + 0x0064, + 0x0068, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
received_data_x	[7:0]	R	Specifies the PCM or stream data for first burst of HDMI SPDIFIN_DATA_BUF_0_1 = SPDIFIN_DATA_BUF_0[7:0] SPDIFIN_DATA_BUF_0_2 = SPDIFIN_DATA_BUF_0[15:8] SPDIFIN_DATA_BUF_0_3 = SPDIFIN_DATA_BUF_0[23:16] If SPDIFIN_CONFIG.data_align is "0" for 16-bit, Received_data is equal to {data_ (N) th, data_ (N + 1) th}. If SPDIFIN_CONFIG.data_align is "1" for 32-bit, Received_data is equal to {U, V, C, P, zero-padding, and data [n: 0]}. If SPDIFIN_CONFIG.U_V_P_report is "0", received_data is equal to {zero-padding, data [n: 0]}. where, "n" is dependent on SPDIFIN_CH_STATUS_1 and word_length if SPDIFIN_CONFIG. Data_type is 0 for PCM. "n" is equal to 15 if SPDIFIN_CONFIG.data_type is 1 for stream. If SPDIFIN_CONFIG.HDMI_burst_size is 1 burst, HDMI accesses only this data register.	0x00

53.10.8.23 SPDIFIN_USER_BUF_0

- Base Address: 0x12D3_0000
- Address = Base Address + 0x006C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
received_data_user_0	[7:4]	R	Specifies the user bit of first burst of HDMI received_data [7:4] = SPDIFIN_DATA_BUF_0 [31:28].	0x0
RSVD	[3:0]	R	Reserved	0x0

53.10.8.24 SPDIFIN_DATA_BUF_1_1/2/3

- Base Address: 0x12D3_0000
- Address = Base Address + 0x0070, + 0x0074, + 0x0078, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
received_data_data_x	[7:0]	R	Specifies the PCM or stream data for second burst of HDMI SPDIFIN_DATA_BUF_0_1 = SPDIFIN_DATA_BUF_0[7:0] SPDIFIN_DATA_BUF_0_2 = SPDIFIN_DATA_BUF_0[15:8] SPDIFIN_DATA_BUF_0_3 = SPDIFIN_DATA_BUF_0[23:16] If SPDIFIN_CONFIG.data_align is "0" for 16-bit, received_data is equal to {data_(N) th, data_(N + 1) th}. If SPDIFIN_CONFIG.data_align is "1" for 32-bit, received_data is equal to {U, V, C, P, zero-padding, data [n: 0]}. If SPDIFIN_CONFIG.U_V_P_report is "0", received_data is equal to {zero-padding, data [n: 0]}. , where "n" is dependent on SPDIFIN_CH_STATUS_1 and word_length if SPDIFIN_CONFIG. data_type is "0" for PCM. "n" is equal to 15 if SPDIFIN_CONFIG.data_type is "1" for stream. If SPDIFIN_CONFIG.HDMI_burst_size is 1 burst, HDMI accesses only this data register.	0x00

53.10.8.25 SPDIFIN_USER_BUF_1

- Base Address: 0x12D3_0000
- Address = Base Address + 0x007C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
received_data_user_1	[7:4]	R	Specifies the user bit of second burst of HDMI received_data[7:4] = SPDIFIN_DATA_BUF_1[31:28]	0x0
RSVD	[3:0]	R	Reserved	0x0

53.10.9 I2S Registers

53.10.9.1 I2S_CLK_CON

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0
i2s_en	[0]	RW	Enables the I2S clock. 0 = Disables I2S (default) 1 = Activates I2S will be activated Sets i2s_en after other registers are configured. If you want to reset I2S, this register is 0 → 1.	0

53.10.9.2 I2S_CON_1

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
r_sc_pol	[1]	RW	Specifies the SDATA is synchronous to 0 = SCLK falling edge 1 = SCLK rising edge	0
r_ch_pol	[0]	RW	Specifies the LRCLK polarity 0 = Left channel for low polarity 1 = Left channel for high polarity	0

53.10.9.3 I2S_CON_2

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0008, Reset Value = 0x16

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	0
mlsb	[6]	RW	0 = MSB first mode 1 = LSB first mode	0
bit_ch	[5:4]	RW	Specifies the bit clock per frame (Frame = left + right) 2b00 = 32 fs 2b01 = 48 fs 2b10 = 64 fs	2b01
data_num	[3:2]	RW	Specifies the serial data bit per channel 2b01 = 16-bit 2b10 = 20-bit 2b11 = 24-bit	2b01
i2s_mode	[1:0]	RW	2b00 = I2S basic format 2b10 = Left justified format 2b11 = Right justified format	2b10

53.10.9.4 I2S_PIN_SEL_0

- Base Address: 0x12D4_0000
- Address = Base Address + 0x000C, Reset Value = 0x77

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	0
pin_sel_1	[6:4]	RW	Selects the SCLK (I2S) 3b111 = i_i2s_in[1] 3b110 = i_i2s_in[6] 3b101 = i_i2s_in[5] 3b100 = i_i2s_in[4] 3b011 = i_i2s_in[3] 3b010 = i_i2s_in[2] 3b001 = i_i2s_in[1] 3b000 = i_i2s_in[0] NOTE: SCLK is selected with i_i2s_in [5] (0x101).	3b111
RSVD	[3]	RW	Reserved	0
pin_sel_0	[2:0]	RW	Selects the LRCK (I2S) 3b111 = i_i2s_in[0] 3b110 = i_i2s_in[6] 3b101 = i_i2s_in[5] 3b100 = i_i2s_in[4] 3b011 = i_i2s_in[3] 3b010 = i_i2s_in[2] 3b001 = i_i2s_in[1] 3b000 = i_i2s_in[0] NOTE: LRCK is selected with i_i2s_in [6] (0x110).	3b111

53.10.9.5 I2S_PIN_SEL_1

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0010, Reset Value = 0x77

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	0
pin_sel_3	[6:4]	RW	Selects the SDATA_1 (I2S) 3b111 = i_i2s_in[3] 3b110 = i_i2s_in[6] 3b101 = i_i2s_in[5] 3b100 = i_i2s_in[4] 3b011 = i_i2s_in[3] 3b010 = i_i2s_in[2] 3b001 = i_i2s_in[1] 3b000 = i_i2s_in[0] NOTE: SDATA_1 is selected with i_i2s_in [3] (0x011).	3b111
RSVD	[3]	RW	Reserved	0
pin_sel_2	[2:0]	RW	Selects the SDATA_0 (I2S) 3b111 = i_i2s_in[2] 3b110 = i_i2s_in[6] 3b101 = i_i2s_in[5] 3b100 = i_i2s_in[4] 3b011 = i_i2s_in[3] 3b010 = i_i2s_in[2] 3b001 = i_i2s_in[1] 3b000 = i_i2s_in[0] NOTE: SDATA_0 is selected with i_i2s_in [4] (0x100).	3b111

53.10.9.6 I2S_PIN_SEL_2

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0014, Reset Value = 0x77

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	0
pin_sel_5	[6:4]	RW	Selects the SDATA_3 (I2S) 3b111 = i_i2s_in[5] 3b110 = i_i2s_in[6] 3b101 = i_i2s_in[5] 3b100 = i_i2s_in[4] 3b011 = i_i2s_in[3] 3b010 = i_i2s_in[2] 3b001 = i_i2s_in[1] 3b000 = i_i2s_in[0] NOTE: SDATA_3 is selected with i_i2s_in [1] (0x001).	3b111
RSVD	[3]	RW	Reserved	0
pin_sel_4	[2:0]	RW	Selects the SDATA_2 (I2S) 3b111 = i_i2s_in[4] 3b110 = i_i2s_in[6] 3b101 = i_i2s_in[5] 3b100 = i_i2s_in[4] 3b011 = i_i2s_in[3] 3b010 = i_i2s_in[2] 3b001 = i_i2s_in[1] 3b000 = i_i2s_in[0] NOTE: SDATA_2 is selected with i_i2s_in [2] (0x010).	3b111

53.10.9.7 I2S_PIN_SEL_3

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0018, Reset Value = 0x07

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0
pin_sel_6	[2:0]	RW	Selects the DSD_D5 (DSD) 3b111 = i_i2s_in[6] 3b110 = i_i2s_in[6] 3b101 = i_i2s_in[5] 3b100 = i_i2s_in[4] 3b011 = i_i2s_in[3] 3b010 = i_i2s_in[2] 3b001 = i_i2s_in[1] 3b000 = i_i2s_in[0] NOTE: DSD_D5 is selected with i_i2s_in [0] (0x000).	3b111

53.10.9.8 I2S_DSD_CON

- Base Address: 0x12D4_0000
- Address = Base Address + 0x001C, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	0
r_dsd_pol	[1]	RW	1 = DSD_DATA changes at DSD_CLK rising edge 0 = DSD_DATA changes at DSD_CLK falling edge	1
dsd_en	[0]	RW	1 = Enables DSD module 0 = Disables DSD module	0

53.10.9.9 I2S_IN_MUX_CON

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0020, Reset Value = 0x60

Name	Bit	Type	Description	Reset Value
f_num	[7:5]	RW	Specifies the number of stage of noise filter for I2S input pins 000 = no filtering 001 = 2-stage filter 010 = 3-stage filter 011 = 4-stage filter 100 = 5-stage filter Others = Reserved	3b011
in_en	[4]	RW	Enables i2s_in, which is a sub-module at the input stage 0 = Disables i2s_in module 1 = Enables i2s_in module If disabled, all output data is "0".	0
audio_sel	[3:2]	RW	Selects the audio 2b00 = Enables SPDIF audio data 2b01 = Enables I2S audio data 2b10 = Enables DSD audio data	0
CUV_sel	[1]	RW	Selects the CUV 0 = Enables SPDIF CUV data 1 = Enables I2S CUV data	0
mux_en	[0]	RW	Enables i2s_mux, which is a sub-module for audio selection 0 = Disables i2s_mux module 1 = Enables i2s_mux module If disabled, all output data is "0".	0

53.10.9.10 I2S_CH_ST_CON

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0024, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0
channel_status_reload	[0]	RW	0 = Updates the shadow channel status registers 1 = Sets this bit to update the shadow channel status registers with The values it updates in I2S_CH_ST_0 to I2S_CH_ST_4. This bit is cleared if the shadow channel status registers are updated.	0

Channel status information needs to be applied to the audio stream at the IEC 60958 block boundary. For this synchronization, there are two register sets for channel status block. You can set the channel status registers, I2S_CH_ST_0 to I2S_CH_ST_4, while the I2S Rx module still refers to the shadow channel status registers, I2S_CH_ST_SH_0 to I2S_CH_ST_CH4.

To reflect the user configuration in the channel status registers, set "channel_status_reload" bit in I2S_CH_ST_CON. I2S Rx module then copy the channel status registers into the shadow channel status registers at the beginning of an IEC-60958 block.

53.10.9.11 I2S_CH_ST_0

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0028, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
channel_status_mode	[7:6]	RW	2b00 = Mode 0 Others = Reserved	0
emphasis	[5:3]	RW	If bit1 = 0, 3b000 = 2 audio channels without pre-emphasis × 3b001 = 2 audio channels with 50 μs/15 μs pre-emphasis If bit1 = 1, 3b000 = default state	0
copyright	[2]	RW	0 = Copyright 1 = No copyright	0
audio_sample_word	[1]	RW	0 = linear PCM 1 = Non-linear PCM	0
channel_status_block	[0]	RW	0 = Consumer format 1 = Professional format	0

The bits listed here in channel status registers look swapped with those in IEC-60958-3 specification, as the bit order is different (LSB is right-most bit).

53.10.9.12 I2S_CH_ST_1

- Base Address: 0x12D4_0000
- Address = Base Address + 0x002C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
category	[7:0]	RW	Specifies the equipment type CD player: 0000_0001 DAT player: L000_0011 DCC player: L100_0011 Mini disc: L100_1001 (L: information about generation status of the material)	0

53.10.9.13 I2S_CH_ST_2

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0030, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
channel_number	[7:4]	RW	Specifies the channel number NOTE: bit [4] is LSB.	0
source_number	[3:0]	RW	Specifies the source number NOTE: bit [0] is LSB.	0

53.10.9.14 I2S_CH_ST_3

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0034, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	2b00
Clock_Accuracy	[5:4]	RW	Specifies the clock accuracy, as specified in IEC-60958-3 2b01 = Level I, ± 50 ppm 2b00 = Level II, ± 1000 ppm 2b10 = Level III, variable pitch shifted	2b00
Sampling_Frequency	[3:0]	RW	Specifies the sampling frequency, as specified in IEC-60958-3 4b0000 = 44.1 kHz 4b0010 = 48 kHz 4b0011 = 32 kHz 4b1010 = 96 kHz …	0

53.10.9.15 I2S_CH_ST_4

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0038, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Org_Sampling_Freq	[7:4]	RW	Specifies the original sampling frequency 4b1111 = 44.1 kHz 4b0111 = 88.2 kHz 4b1011 = 22.05 kHz 4b0011 = 176.4 kHz … Refer to original sampling frequency specified in IEC-60958-3 for other frequencies.	0x0
Word_Length	[3:1]	RW	Specifies the word length Maximum length 24-bit 20-bit 3b000 = not defined not defined 3b001 = 20-bit 16-bit 3b010 = 22-bit 18-bit 3b100 = 23-bit 19-bit 3b101 = 24-bit 20-bit 3b110 = 21-bit 17-bit	3b000
Max_Word_Length	[0]	RW	Specifies the maximum sample word length. 1 = 24-bit 0 = 20-bit	0

53.10.9.16 I2S_CH_ST_SH_0

- Base Address: 0x12D4_0000
- Address = Base Address + 0x003C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
channel_status_mode	[7:6]	R	2b00 = Mode 0 Others = Reserved	0
emphasis	[5:3]	R	If bit1 = 0, 3b000 = 2 audio channels without pre-emphasis × 3b001 = 2 audio channels with 50 μs/15 μs pre-emphasis If bit1 = 1, 3b000 = default state	0
copyright	[2]	R	0 = Copyright 1 = No copyright	0
audio_sample_word	[1]	R	0 = linear PCM 1 = Non-linear PCM	0
channel_status_block	[0]	R	0 = Consumer format 1 = Professional format	0

The bits listed here in channel status registers look swapped with those in IEC-60958-3 specification, as the bit order is different (LSB is right-most bit).

53.10.9.17 I2S_CH_ST_SH_1

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0040, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
category	[7:0]	R	Specifies the equipment type CD player: 0000_0001 DAT player: L000_0011 DCC player: L100_0011 Mini disc: L100_1001 (L: information about generation status of material)	0

53.10.9.18 I2S_CH_ST_SH_2

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0044, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
channel_number	[7:4]	R	Specifies the channel number NOTE: bit [4] is LSB.	0
source_number	[3:0]	R	Specifies source number NOTE: bit [0] is LSB.	0

53.10.9.19 I2S_CH_ST_SH_3

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0048, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	R	Reserved	2b00
Clock_Accuracy	[5:4]	R	Specifies the clock accuracy, as specified in IEC-60958-3 2b01 = Level I, ± 50 ppm 2b00 = Level II, ± 1000 ppm 2b10 = Level III, variable pitch shifted	2b00
Sampling_Frequency	[3:0]	R	Specifies the sampling frequency, as specified in IEC-60958-3 4b0000 = 44.1 kHz 4b0010 = 48 kHz 4b0011 = 32 kHz 4b1010 = 96 kHz …	0

53.10.9.20 I2S_CH_ST_SH_4

- Base Address: 0x12D4_0000
- Address = Base Address + 0x004C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Org_Sampling_Freq	[7:4]	R	Specifies the original sampling frequency 4b1111 = 44.1 kHz 4b0111 = 88.2 kHz 4b1011 = 22.05 kHz 4b0011 = 176.4 kHz … Refer to original sampling frequency specified in IEC-60958-3 for other frequencies.	0x0
Word_Length	[3:1]	R	Specifies the word length Maximum. length 24-bit 20-bit 3b000 = not defined not defined 3b001 = 20-bit 16-bit 3b010 = 22-bit 18-bit 3b100 = 23-bit 19-bit 3b101 = 24-bit 20-bit 3b110 = 21-bit 17-bit	3b000
Max_Word_Length	[0]	R	Specifies the maximum sample word length 1 = 24-bit 0 = 20-bit	0

53.10.9.21 I2S_VD_DATA

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0050, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	7b0000000
validity_flag	[0]	RW	Specifies the validity bit 0 = Audio sample is reliable 1 = Audio sample is unreliable	0

53.10.9.22 I2S_MUX_CH

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0054, Reset Value = 0x03

Name	Bit	Type	Description	Reset Value
CH3_R_en	[7]	RW	0 = Disables channel 3 right audio data output 1 = Enables channel 3 right audio data output	0
CH3_L_en	[6]	RW	0 = Disables channel 3 left audio data output 1 = Enables channel 3 left audio data output	0
CH2_R_en	[5]	RW	0 = Disables channel 2 right audio data output 1 = Enables channel 2 right audio data output	0
CH2_L_en	[4]	RW	0 = Disables channel 2 left audio data output 1 = Enables channel 2 left audio data output	0
CH1_R_en	[3]	RW	0 = Disables channel 1 right audio data output 1 = Enables channel 1 right audio data output	0
CH1_L_en	[2]	RW	0 = Disables channel 1 left audio data output 1 = Enables channel 1 left audio data output	0
CH0_R_en	[1]	RW	0 = Disables channel 0 right audio data output 1 = Enables channel 0 right audio data output	1
CH0_L_en	[0]	RW	0 = Disables channel 0 left audio data output 1 = Enables channel 0 left audio data output	1

53.10.9.23 I2S_MUX_CUV

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0058, Reset Value = 0x03

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	6b000000
CUV_R_en	[1]	RW	0 = Disables right channel CUV data 1 = Enables right channel CUV data	1
CUV_L_en	[0]	RW	0 = Disables left channel CUV data 1 = Enables left channel CUV data	1

53.10.9.24 I2S_CHX_Y_Z

- Base Address: 0x12D4_0000
- Address = Base Address + 0x0064, + 0x0068, + 0x006C,
  + 0x0070, + 0x0074, + 0x0078, + 0x007C,
  + 0x0080, + 0x0084, + 0x0088, + 0x008C,
  + 0x0090, + 0x0094, + 0x0098, + 0x009C,
  + 0x00A0, + 0x00A4, + 0x00A8, + 0x00AC,
  + 0x00B0, + 0x00B4, + 0x00B8, + 0x00BC,
  + 0x00C0, + 0x00C4, + 0x00C8, + 0x00CC,
  + 0x00D0, + 0x00D4, + 0x00D8, + 0x00DC, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	Specifies the PCM output data from I2S Rx module X = Channel = 0, 1, 2 Y = Left/Right = L, R Z = Byte number I2S_CHX_Y_0 = PCMX_Y[7:0] I2S_CHX_Y_1 = PCMX_Y[15:8] I2S_CHX_Y_2 = PCMX_Y[23:16] I2S_CHX_Y_3 = PCMX_Y[27:24] Channel 3 has 24-bit width. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	0x00

53.10.9.25 I2S_CUV_L_R

- Base Address: 0x12D4_0000
- Address = Base Address + 0x00DC, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	R	Reserved	0
CUV_R	[6:4]	R	Specifies the VUCP data of right channel CUV_R[3:0] = {Valid bit, User bit, Channel state bit, Parity bit}	3b000
RSVD	[3]	R	Reserved	0
CUV_L	[2:0]	R	Specifies the VUCP data of left channel CUV_L[3:0] = {Valid bit, User bit, Channel state bit, Parity bit}	3b000

53.10.10 Timing Generator Register

53.10.10.1 TG_CMD

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	000
getsync_type	[4]	RW	Specifies the timing correction enable bit If this bit is set, the input VSYNC timing error relative to output VSYNC is corrected. 0 = Disables correction bit 1 = Enables correction bit	0
getsync_en	[3]	RW	Enables BT656 input synchronization.	0
Reserved	[2]	RW	Reserved	0
field_en	[1]	RW	Enables field mode. For 1080i, this should be enabled.	0
tg_en	[0]	RW	Specifies the TG global enable bit	0

53.10.10.2 TG_H_FSZ_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0018, Reset Value = 0x72

Name	Bit	Type	Description	Reset Value
TG_H_FSZ_L	[7:0]	RW	Specifies the horizontal full size (1 to 8191) (Lower part)	0x72

53.10.10.3 TG_H_FSZ_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x001C, Reset Value = 0x06

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	0x0
TG_H_FSZ_H	[4:0]	RW	Specifies the horizontal full size (1 to 8191) (Upper part)	0x6

53.10.10.4 TG_HACT_ST_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0020, Reset Value = 0x05

Name	Bit	Type	Description	Reset Value
TG_HACT_ST_L	[7:0]	RW	Specifies the horizontal active start position (1 to 4095) (Lower part)	0x05

53.10.10.5 TG_HACT_ST_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0024, Reset Value = 0x01

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	0x0
TG_HACT_ST_H	[3:0]	RW	Specifies the horizontal active start position (1 to 4095) (Upper part)	0x1

53.10.10.6 TG_HACT_SZ_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0028, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
TG_HACT_SZ_L	[7:0]	RW	Specifies the horizontal active size (0 to 4095) (Lower part)	0x00

53.10.10.7 TG_HACT_SZ_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x002C, Reset Value = 0x05

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	0x0
TG_HACT_SZ_H	[3:0]	RW	Specifies the horizontal active size (0 to 4095) (Upper part)	0x5

53.10.10.8 TG_V_FSZ_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0030, Reset Value = 0xEE

Name	Bit	Type	Description	Reset Value
TG_V_FSZ_L	[7:0]	RW	Specifies the vertical full size (1 to 2047) (Lower part)	0xEE

53.10.10.9 TG_V_FSZ_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0034, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_V_FSZ_H	[2:0]	RW	Specifies the vertical full size (1 to 2047) (Upper part)	0x2

53.10.10.10 TG_VSYNC_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0038, Reset Value = 0x01

Name	Bit	Type	Description	Reset Value
TG_VSYNC_L	[7:0]	RW	Specifies the vertical sync position When field enable is set, this bit takes the top field Vsync position (1 to 2047) (Lower part).	0x01

53.10.10.11 TG_VSYNC_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x003C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_VSYNC_H	[2:0]	RW	Specifies the vertical sync position When field enable is set, this bit takes the top field Vsync position (1 to 2047) (Upper part).	0x0

53.10.10.12 TG_VSYNC2_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0040, Reset Value = 0x33

Name	Bit	Type	Description	Reset Value
TG_VSYNC2_L	[7:0]	RW	Specifies the vertical sync position for bottom field (1 to 2047) (Lower part)	0x33

53.10.10.13 TG_VSYNC2_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0044, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_VSYNC2_H	[2:0]	RW	Specifies the vertical sync position for bottom field (1 to 2047) (Upper part)	0x2

53.10.10.14 TG_VACT_ST_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0048, Reset Value = 0x1A

Name	Bit	Type	Description	Reset Value
TG_VACT_ST_L	[7:0]	RW	Specifies the vertical active start position (1 to 2047) (Lower part).	0x1a

53.10.10.15 TG_TACT_ST_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x004C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_VACT_ST_H	[2:0]	RW	Specifies the vertical active start position (1 to 2047) (Upper part)	0x0

53.10.10.16 TG_VACT_SZ_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0050, Reset Value = 0xD0

Name	Bit	Type	Description	Reset Value
TG_VACT_SZ_L	[7:0]	RW	Specifies the vertical active size (0 to 2047) (Lower part)	0xD0

53.10.10.17 TG_TACT_SZ_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0054, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_VACT_SZ_H	[2:0]	RW	Specifies the vertical active size (0 to 2047) (Upper part)	0x2

53.10.10.18 TG_FIELD_CHG_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0058, Reset Value = 0x33

Name	Bit	Type	Description	Reset Value
TG_FIELD_CHG_L	[7:0]	RW	Specifies the HDMI field position. (Lower part)	0x33

53.10.10.19 TG_FIELD_CHG_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x005C, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_FIELD_CHG_H	[2:0]	RW	Specifies the HDMI field position. (Upper part)	0x2

53.10.10.20 TG_VACT_ST2_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0060, Reset Value = 0x48

Name	Bit	Type	Description	Reset Value
TG_VACT_ST2_L	[7:0]	RW	Specifies the HDMI vertical active start position for bottom field (Lower part).	0x48

53.10.10.21 TG_VACT_ST2_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0064, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_VACT_ST2_H	[2:0]	RW	Specifies the HDMI vertical active start position for bottom field (Upper part).	0x2

53.10.10.22 TG_VACT_ST3_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0068, Reset Value = 0x7B

Name	Bit	Type	Description	Reset Value
TG_VACT_ST3_L	[7:0]	RW	Specifies the third HDMI vertical active start position for bottom field (Lower part).	0x7B

53.10.10.23 TG_VACT_ST3_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x006C, Reset Value = 0x04

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_VACT_ST3_H	[2:0]	RW	Specifies the third HDMI vertical active start position for bottom field (Upper part).	0x4

53.10.10.24 TG_VACT_ST4_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0070, Reset Value = 0xAE

Name	Bit	Type	Description	Reset Value
TG_VACT_ST4_L	[7:0]	RW	Specifies the fourth HDMI vertical active start position for bottom field (Lower part).	0xAE

53.10.10.25 TG_VACT_ST4_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0074, Reset Value = 0x06

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_VACT_ST4_H	[2:0]	RW	Specifies the fourth HDMI vertical active start position for bottom field (Upper part).	0x6

53.10.10.26 TG_VSYNC_TOP_HDMI_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0078, Reset Value = 0x01

Name	Bit	Type	Description	Reset Value
TG_VSYNC_TOP_HDMI_L	[7:0]	RW	Specifies the HDMI Vsync position for top field (Lower part).	0x01

53.10.10.27 TG_VSYNC_TOP_HDMI_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x007C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_VSYNC_TOP_HDMI_H	[2:0]	RW	Specifies the HDMI Vsync position for top field (Upper part).	0x0

53.10.10.28 TG_VSYNC_BOT_HDMI_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0080, Reset Value = 0x33

Name	Bit	Type	Description	Reset Value
TG_VSYNC_BOT_ HDMI_L	[7:0]	RW	Specifies the HDMI VSYNC position for bottom field (Lower part).	0x33

53.10.10.29 TG_VSYNC_BOT_HDMI_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0084, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_VSYNC_BOT_ HDMI_H	[2:0]	RW	Specifies the HDMI VSYNC position for bottom field (Upper part).	0x2

53.10.10.30 TG_FIELD_TOP_HDMI_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0088, Reset Value = 0x01

Name	Bit	Type	Description	Reset Value
TG_FIELD_TOP_ HDMI_L	[7:0]	RW	Specifies the HDMI top field start position (Lower part).	0x01

53.10.10.31 TG_FIELD_TOP_HDMI_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x008C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_FIELD_TOP_ HDMI_H	[2:0]	RW	Specifies the HDMI top field start position (Upper part).	0x0

53.10.10.32 TG_FIELD_BOT_HDMI_L

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0090, Reset Value = 0x33

Name	Bit	Type	Description	Reset Value
TG_FIELD_BOT_ HDMI_L	[7:0]	RW	Specifies the HDMI bottom field start position (Lower part).	0x33

53.10.10.33 TG_FIELD_BOT_HDMI_H

- Base Address: 0x12D5_0000
- Address = Base Address + 0x0094, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	0x0
TG_FIELD_BOT_ HDMI_H	[2:0]	RW	Specifies the HDMI bottom field start position (Upper part).	0x2

53.10.10.34 TG_3D

- Base Address: 0x12D5_0000
- Address = Base Address + 0x00F0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	0x0
TG_3D	[0]	RW	Specifies the Stereoscopy timing generation	0

53.10.11 HDCP eFUSE Registers

53.10.11.1 HDCP_E_FUSE_CTRL

- Base Address: 0x12D6_0000
- Address = Base Address + 0x0000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	4b0000
HDCP_KEY_READ	[0]	RW	0 = Normal 1 = To read HDCP key from e-fuse.	0

53.10.11.2 HDCP_E_FUSE_STATUS

- Base Address: 0x12D6_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	R	Reserved	4b0000
EFUSE_ECC_FAIL	[2]	R	0 = Normal 1 = ECC fail	–
EFUSE_ECC_BUSY	[1]	R	0 = Not busy 1 = Busy	–
EFUSE_ECC_DONE	[0]	R	0 = Normal 1 = ECC done	0

53.10.11.3 EFUSE_ADDR_WIDTH

- Base Address: 0x12D6_0000
- Address = Base Address + 0x0008, Reset Value = 0x14

Name	Bit	Type	Description	Reset Value
EFUSE_ADDR_WIDTH	[7:0]	RW	Specifies the address width (Unit: HDMI link PCLK, default: 83 MHz, 12n).	0x14

53.10.11.4 EFUSE_SIGDEV_ASSERT

- Base Address: 0x12D6_0000
- Address = Base Address + 0x000C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
EFUSE_SIGDEV_ASSERT	[7:0]	RW	Specifies the SIGDEV asserting position (Unit: HDMI Link PCLK, default: 83 MHz, 12n).	0x0

53.10.11.5 EFUSE_SIGDEV_DE_ASSERT

- Base Address: 0x12D6_0000
- Address = Base Address + 0x0010, Reset Value = 0x08

Name	Bit	Type	Description	Reset Value
EFUSE_SIGDEV_DEASSERT	[7:0]	RW	Specifies the SIGDEV de-asserting position (Unit: HDMI Link PCLK, default: 83 MHz, 12n)	0x8

53.10.11.6 EFUSE_PRCHG_ASSERT

- Base Address: 0x12D6_0000
- Address = Base Address + 0x0014, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
EFUSE_PRCHG_ASSERT	[7:0]	RW	Specifies the PRCHG asserting position (Unit: HDMI Link PCLK, default: 83 MHz, 12n)	0x0

53.10.11.7 EFUSE_PRCHG_DE_ASSERT

- Base Address: 0x12D6_0000
- Address = Base Address + 0x0018, Reset Value = 0x0C

Name	Bit	Type	Description	Reset Value
EFUSE_PRCHG_DEASSERT	[7:0]	RW	Specifies the PRCHG de-asserting position (Unit: HDMI link PCLK, default: 83 MHz, 12n).	0xC

53.10.11.8 EFUSE_FSET_ASSERT

- Base Address: 0x12D6_0000
- Address = Base Address + 0x001C, Reset Value = 0x04

Name	Bit	Type	Description	Reset Value
EFUSE_FSET_ASSERT	[7:0]	RW	Specifies the FSET asserting position (Unit: HDMI link PCLK, default: 83 MHz, 12n)	0x4

53.10.11.9 EFUSE_FSET_DE_ASSERT

- Base Address: 0x12D6_0000
- Address = Base Address + 0x0020, Reset Value = 0x10

Name	Bit	Type	Description	Reset Value
EFUSE_FSET_DEASSERT	[7:0]	RW	Specifies the FSET de-asserting position (Unit: HDMI link PCLK, default: 83 MHz, 12n)	0x10

53.10.11.10 EFUSE_SENSING

- Base Address: 0x12D6_0000
- Address = Base Address + 0x0024, Reset Value = 0x14

Name	Bit	Type	Description	Reset Value
EFUSE_SENSING	[7:0]	RW	Specifies the sensing width (Unit: HDMI link PCLK, default: 83 MHz, 12n).	0x14

53.10.11.11 EFUSE_SCK_ASSERT

- Base Address: 0x12D6_0000
- Address = Base Address + 0x0028, Reset Value = 0x04

Name	Bit	Type	Description	Reset Value
EFUSE_SCK_ASSERT	[7:0]	RW	Specifies the SCK asserting position (Unit: HDMI link PCLK, default: 83 MHz, 12n)	0x4

53.10.11.12 EFUSE_SCK_DE_ASSERT

- Base Address: 0x12D6_0000
- Address = Base Address + 0x002C, Reset Value = 0x0C

Name	Bit	Type	Description	Reset Value
EFUSE_SCK_DEASSERT	[7:0]	RW	Specifies the SCK de-asserting position (Unit: HDMI link PCLK, default: 83 MHz, 12n)	0xC

53.10.11.13 EFUSE_SDOUT_OFFSET

- Base Address: 0x12D6_0000
- Address = Base Address + 0x0030, Reset Value = 0x10

Name	Bit	Type	Description	Reset Value
EFUSE_SDOUT_OFFSET	[7:0]	RW	Specifies the SDOUT offset (Unit: HDMI link PCLK, default: 83 MHz, 12n)	0x10

53.10.11.14 EFUSE_READ_OFFSET

- Base Address: 0x12D6_0000
- Address = Base Address + 0x0034, Reset Value = 0x14

Name	Bit	Type	Description	Reset Value
EFUSE_READ_OFFSET	[7:0]	RW	Specifies the READ Offset (Unit: HDMI link PCLK, default: 83 MHz, 12n).	0x14

53.10.12 CEC Configure Registers

53.10.12.1 CEC_TX_STATUS_0

- Base Address: 0x100B_0000
- Address = Base Address + 0x0000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	R	Reserved	4b0000
Tx_Error	[3]	R	Specifies the CEC Tx_Error interrupt flag This bit field also specifies the status of Tx_Error interrupt and is valid only if Tx_Done bit is set. 0 = No error occurs 1 = An error occurs during CEC Tx transfer It will be cleared • If set to 0 by Tx_Enable bit of CEC_TX_CTRL register • If set Clear_Intr_Tx_Done or Clear_Intr_Tx_Error bit in CEC_INTR_CLEAR register	0
Tx_Done	[2]	R	Specifies the CEC Tx_Done interrupt flag This bit field also specifies the status of Tx_Done interrupt. 0 = Running or idle 1 = Finishes CEC Tx transfer It will be cleared • If Tx_Enable bit of CEC_TX_CTRL_0 is reset • If Clear_Intr_Tx_Done or Clear_Intr_Tx_Error bit in CEC_INTR_CLEAR register is set	0
Tx_Transferring	[1]	R	If TX-Running is set, this field is valid. 0 = Tx waits for the CEC Bus 1 = CEC Tx transfers data through CEC Bus	0
Tx_Running	[0]	R	0 = Tx Idle 1 = Enables CEC Tx, and waits for the CEC bus or transfers the message.	0

53.10.12.2 CEC_TX_STATUS_1

- Base Address: 0x100B_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Tx_Bytes_Transferred	[7:0]	R	Specifies the number of blocks transferred (1 byte = 1 block in a CEC message). After sending the CEC message, this field will be updated. It will be cleared if Clear_Intr_Tx_Done or Clear_Intr_Tx_Error bit is set in CEC_Intr_Clear register.	0

53.10.12.3 CEC_RX_STATUS_0

- Base Address: 0x100B_0000
- Address = Base Address + 0x0008, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	R	Reserved	3b000
Rx_BCast	[4]	R	Specifies the broadcast message flag 0 = Receives CEC message, it is the address to a single device 1 = Receives CEC message, it is the broadcast message It will be cleared • if Rx_Enable bit of CEC_RX_CTRL_0 is reset • if Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register is set	0
Rx_Error	[3]	R	Specifies the CEC Rx_Error interrupt flag This bit field also specifies the status of Rx_Error interrupt and is valid only if Rx_Done bit is set. 0 = No error occurs 1 = An error occurs while receiving a CEC message It will be cleared • if Rx_Enable bit of CEC_RX_CTRL_0 is reset • if Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register is set	0
Rx_Done	[2]	R	Specifies the CEC Rx done interrupt flag This bit field also specifies the status of Rx_Done interrupt. 0 = Running or Idle 1 = Finishes CEC Rx transfer It will be cleared: • if Rx_Enable bit of CEC_RX_CTRL_0 is reset • if Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register is set	0
Rx_Receiving	[1]	R	0 = Rx waits for a CEC message 1 = Rx receives data through CEC bus	0
Rx_Running	[0]	R	0 = Disables Rx 1 = Enables CEC Rx and waits for a message on the CEC bus.	0

53.10.12.4 CEC_RX_STATUS_1

- Base Address: 0x100B_0000
- Address = Base Address + 0x000C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Rx_Bytes_Received	[7:0]	R	Specifies the number of blocks received (1 byte = 1 block in a CEC message). After receiving the CEC message, the field will be updated. It will be cleared if Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_Intr_Clear register is set.	0

53.10.12.5 CEC_INTR_MASK

- Base Address: 0x100B_0000
- Address = Base Address + 0x0010, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	2b00
Mask_Intr_Rx_Error	[5]	RW	Specifies the Rx_Error interrupt mask bit 0 = Enables interrupt 1 = Disables interrupt	0
Mask_Intr_Rx_Done	[4]	RW	Specifies the Rx_Done interrupt mask bit 0 = Enables interrupt 1 = Disables interrupt	0
RSVD	[3:2]	RW	Reserved	2b00
Mask_Intr_Tx_Error	[1]	RW	Specifies the Tx_Error interrupt mask bit 0 = Enables interrupt 1 = Disables interrupt	0
Mask_Intr_Tx_Done	[0]	RW	Specifies the Tx_Done interrupt mask bit 0 = Enables interrupt 1 = Disables interrupt	0

53.10.12.6 CEC_INTR_CLEAR

- Base Address: 0x100B_0000
- Address = Base Address + 0x0014, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	2b00
Clear_Intr_Rx_Error	[5]	RW	Specifies the Rx_Error interrupt clear bit 0 = No effect 1 = Clears Rx_Error and Rx_Bytes_Received fields in CEC_RX_STATUS_0 and 1 registers. It will be cleared after one clock.	0
Clear_Intr_Rx_Done	[4]	RW	Specifies the Rx_Done interrupt clear bit 0 = No effect 1 = Clears Rx_Done and Rx_Bytes_Received fields in CEC_RX_STATUS_0 and 1 registers. Resets to 0 after one clock.	0
RSVD	[3:2]	RW	Reserved	2b00
Clear_Intr_Tx_Error	[1]	RW	Specifies the Tx_Error interrupt clear bit 0 = No effect 1 = Clears Tx_Error and Tx_Bytes_Received fields in CEC_TX_STATUS_0 and 1 registers. Resets to 0 after one clock.	0
Clear_Intr_Tx_Done	[0]	RW	Specifies the Tx_Done interrupt clear bit 0 = No effect 1 = Clears Tx_Done and Tx_Bytes_Received fields in CEC_TX_STATUS_0 and 1 registers. Resets to 0 after one clock.	0

53.10.12.7 CEC_LOGIC_ADDR

- Base Address: 0x100B_0000
- Address = Base Address + 0x0020, Reset Value = 0x0F

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	4b0000
Logic_Addr	[3:0]	RW	Specifies the HDMI Tx logical address (0 to 15)	4b1111

53.10.12.8 CEC_DIVISOR_X

- Base Address: 0x100B_0000
- Address = Base Address + 0x0030, + 0x0034, + 0x0038, + 0x003C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
CEC_Divisor	[7:0]	RW	Specifies the divisor used in counting 0.05ms period This divisor should satisfy the equation: (CEC_DIVISOR+1) × (clock cycle time(ns)) = 0.05 ms NOTE: To apply CEC_Divisor, it should be "0" for Tx_Reset and Rx_Reset, while Tx_Start and Rx_Start are "0".	0

53.10.12.9 CEC_TX_CTRL

- Base Address: 0x100B_0000
- Address = Base Address + 0x0040, Reset Value = 0x10

Name	Bit	Type	Description	Reset Value
Reset	[7]	RW	Specifies the CEC Tx reset bit 0 = No effect 1 = Immediately resets CEC Tx related registers and state machines to its reset value. Resets to 0 after one clock.	0
Tx_Retrans_Num	[6:4]	RW	Specifies the number of retransmissions tried (according to CEC specification on page CEC-13). On the basis of specification , this value should be set to 5.	3b001
RSVD	[3:2]	RW	Reserved	2b00
Tx_BCast	[1]	RW	Specifies the CEC Tx broadcast message bit This bit also specifies the CEC message in CEC_TX_BUFFER_00 to 15, which directly addresses (addresses to a single device) or broadcast. This bit determines whether a block transfer is acknowledged or not (according to ACK scheme in CEC specification (section CEC 6.1.2)) 0 = Directly addresses message 1 = Broadcast message	0
Tx_Start	[0]	RW	Specifies the CEC Tx start bit 0 = Tx idle 1 = Starts CEC message transfer (Resets to 0 after start)	0

53.10.12.10 CEC_TX_BYTE_NUM

- Base Address: 0x100B_0000
- Address = Base Address + 0x0044, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Tx_Byte_Num	[7:0]	RW	Specifies the number of blocks in a message that has to be sent (1 byte = 1 block in a CEC message).	0

53.10.12.11 CEC_TX_STATUS_2

- Base Address: 0x100B_0000
- Address = Base Address + 0x0060, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Tx_Wait	[7]	R	Specifies the CEC Tx signal free time waiting flag bit 0 = Tx is in other state 1 = CEC Tx waits for signal free time (stops sending messages after earlier attempts to send message).	0
Tx_Sending_Start_Bit	[6]	R	Specifies the CEC Tx start bit sending flag bit 0 = Tx is in other state 1 = CEC Tx sends a start bit	0
Tx_Sending_Hdr_Blk	[5]	R	Specifies the CEC Tx header block sending flag bit 0 = Tx is in other state 1 = CEC Tx sends the header block	0
Tx_Sending_Data_Blk	[4]	R	Specifies the CEC Tx data block sending flag bit 0 = Tx is in other state 1 = CEC Tx sends data blocks	0
Tx_Latest_Initiator	[3]	R	Specifies the CEC Tx last initiator flag bit 0 = This device is not the latest initiator on the CEC bus 1 = This CEC device is the latest initiator to send a CEC message. No other CEC device sends a message. It will be cleared if Rx detects a start bit on the CEC line or sets Tx_Enable bit of CEC_Tx_Ctrl_0 (that is becomes a new initiator)	0
RSVD	[2:0]	R	Reserved	3b000

53.10.12.12 CEC_TX_STATUS_3

- Base Address: 0x100B_0000
- Address = Base Address + 0x0064, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	R	Reserved	0
Tx_Wait_SFT_Succ	[6]	R	Specifies the CEC Tx signal free time for successive message transfer waiting flag bit. 0 = Tx is in other state 1 = Tx waits for signal free time (SFT) with a precondition that Tx is the most recent initiator on the CEC bus. It also sends another frame immediately after its previous frame (SFT ≥ 7 × 2.4 ms).	0
Tx_Wait_SFT_New	[5]	R	Specifies the CEC Tx signal free time for a new initiator waiting flag bit. 0 = Tx is in other state 1 = Tx waits for SFT with a precondition that Tx is the new initiator and wants to send a frame (SFT ≥ 5 × 2.4 ms).	0
Tx_Wait_SFT_Retran	[4]	R	Specifies the CEC Tx signal free time for a new initiator waiting flag bit. 0 = Tx is in other state 1 = Tx waits for SFT with a precondition (the precondition is that Tx should attempt to retransmit the message (SFT ≥ 3 × 2.4 ms))	0
Tx_Retrans_Cnt	[3:1]	R	Specifies the current retransmission count. If "0", no retransmission occurs. It will be cleared if Clear_Intr_Tx_Done or Clear_Intr_Tx_Error bit in CEC_Intr_Clear register is set.	3b000
Tx_ACK_Failed	[0]	R	Specifies the CEC Tx acknowledge failed flag bit 0 = Tx is in the other state 1 = Tx is not acknowledged. This bit is set if • ACK bit in a block is logical 1 in a directly addressed message • ACK bit in a block is logical 0 in a broadcast message	0

53.10.12.13 CEC_TX_BUFFER_X

- Base Address: 0x100B_0000
- Address = Base Address + 0x0080, + 0x0084, + 0x0088, + 0x08C,
  0x0090, + 0x0094, + 0x0098, + 0x009C,
  0x00A0, + 0x00A4, + 0x00A8, + 0x00AC,
  0x00B0, + 0x00B4, + 0x00B8, + 0x00BC, Reset Value = 0x10

Name	Bit	Type	Description	Reset Value
Tx_Block_0 to Tx_Block_15	[7:0]	RW	Specifies the byte #0 to #15 of CEC message Each byte corresponds to a block in a message. Block_0 is the header block and block_1 to15 are data blocks. NOTE: The initiator and destination logical address in a header block should be written by software.	0

53.10.12.14 CEC_RX_CTRL

- Base Address: 0x100B_0000
- Address = Base Address + 0x00C0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Reset	[7]	RW	Specifies the CEC Rx reset bit 0 = No effect 1 = Immediately resets CEC Rx related registers and state machines to its reset value. It will be cleared after one clock	0
Check_Sampling_Error	[6]	RW	Specifies the CEC Rx sampling error check enable bit 0 = Does not check sampling error 1 = Checks sampling error while receiving data bits CEC Rx samples the CEC bus three times (at 1.00, 1.05, and 1.10 ms) and checks whether the three samples are identical.	0
Check_Low_Time_Error	[5]	RW	Specifies the CEC Rx low-time error check enable bit 0 = Does not check low-time error 1 = Checks low-time error while receiving data bits On receiving each bit from the CEC bus, CEC Rx checks the duration of logical 0 from the starting of one bit transfer (falling edge on the CEC bus). Rx checks whether the duration is longer than the maximum time the CEC bus can be in logical 0 (max 1.7 ms).	0
Check_Start_Bit_Error	[4]	RW	Specifies the CEC Rx start bit error check enable bit 0 = Does not check start bit error. 1 = Checks start bit error while receiving a start bit. After receiving a start bit from the CEC bus, CEC Rx checks the duration of logical 0 and 1 of start bit (as specified in CEC specification on page CEC-8). Rx checks whether the duration meets the specification.	0
RSVD	[3:2]	RW	Reserved	2b00
Rx_Host_Busy	[1]	RW	Specifies the CEC Rx host busy bit 0 = Rx receives incoming message and sends acknowledgement. 1 = A host processor is unavailable to receive and process CEC messages. Rx sends "not acknowledged" signal to a message initiator. It indicates that a host processor is unavailable and to receive and process CEC messages.	0
Rx_Enable	[0]	RW	Specifies the CEC Rx start bit 0 = Disables Rx 1 = Enables CEC Rx module to receive a message This bit is cleared after receiving a message.	0

53.10.12.15 CEC_RX_STATUS_2

- Base Address: 0x100B_0000
- Address = Base Address + 0x00E0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Rx_Waiting	[7]	R	Specifies the CEC Rx waiting flag bit 0 = Rx is in other state 1 = CEC Rx waits for a message	0
Rx_Receiving_Start_Bit	[6]	R	Specifies the CEC Rx start bit receiving flag bit 0 = Rx is in other state 1 = CEC Rx receives a start bit	0
Rx_Receiving_Hdr_Blk	[5]	R	Specifies the CEC Rx header block receiving flag bit 0 = Rx is in other state 1 = CEC Rx receives a header block	0
Rx_Receiving_Data_Blk	[4]	R	Specifies the CEC Rx data block receiving flag bit 0 = Rx is in other state 1 = CEC Rx receives data blocks	0
RSVD	[3:0]	R	Reserved	4b0000

53.10.12.16 CEC_RX_STATUS_3

- Base Address: 0x100B_0000
- Address = Base Address + 0x00E4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	R	Reserved	0
Sampling_Error	[6]	R	Specifies the CEC Rx sampling error flag bit 0 = No sampling error occurs 1 = A sampling error occurs while receiving a message CEC Rx samples the CEC bus three times (at 1.00, 1.05, and 1.10 ms). It also sets this bit if Check_Sampling_Error bit in CEC_RX_CTRL_0 is set and if three samples are not identical. It will be cleared if Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register is set to 0.	0
Low_Time_Error	[5]	R	Specifies CEC Rx low-time error flag bit 0 = No low-time error occurs 1 = A low-time error occurs while receiving a message While receiving each bit from the CEC bus, CEC Rx checks the duration of logical 0 from the start of one-bit transfer (falling edge on the CEC bus). If the duration is longer than the maximum time, the CEC bus can be logical 0 (maximum 1.7 ms). CEC RX sets this bit. This bit field will be set to 0 if Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register is set.	0
Start_Bit_Error	[4]	R	Specifies the CEC Rx start bit error flag bit 0 = No start bit error occurs 1 = A start bit error occurs while receiving a message While receiving a start bit from the CEC bus, CEC Rx checks the duration of logical 0 and 1 of a starting bit (as specified in CEC spec. page CEC-8). If the duration does not meet the specification, CEC RX sets this bit. This bit field will be set to 0 if Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register is set.	0
RSVD	[3:1]	R	Reserved	3b000
CEC_Line_Error	[0]	R	Specifies the CEC Rx line error flag bit 0 = No line error occurs 1 = A start bit error line occurs while receiving a message In CEC specification page CEC-13, CEC line error occurs in a period when two consecutive falling	0

Name	Bit	Type	Description	Reset Value
			edges is smaller than a minimum data bit period. Rx checks for this condition, and when it occurs, it sends the line error notification, that is, it sends logical 0 for more than 1.4 to 1.6 times of the nominal data bit period (2.4 ms). This bit will be cleared: • if Rx_Enable bit of CEC_RX_CTRL_0 is set • if Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register is set	

53.10.12.17 CEC_RX_BUFFER_X

- Base Address: 0x100B_0000
- Address = Base Address + 0x0100, 0x0104, + 0x0108, 0x010C
  0x0110, + 0x0114, + 0x0118, + 0x011C,
  0x0120, + 0x0124, + 0x0128, + 0x012C
  0x0130, + 0x0134, + 0x0138, + 0x013C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
Rx_Block_0 to Rx_Block_15	[7:0]	RW	Specifies byte #0 to #15 of CEC message Each byte corresponds to a block in a message. Block_0 is header block and Block_1 to 15 are data blocks.	0

53.10.12.18 CEC_FILTER_CTRL

- Base Address: 0x100B_0000
- Address = Base Address + 0x0180, Reset Value = 0x81

Name	Bit	Type	Description	Reset Value
Filter_Cur_Val	[7]	RW	Specifies the CEC filter current value bit Indicates current value fed to CEC Tx, Rx. If the filter is enabled, this bit specifies the latest value on CEC bus. This value is stable for more than Filter_Th cycles.	1
RSVD	[6:1]	RW	Reserved	6b000000
Filter_Enable	[0]	RW	Specifies the CEC filter enable bit 0 = Disables filter. Directly passes CEC input to CEC Tx, Rx. 1 = Enables filter. Filter propagates signals stable for more Filter_Th cycles.	1

53.10.12.19 CEC_FILTER_TH

- Base Address: 0x100B_0000
- Address = Base Address + 0x0184, Reset Value = 0x03

Name	Bit	Type	Description	Reset Value
Filter_Th	[7:0]	RW	Specifies the filter threshold value If the filter is enabled, it filters out signals that are less stable than Filter_Th cycles.	8b00000011

54 Security Sub System

54.1 Overview

Security Sub-system (SSS) has the following components:

- Advanced Encryption Standard (AES)
- Data Encryption Standard (DES)/3DES
- SHA-1/SHA-256/MD5/PRNG
- Public Key Accelerator (PKA)
- Feeder (FeedCtrl)

Feeder supplies data to a Cryptographic IP and drains data from IP. Feeder has the following external components:

- Block Cipher Receiving DMA (BRDMA)
- Block Cipher Transmission DMA (BTDMA)
- Hash Receiving DMA (HRDMA)
- FIFO and FIFO Interconnections
- Interrupt Controller
- FIFO Controller

SSS has the following external components:

- One APB slave port (for setting SFR)
- One AXI master ports (for DMA)
- One interrupt
  - DMA done interrupts (to notify end of DMA operations)
  - Hash interrupts (to notify end of Hash or Pseudo Random Number Generator (PRNG) operations)
  - A Public Key Accelerator (PKA) done interrupt (to notify end of PKA operations)

Each security IP can be accessed in CPU mode or DMA mode:

- CPU mode
  - For AES, DES, 3DES, SHA-1, SHA-256, MD5, PRNG, and PKA
  - Every input and output data should be carried to and from SSS by the host processor.
- DMA mode (also known as FIFO mode)
  - For AES, DES, 3DES, SHA-1, SHA-256, and MD5
  - DMA supplies input data to each IP through FIFO.
  - DMA drains output data from each IP except hashes and PRNG through FIFO.
  - Block ciphers and hashes can share input data.
  - Output data of block ciphers can be used as input data of the hashes.

Figure 54-1 illustrates the block diagram of SSS.

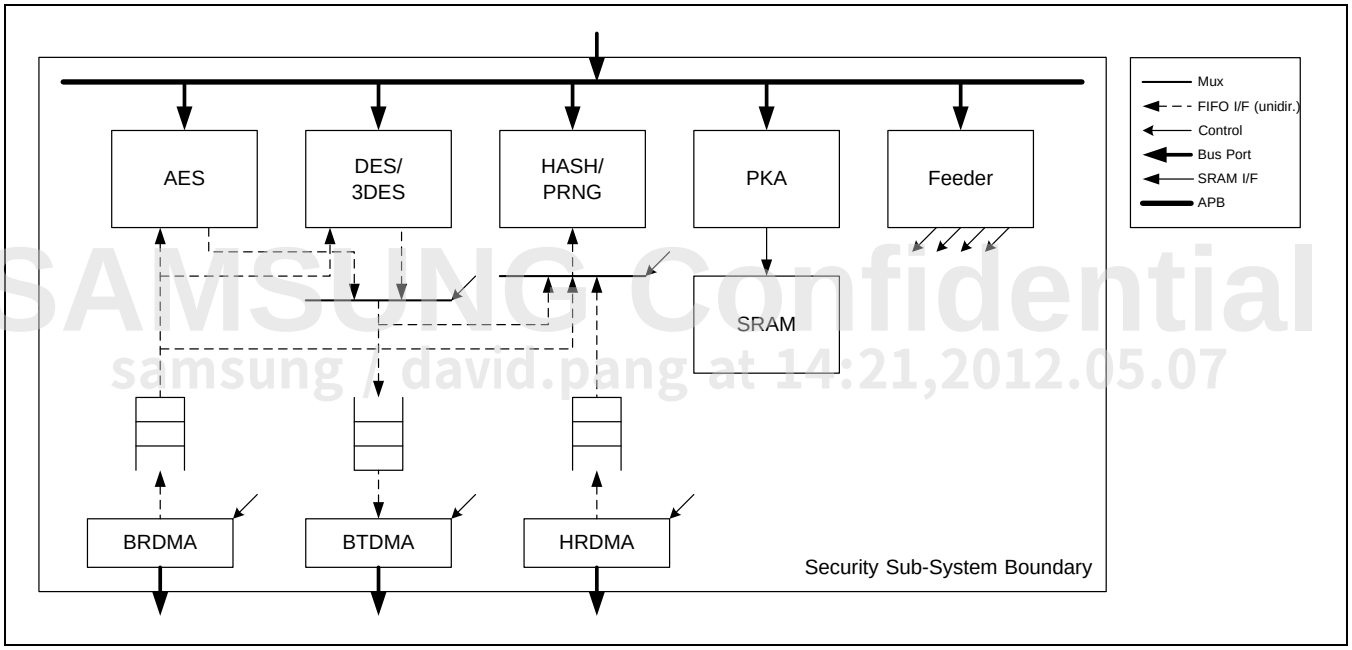


Figure 54-1 Block Diagram of SSS

54.1.1 Features

The features of SSS are:

- AES (ECB, CBC, and CTR mode)
- DES (ECB and CBC mode)
- 3DES (ECB, CBC, EDE, and EEE mode)
- SHA-1 (with hardware padding) and SHA-1 HMAC
- SHA-256 (with hardware padding) and SHA-256 HMAC
- MD5 (w/hardware padding) and MD5 HMAC
- Pseudo Random Number Generator (PRNG)
- Public Key Accelerator (PKA)
- DMA Support for AES, DES, 3DES, SHA-1, SHA-256, and MD5
- Block Ciphers combined with Hashing
  - Hashing Ciphered/Deciphered Data
  - Hashing Data before Ciphering/Deciphering

54.2 Functional Description

This section includes:

- Data flows through SFR (CPU mode)
- Data flows through DMA (DMA mode)
- Byte swapping options

54.2.1 Data Flows through SFR (CPU Mode)

Using SFR, you can access full functions of AES, DES, and Hash/PRNG. You can supply input data to SSS, trigger an operation of SSS and extract output data from SSS through SFR.

54.2.2 Data Flows through DMA (DMA Mode)

Block cipher Receiving DMA (BRDMA) supplies input data into block ciphers like AES or DES as illustrated in *Figure 54-2*. On the other hand, BTDMA receives output data from AES or DES. Only one block, AES or DES, can use DMA. If other block that does not occupy DMA, it can be used in CPU mode.

Figure 54-2 illustrates the DES or 3DES data flow.

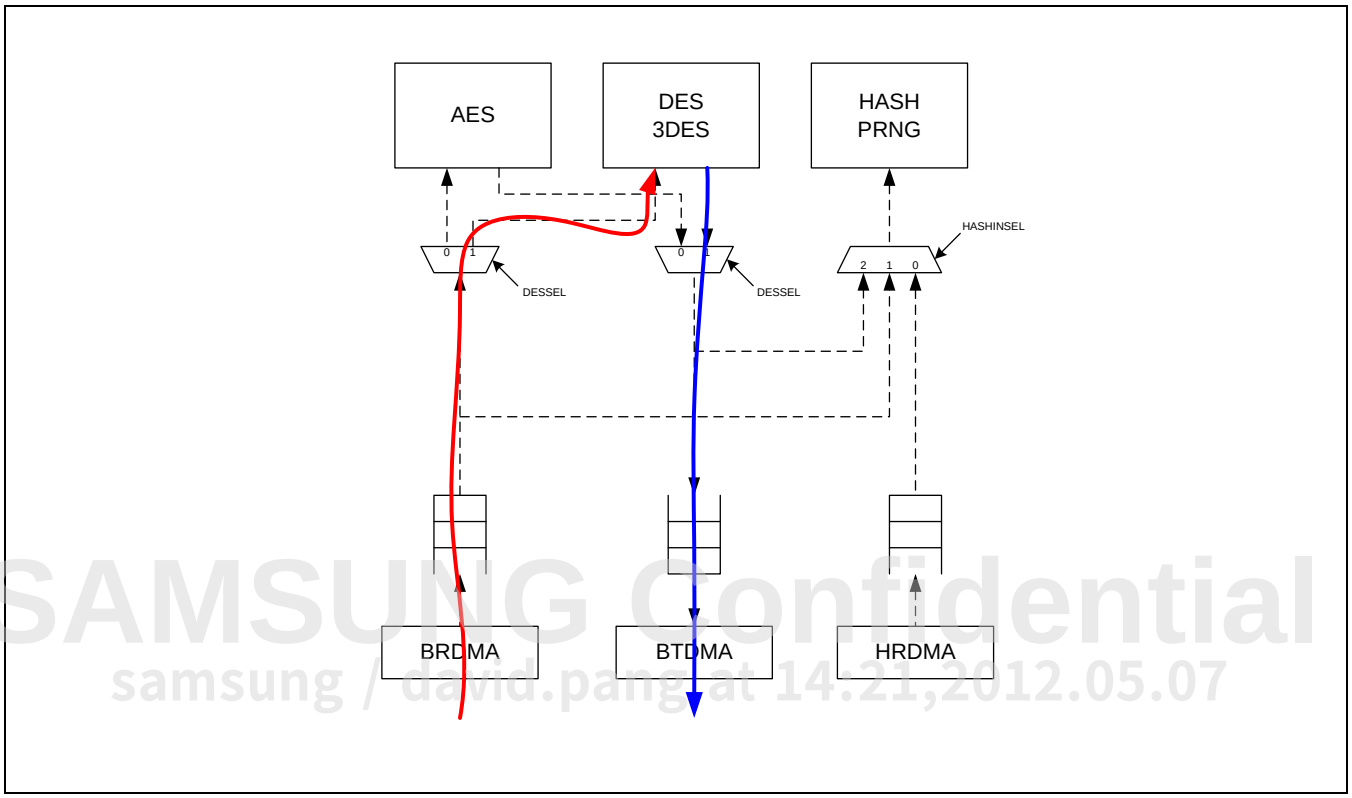


Figure 54-2 DES or 3DES Only Data Flow

The hash block uses HRDMA, which represents Hash Receiving DMA. HRDMA can work independently of BRDMA or BTDMA. In this case, the hash processes different data stream than block ciphers.

Figure 54-3 illustrates the AES and Hash parallel data flow.

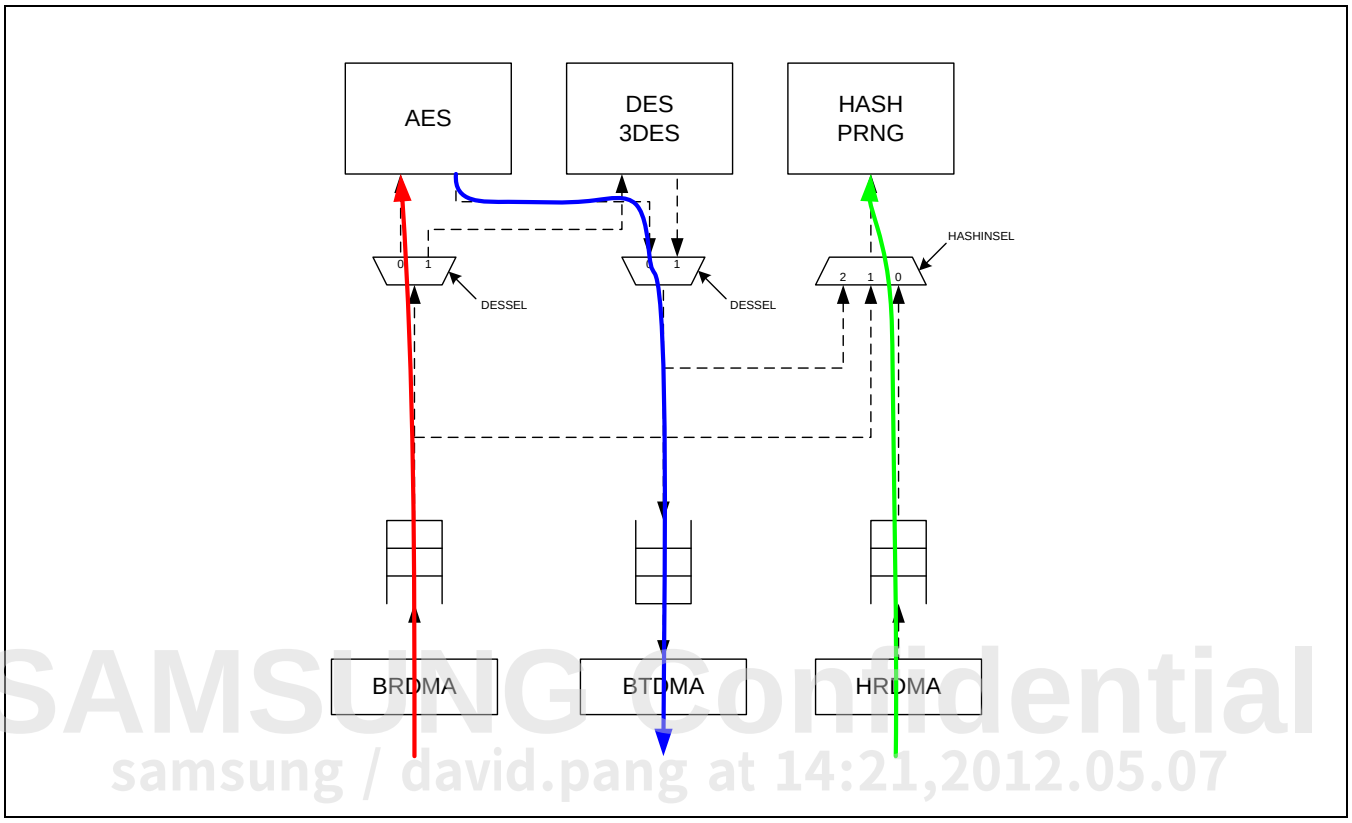


Figure 54-3 AES and Hash Parallel Data Flow

Figure 54-4 illustrates the configuration of AES and hash with shared input data from external memory.

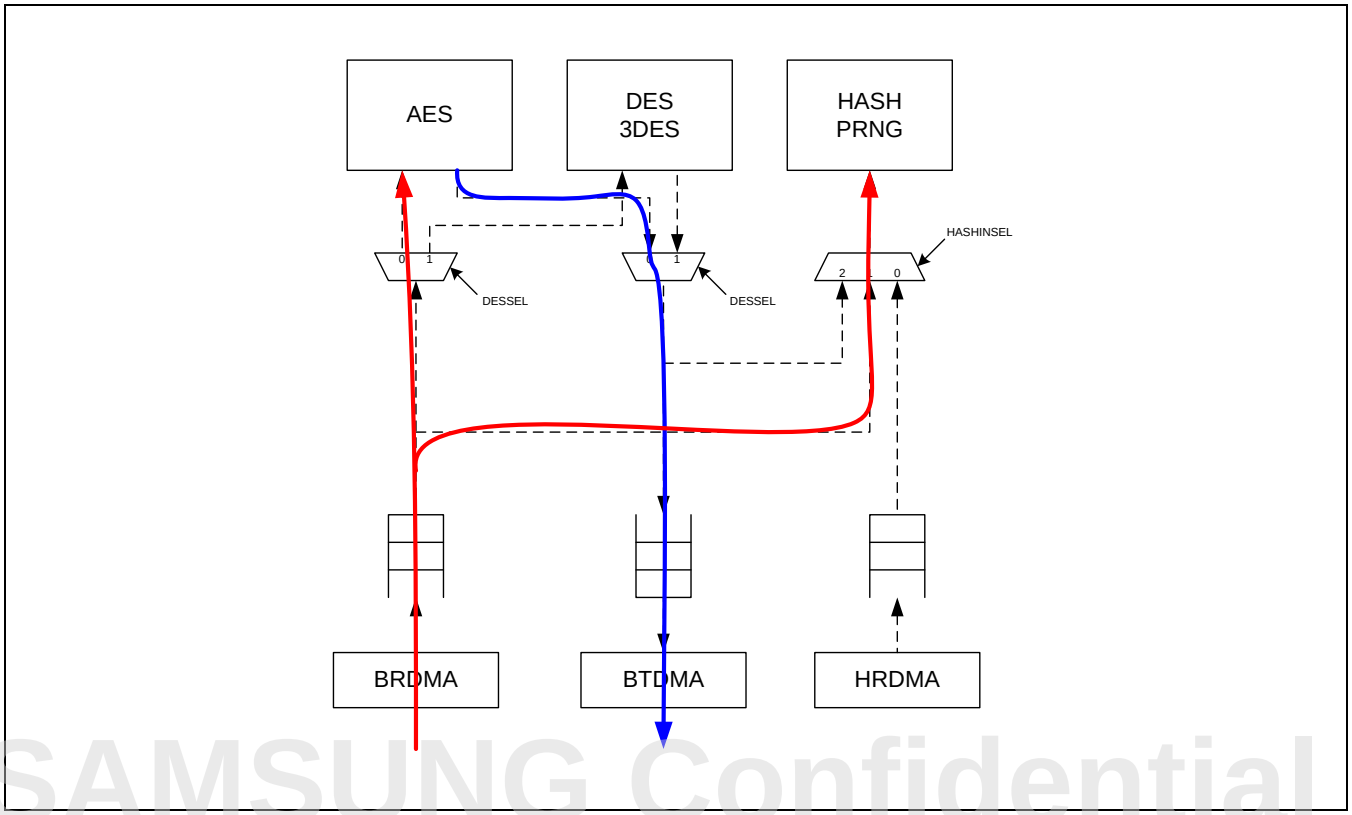


Figure 54-4 Data Flow of AES and Hash with Shared Input

Figure 54-5 illustrates the configuration of hash when it processes the output of AES.

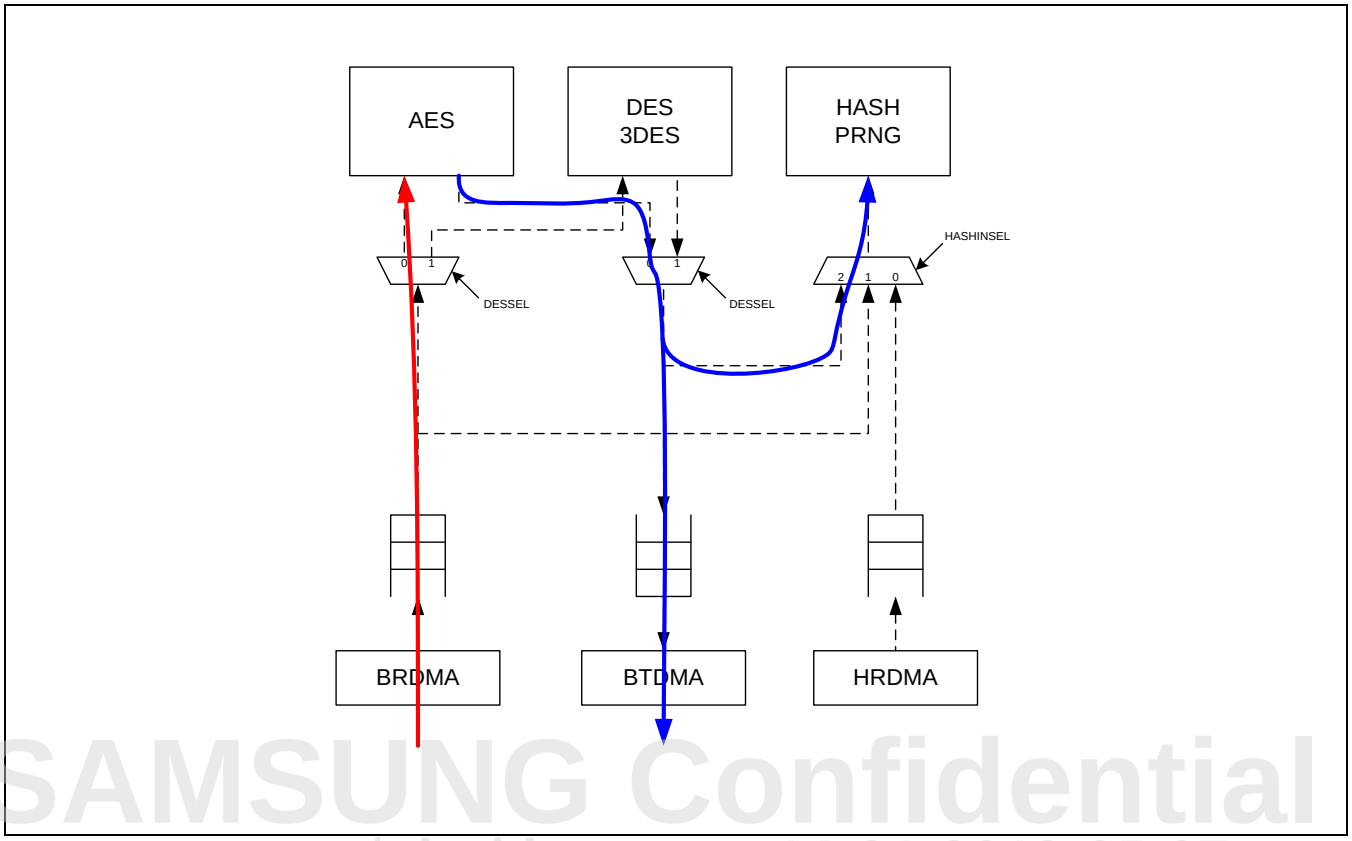


Figure 54-5 Data Flow of Hashing the Output of AES

The HRDMA cannot be used in the above two cases.

54.2.2.1 FIFO Configuration

The FCFIFOCTRL register changes FIFO configuration. The DESSEL bit of FCFIFOCTRL selects Data Encryption Standard (DES) or Advanced Encryption Standard (AES). Also, HASHINSEL bits select the hash input data from three possible inputs that comes from HRDMA, input of block cipher and output of block cipher.

Figure 54-6 illustrates the FIFO and FIFO interconnections.

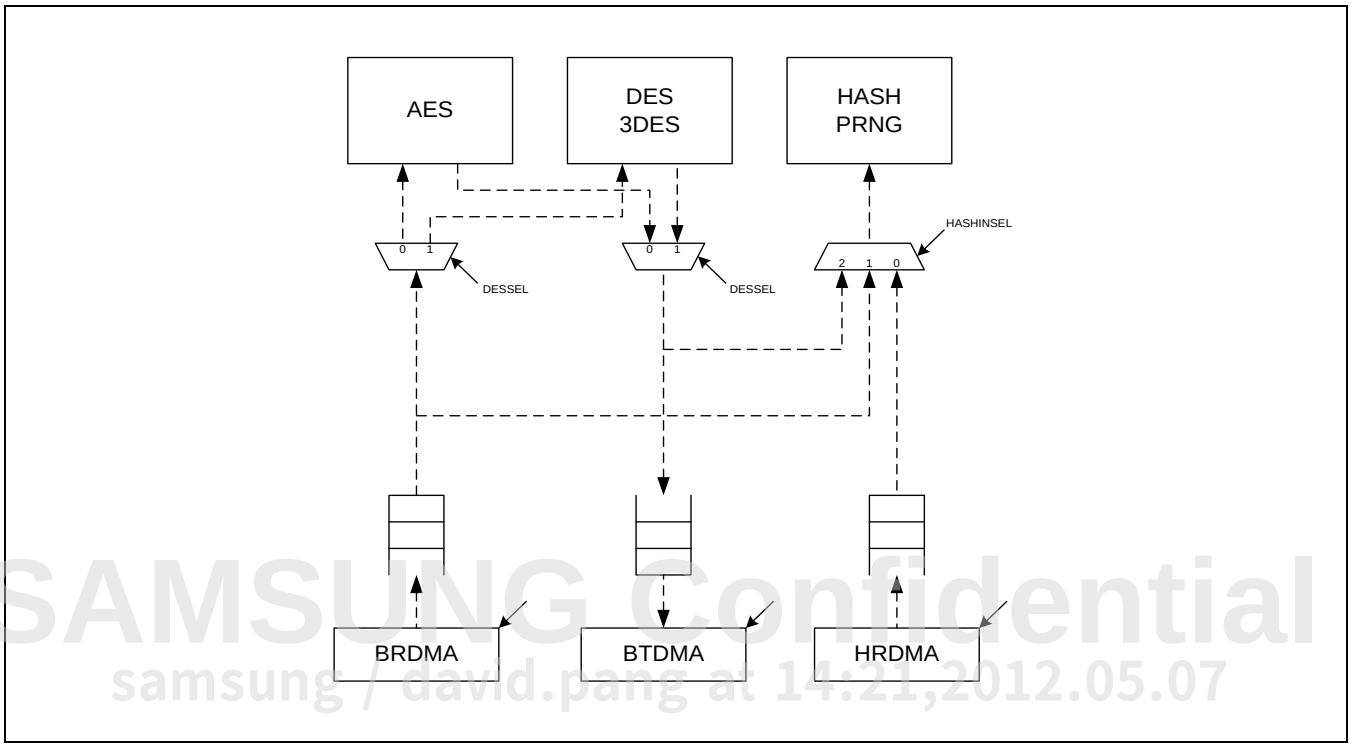


Figure 54-6 FIFO and FIFO Interconnections

Table 54-1 describes the registers of FCFIFOCTRL.

Table 54-1 Register Description of FCFIFOCTRL

Name	Bit	Description	Reset Value
RSVD	[31:3]	Reserved	29'd0
DESSEL	[2]	Destination block cipher of FIFO. AES (= 0)/DES (= 1)	1'd0
HASHINSEL	[1:0]	Data from independent source (= 0) Data from block cipher input (= 1) Data from block cipher output (= 2) Reserved (= 3)	2'd0

Table 54-2 describes the register values for all possible use cases.

Table 54-2 Register Values for All Possible Use Cases

Mode	Description	DESSEL	HASHINSEL
AES then Hash	AES output is fed to Hash.	1'd0	2'd2
DES then Hash	DES output is fed to Hash.	1'd1	2'd2
AES and Hash	AES and Hash shares input.	1'd0	2'd1
DES and Hash	DES and Hash shares input.	1'd1	2'd1
Hash alone and AES	AES and Hash operates independently.	1'd0	2'd0
Hash alone and DES	DES and Hash operates independently.	1'd1	2'd0

54.2.2.2 DMA Configuration

Each DMA has the main parameters as follows:

- STARTADDR (32-bit): Specifies start address of DMA. The address does not need to be 32 bit aligned. Its value increases by eight after every transaction.
- LENGTH (32-bit): Specifies data length of DMA. The length needs to be 64 bit aligned. In other words, value should be a multiple of eight. HRDMA does not have this constraint. HRDMA rounds up any value to a multiple of eight. For example, if the value is 13, the value becomes 16 automatically. Its value decreases by eight after every transfer. When a number is written to this register, DMA starts immediately.
- FLUSH (1-bit): If this bit is set to high, then data flushes from FIFO and DMA. This bit clears automatically.
- BYTESWAP (1-bit): If this bit is set to high, then the data is byte-swapped at 64-bit boundaries. **If this bit is set to 0(default), the data is transferred without byte-swap**. Normally this bit should be "0".
- AWCACHE, ARCACHE (4-bit): The AXI signals for each channel are determined by the value of this field.
- AWPROT, ARPROT (3-bit): The AXI signals for each channel are determined by the value of this field.

Table 54-3 lists the DMA options.

Table 54-3 Comparison of DMA Options

DMA	BRDMA	BTDMA	HRDMA
Address Alignment	Byte	Byte	Byte
Length Alignment	Block (multiple of 8 Bytes for DES and 16 Bytes for AES)	Block (multiple of 8 Bytes for DES and 16 Bytes for AES)	Byte
Direction	Receive (into SSS)	Transmit (from SSS)	Receive (into SSS)
Address Registers	Source Address	Destination Address	Source Address
Interrupt	BRDMA_DONE	BTDMA_DONE	HRDMA_DONE

54.2.2.3 Internal Interrupt Controller (IntCtrl)

Each of three DMA interrupt signals has control scheme. Each Interrupt signal is generated by a DMA and masked by FCINTENSET register in bitwise operation. Each bit in FCINTENSET can be set by writing "1" to the corresponding bit in FCINTENSET and cleared by writing "1" to the corresponding bit in FCINTENCLR.

Figure 54-7 illustrates the interrupt controller scheme for one DMA interrupt signal.

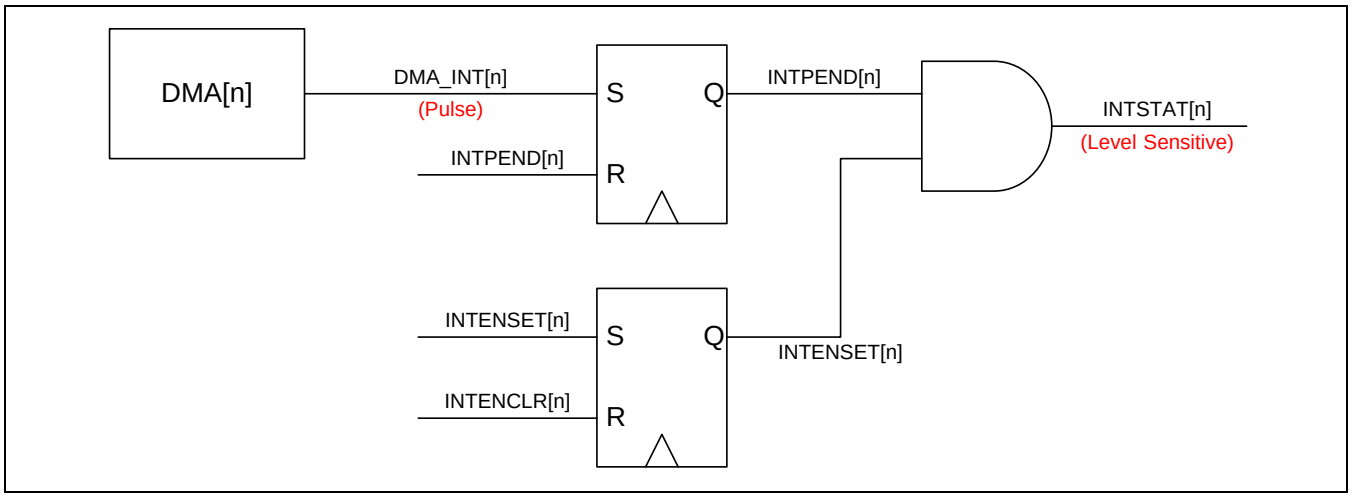


Figure 54-7 Interrupt Controller Scheme for One DMA Interrupt Signal

Table 54-4 describes the four interrupt controller registers.

Table 54-4 Four Interrupt Controller Registers

Name	Base Address	Description
FCINTSTAT	FEED_REG_BASE + 0x0000	Interrupt Status of Feeder
FCINTENSET	FEED_REG_BASE + 0x0004	Interrupt Enable Set Register of Feeder
FCINTENCLR	FEED_REG_BASE + 0x0008	Interrupt Enable Clear Register of Feeder
FCINTPEND	FEED_REG_BASE + 0x000C	Pending Interrupts of Feeder

Table 54-5 describes the bit field of FCINTSTAT. The same bit order is applied to all four interrupt registers (FCINTSTAT, FCINTENSET, FCINTENCLR, and FCINTPEND).

Table 54-5 Bit Field Description of FCINTSTAT

Name	Bit	Description	Reset Value
RSVD	[31:8]	Reserved	24’d0
PARTIAL_DONE	[7]	Interrupt signal of Hash that informs end of partial operation.	1’b0
PRNG_DONE	[6]	Interrupt signal of PRNG that informs end of PRNG operation.	1’b0
MSG_DONE	[5]	Interrupt signal of Hash that informs end of normal hashing operation.	1’b0
PRNG_ERROR	[4]	Interrupt signal of PRNG that informs abnormal access such as getting random number before seed setting.	1’b0
BRDMA_DONE	[3]	Interrupt signal of Block cipher Receiving DMA that informs end of DMA operation.	1’d0
BTDMA_DONE	[2]	Interrupt signal of Block cipher Transmitting DMA that informs end of DMA operation.	1’d0
HRDMA_DONE	[1]	Interrupt signal of Hash Receiving DMA that informs end of DMA operation.	1’d0
PKA_DONE	[0]	Interrupt signal of PKA that informs end of PKA operation.	1’d0

Table 54-6 describes how to clear each interrupt.

Table 54-6 How to Clear Each Interrupt

Interrupt Source	Interrupt Name	How to Clear
Hash	PRNG_DONE	Write "1" into each bit of HASH_STATUS (Hash Status Register)
	PARTIAL_DONE	
	MSG_DONE	
	PRNG_ERROR	Cleared only by setting seed
DMA	BRDMA_DONE	Write "1" into each bit of FCINTPEND in Internal Interrupt Controller
	BTDMA_DONE	
	HRDMA_DONE	
PKA	PKA_DONE	

54.2.3 Byte Swapping Options

This section describes byte swapping of AES, DES and Hash.

54.2.3.1 Byte Swapping of AES, DES, and Hash

SSS supports byte-swapping options for various data. Byte swapping in this context refers to byte order reverse in a 32-bit or 64-bit word boundary.

Table 54-8 illustrates the AES byte swapping scheme.

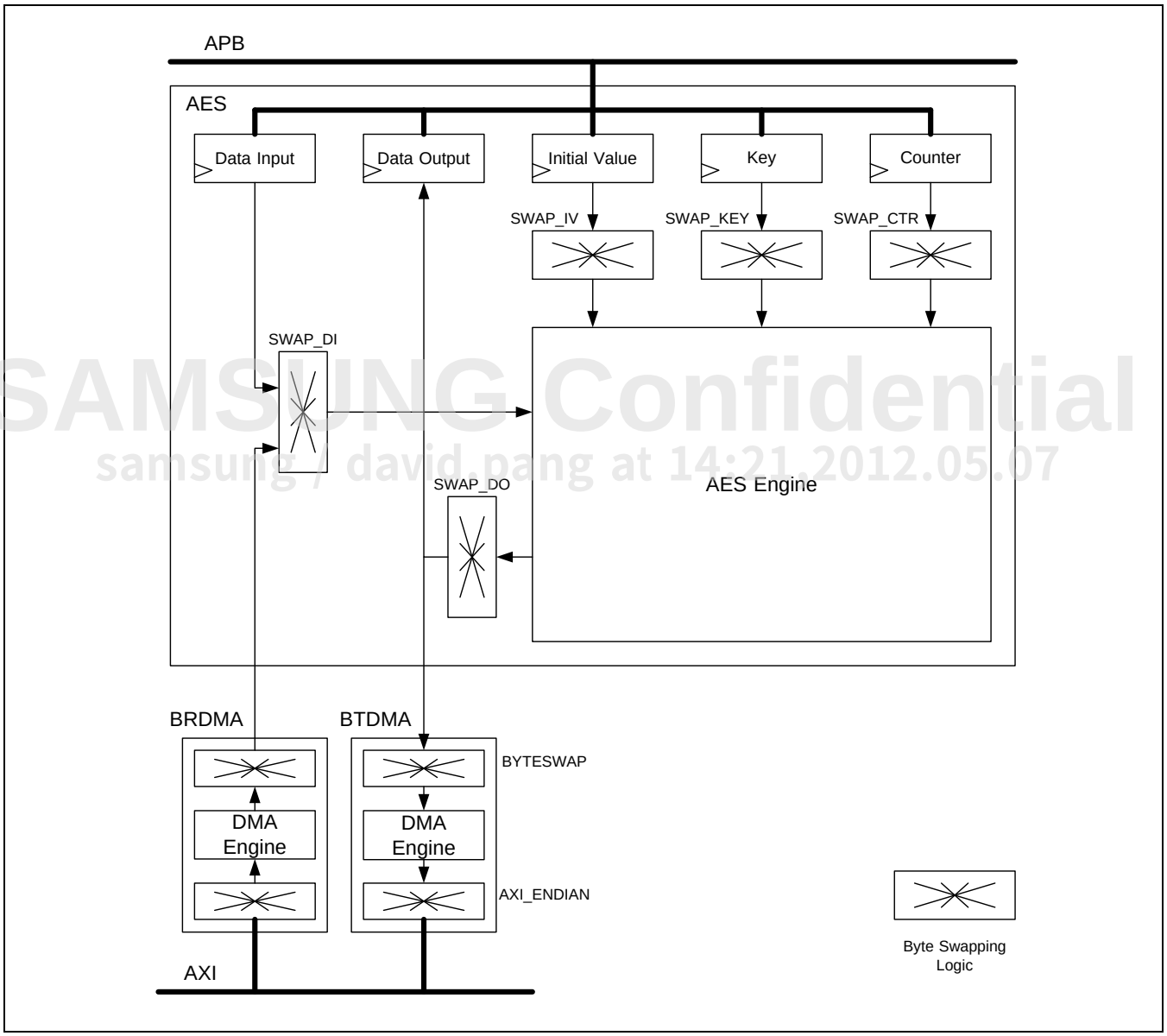


Figure 54-8 AES Byte Swapping Scheme

As illustrated in *Figure 54-8*, AES has five swapping options for each of the five data. They are:

- data input
- data output
- initial value
- key
- counter

Additionally, each DMA (BRDMA, BTDMA, and HRDMA) has its own two swapping options. They are:

1. **AXI_ENDIAN:** This option is in FCGLOBAL and applies to each DMA. For most little endian system, this option should be set as LITTLE (2'b00), which is the default value.

2. **BYTESWAP:** This option controls byte swapping of 64-bit.

In case of DES and Hash, they have the same scheme.

Table 54-7 lists the change of byte orders from memory to AES/DES/Hash.

Table 54-7 Change of byte Orders from Memory to AES/DES/Hash

Change of Byte Orders from Memory to AES/DES/Hash				
Data in Memory	"ABCDEFGH"			
Data in BUS	"HGFEDCBA" (Little Endian)	"ABCDEFGH" (Byte-Invariant Big Endian)	"EFGHABCD" (Word-Invariant Big Endian)	
AXI_ENDIAN in FCGLOBAL Register	2'b00 (Default) (Little Endian)	2'b01 (Byte-Invariant Big Endian)	2'b10 (Word-Invariant Big Endian)	
Data in DMA	"HGFEDCBA"			
BYTESWAP in FCBRDMAC/ FCBTDMAC/FCHRDMAC	NO SWAP (Default)		SWAP	
SWAP_DI/DO in AES/DES/Hash	SWAP (Default)	NO SWAP (Word Swap)	SWAP (Default)	NO SWAP (Word Swap)
Data in AES/DES/Hash	"ABCDEFGH"	"DCBAHGFE"	"HGFEDCBA"	"EFGHABCD"

DES and Hash have the same scheme.

Figure 54-9 illustrates the DES byte swapping scheme.

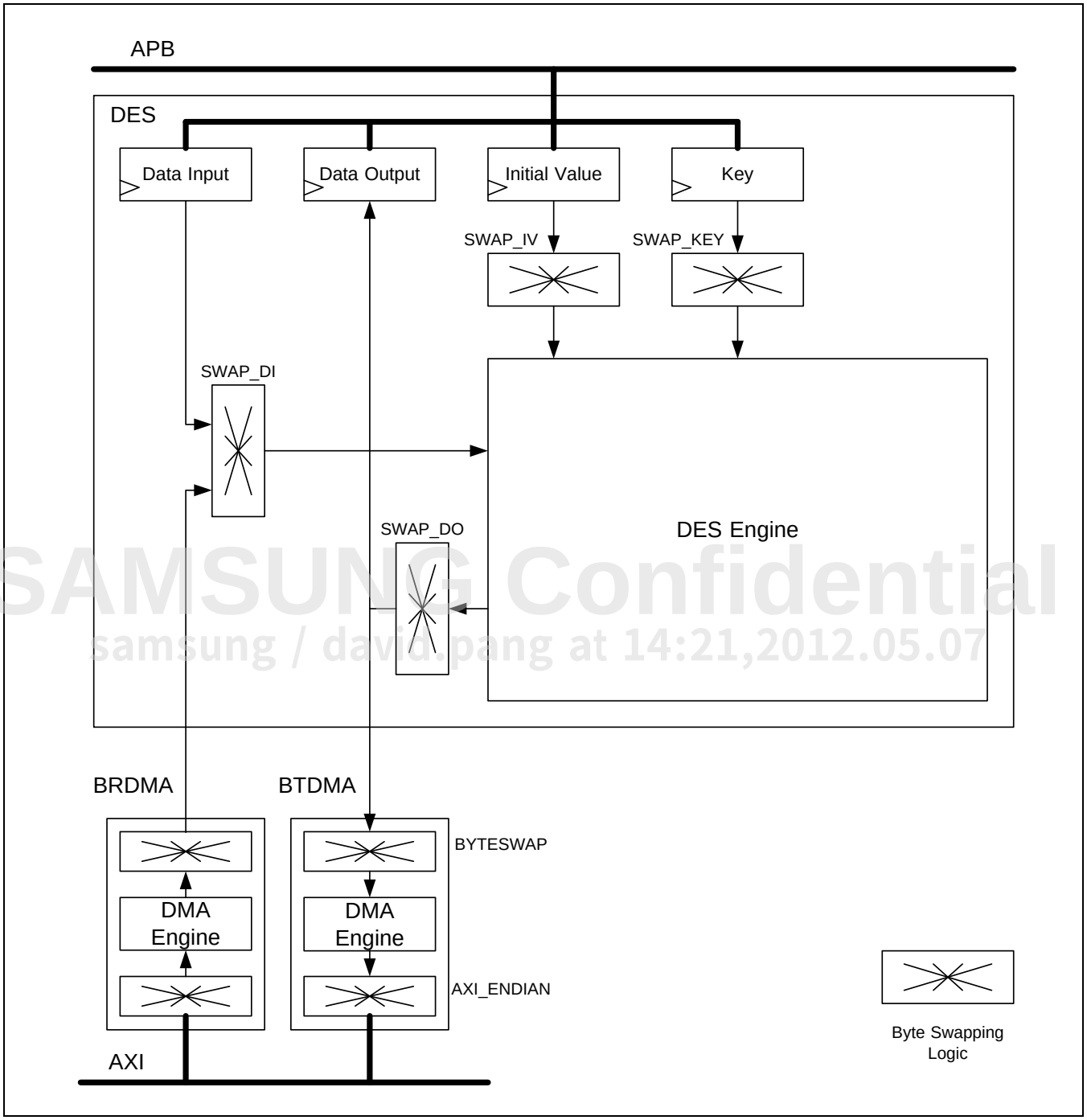


Figure 54-9 DES Byte Swapping Scheme

Figure 54-10 illustrates the Hash byte swapping scheme.

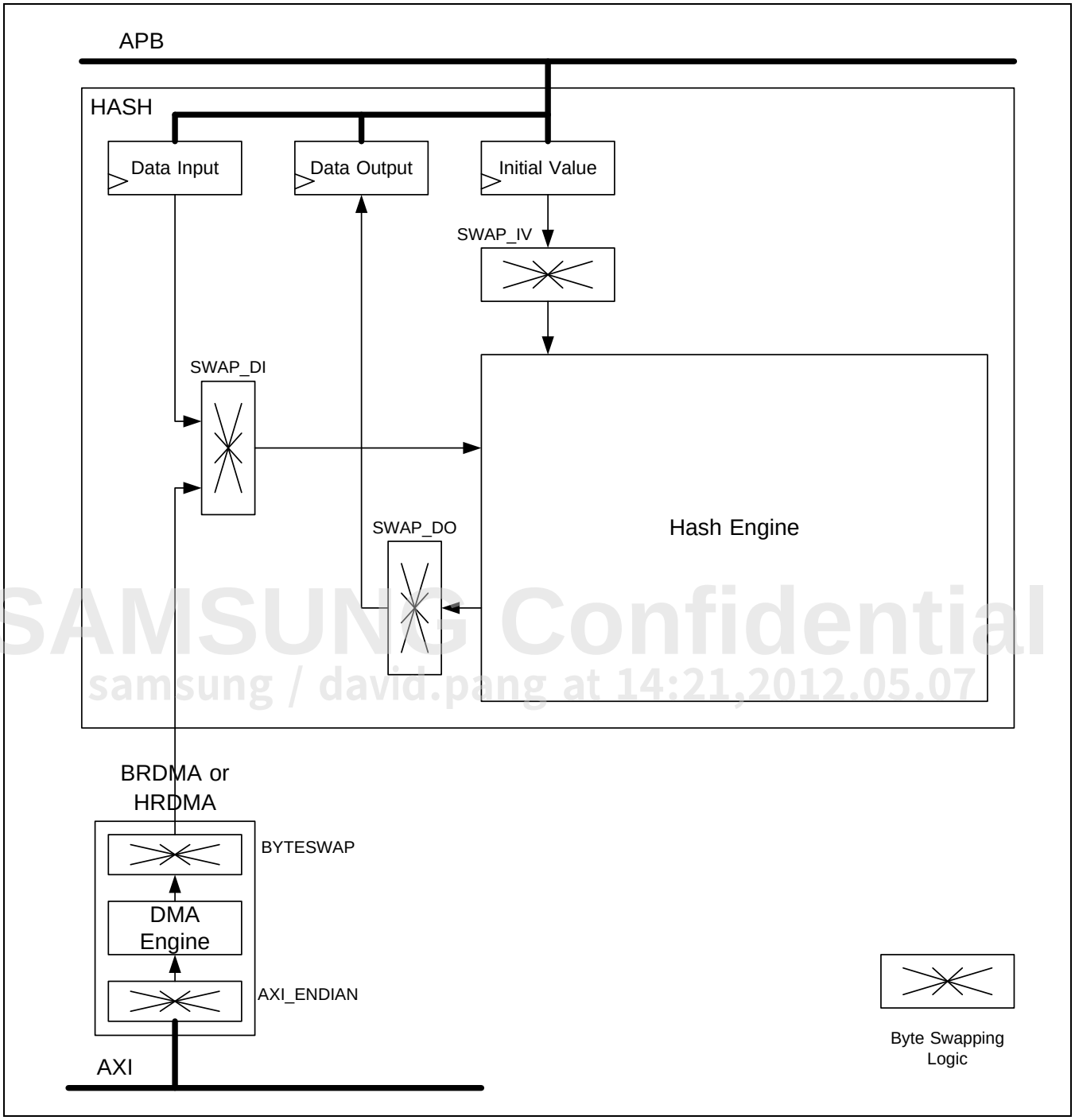


Figure 54-10 Hash Byte Swapping Scheme

54.3 Programmer's Model

This section includes:

- Feeder
- AES
- DES/3DES
- HASH/PRNG
- PKA

54.3.1 Feeder

For FIFO mode block ciphers, the flow chart may be a software guide. The order of each step can be interchangeable. However, before "Starts the Block Cipher" step, configuration of FIFO and Block Cipher should be ready. You can change the last step "Wait for a DMA interrupt" into polling FCINTPEND register. However, the resulting performance degradation of bus performance should be considered.

Figure 54-11 illustrates the flow chart of FIFO mode block cipher.

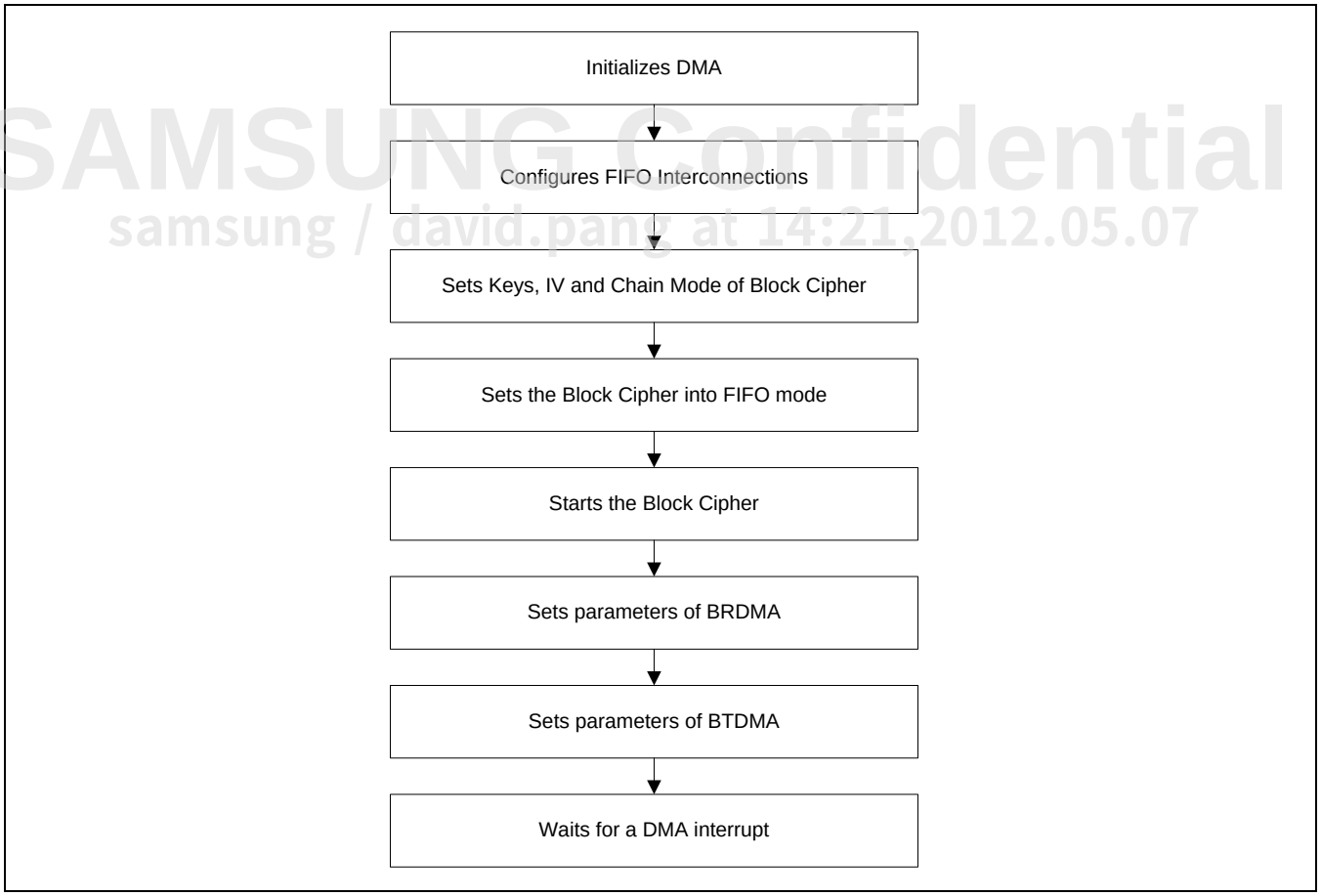


Figure 54-11 Flow Chart of FIFO Mode Block Cipher

Figure 54-12 illustrates the flow chart of FIFO mode hashing. The main difference from the prior one is that only one DMA and HRDMA should be set.

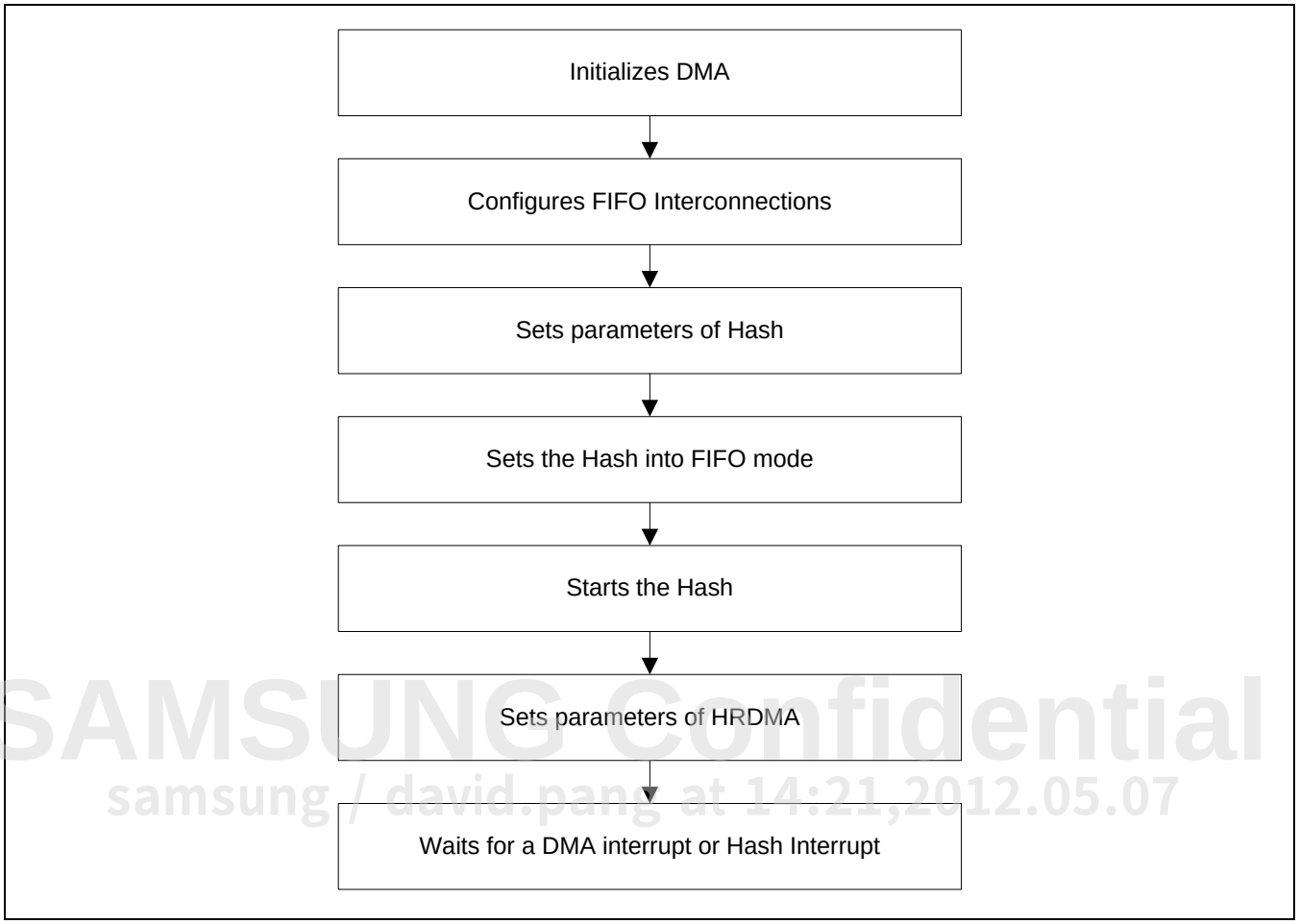


Figure 54-12 Flow Chart of FIFO Mode Hashing

Figure 54-13 illustrates the flow chart of FIFO mode block cipher combined with hashing. Before starting block cipher and hashing, FIFO interconnections and block cipher and hashing should be ready for processing.

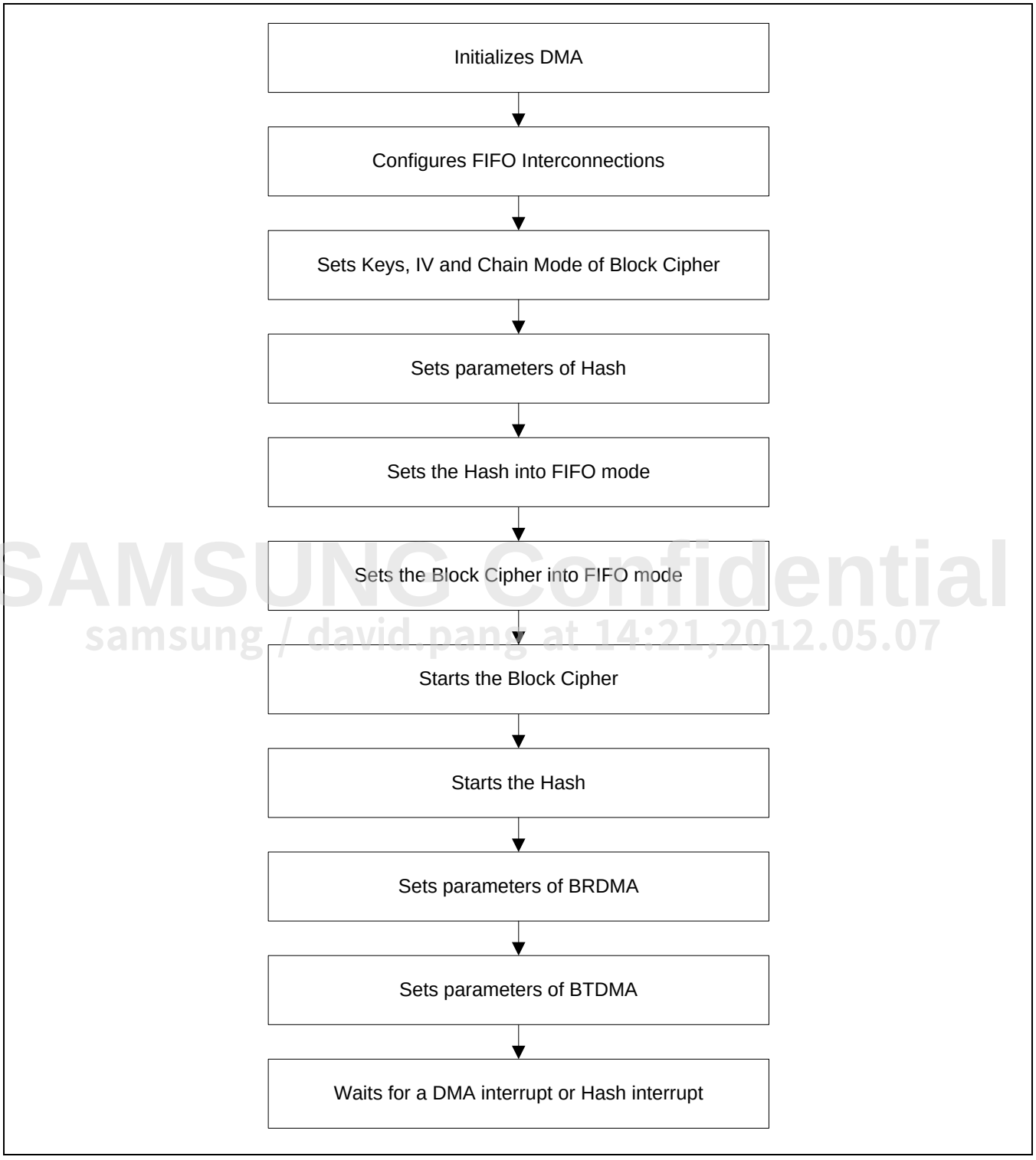


Figure 54-13 Flow Chart of FIFO Mode Block Cipher Combined with Hashing

54.3.2 AES

The following flows show how to operate AES:

1. AES CPU flow using one buffer
2. AES CPU flow using two buffers
3. AES DMA flow

AES operation automatically starts as soon as input register fills up. In "Data output using SFR" step, we have to zeroize output valid signal after reading output data using SFR. The next step of "Data output using SFR" can be either "Data input using SFR" or "Finish" by existence of input data.

During executing an operation, we must not write key, IV, counter, and control registers. Consequently, it is not possible to configure next AES operation (packet) in advance.

Figure 54-14 illustrates the AES CPU flow using one buffer.

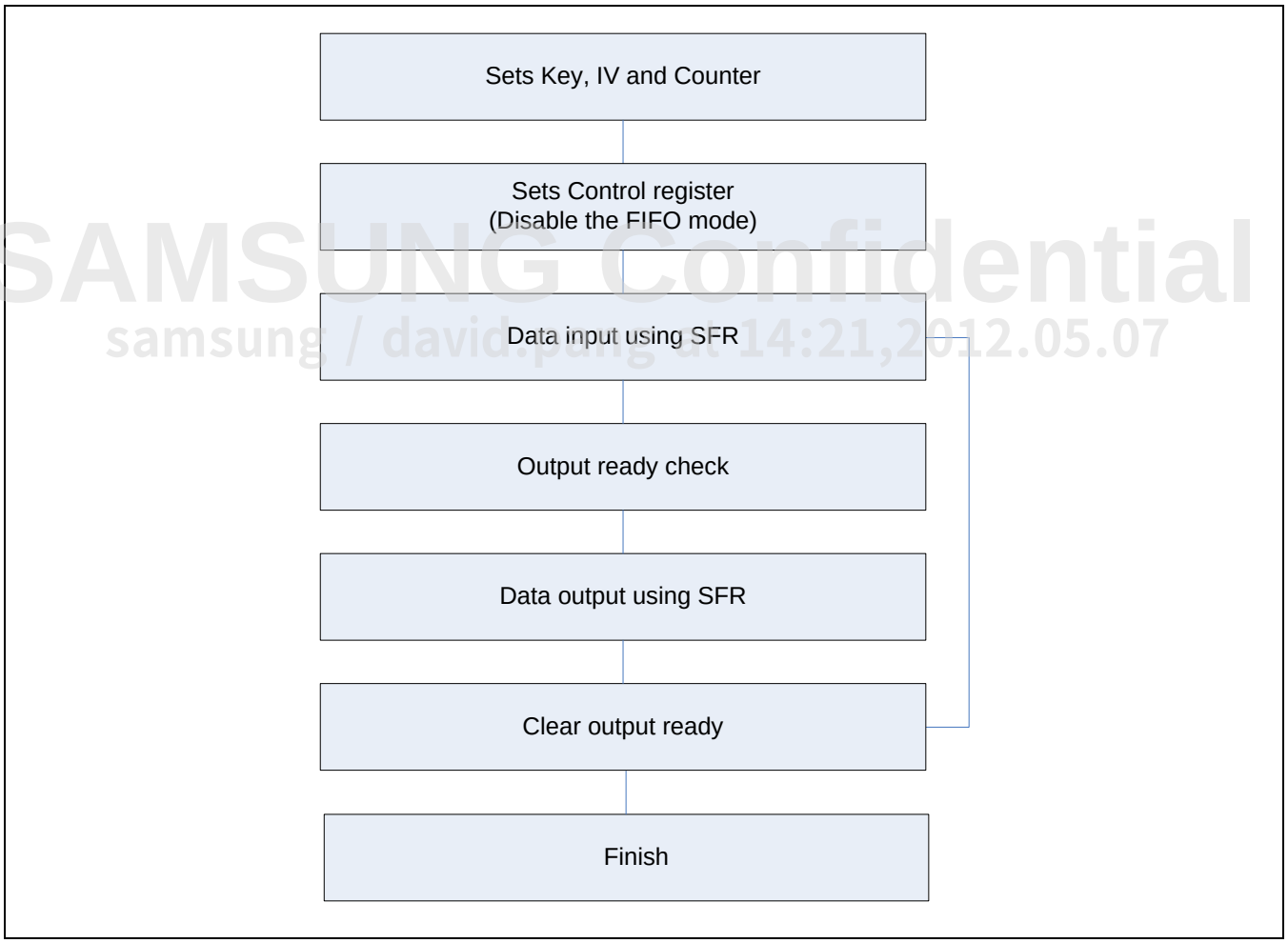


Figure 54-14 AES CPU Flow Using One Buffer

Figure 54-15 illustrates the AES CPU flow using two buffers. The difference from prior is that it continuously gets two input using input buffer and working buffer.

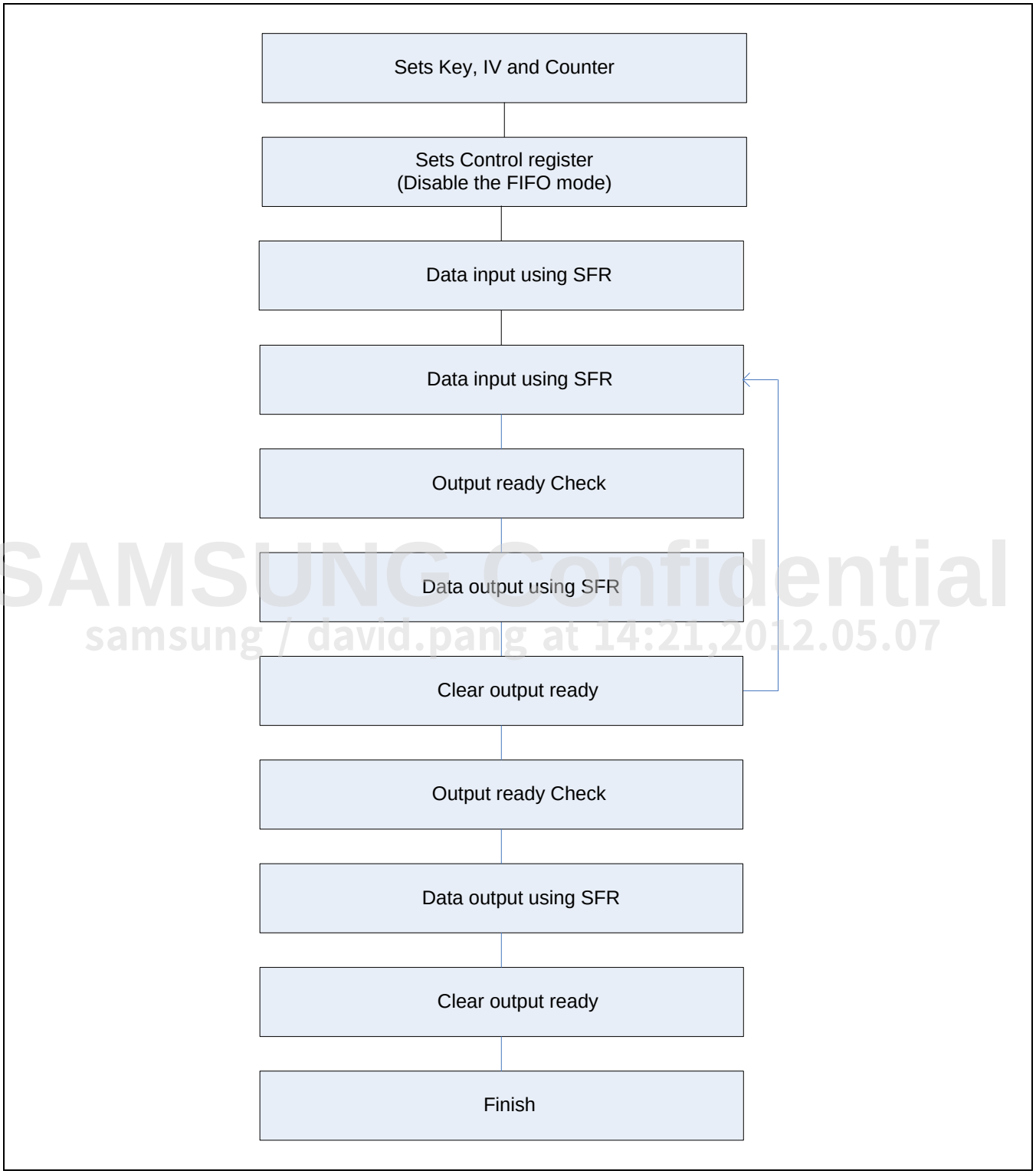


Figure 54-15 AES CPU Flow Using Two Buffer

Figure 54-16 illustrates the AES DMA flow. This is the fastest way to perform AES operations.

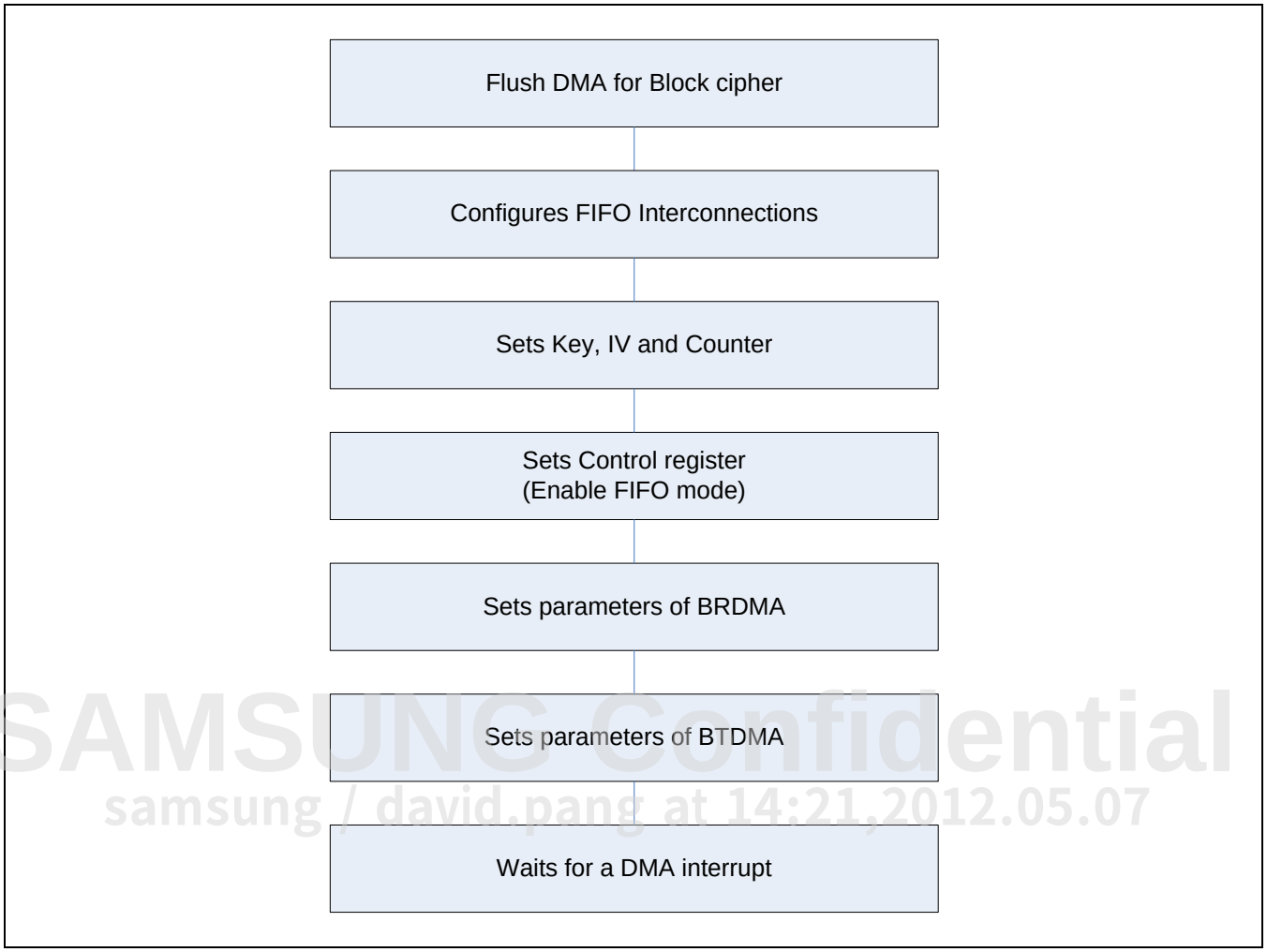


Figure 54-16 AES DMA Flow

54.3.3 DES/3DES

The following flows show describe how to operate DES/3DES:

1. DES/3DES CPU flow using one buffer
2. DES/3DES CPU flow using two buffers
3. DES/3DES DMA flow

DES/3DES operation automatically starts as soon as input register fills up. In "Data output using SFR" step, we have to zeroize output valid signal after we read output data using SFR. The next step of "Data output using SFR" can be either "Data input using SFR" or "Finish" by existence of input data.

Figure 54-17 illustrates the DES/3DES CPU flow using one buffer.

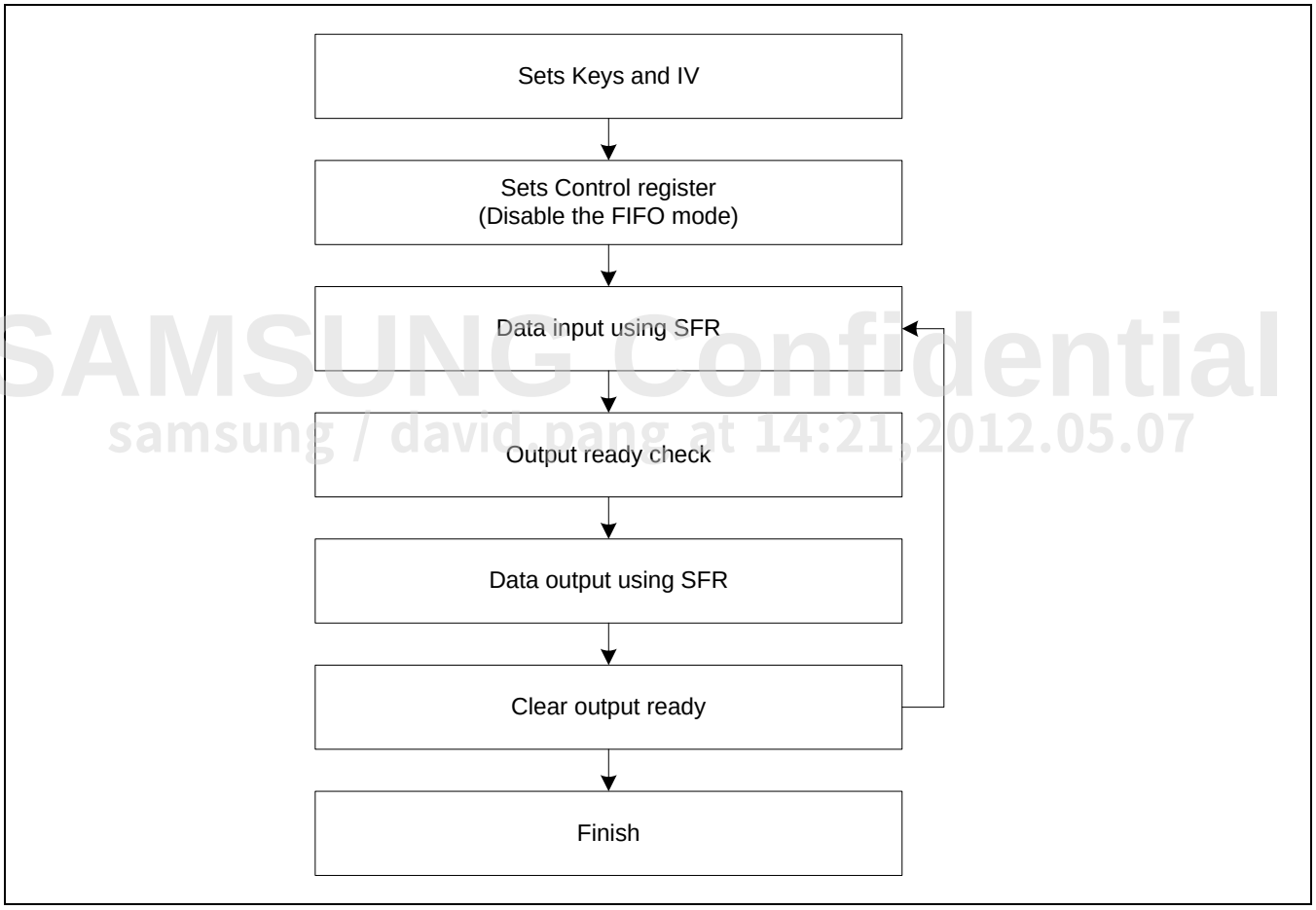


Figure 54-17 DES/3DES CPU Flow Using One Buffer

Figure 54-18 illustrates the DES/3DES CPU flow using two buffers. The difference from prior is that it continuously gets two input using input buffer and working buffer.

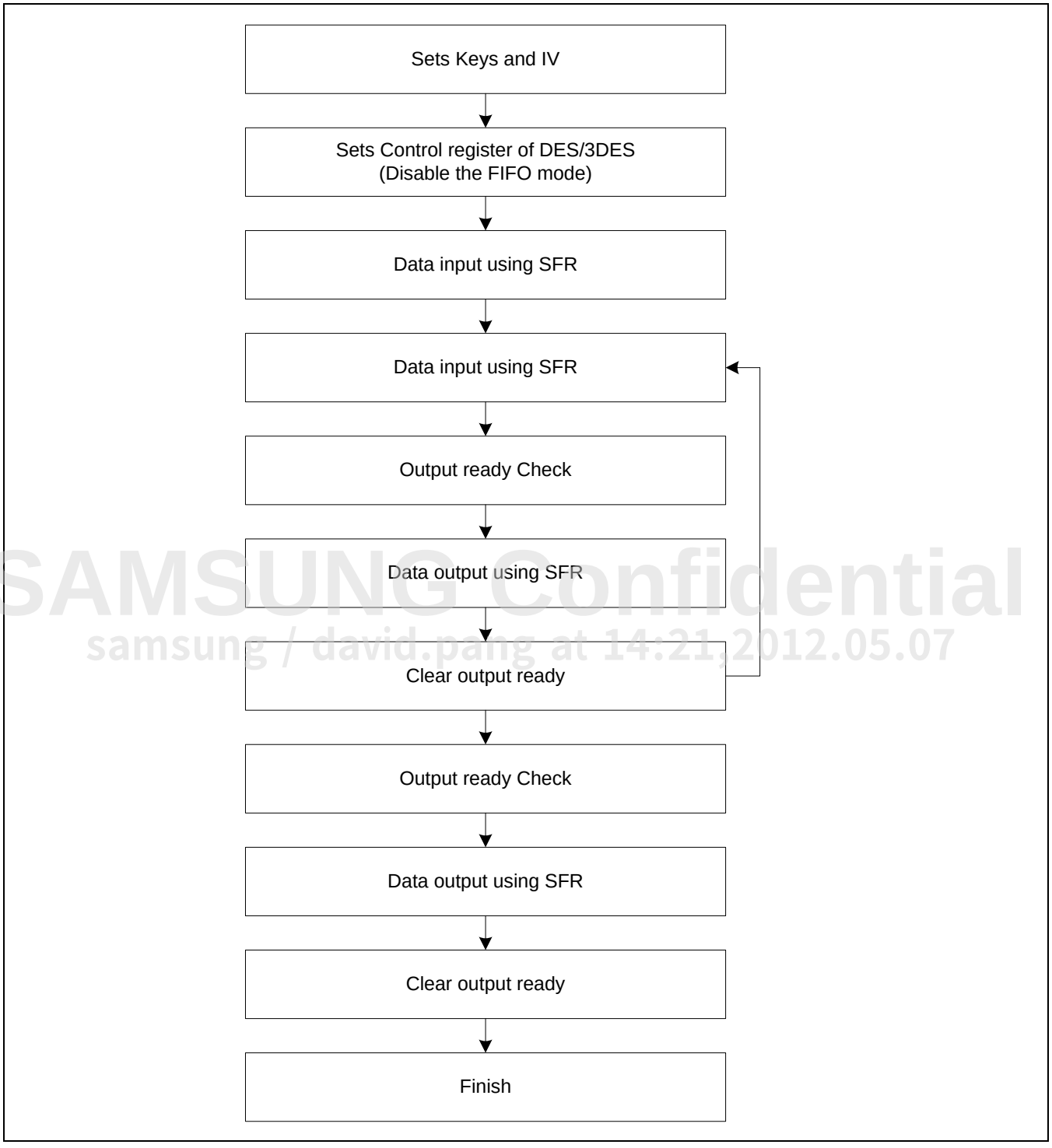


Figure 54-18 DES/3DES CPU Flow Using Two Buffer

Figure 54-19 illustrates the DES/3DES DMA flow. This is the fastest way to perform DES/3DES operations.

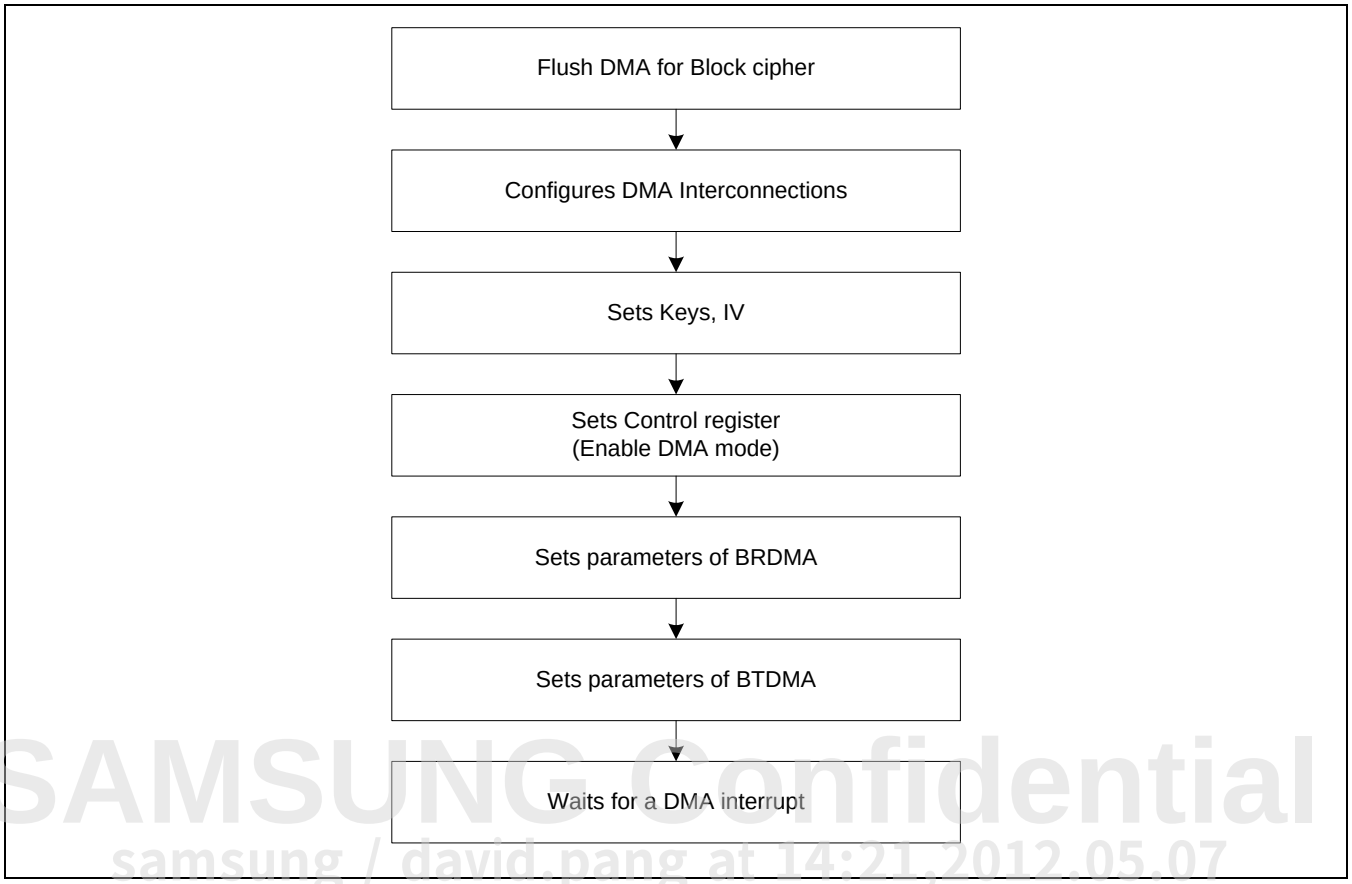


Figure 54-19 DES/3DES DMA Flow

54.3.4 HASH/PRNG

This section includes:

- Hash operation flow
- HMAC operation flow

54.3.4.1 Hash Operation Flow

Figure 54-20 illustrates the Hash operation flow.

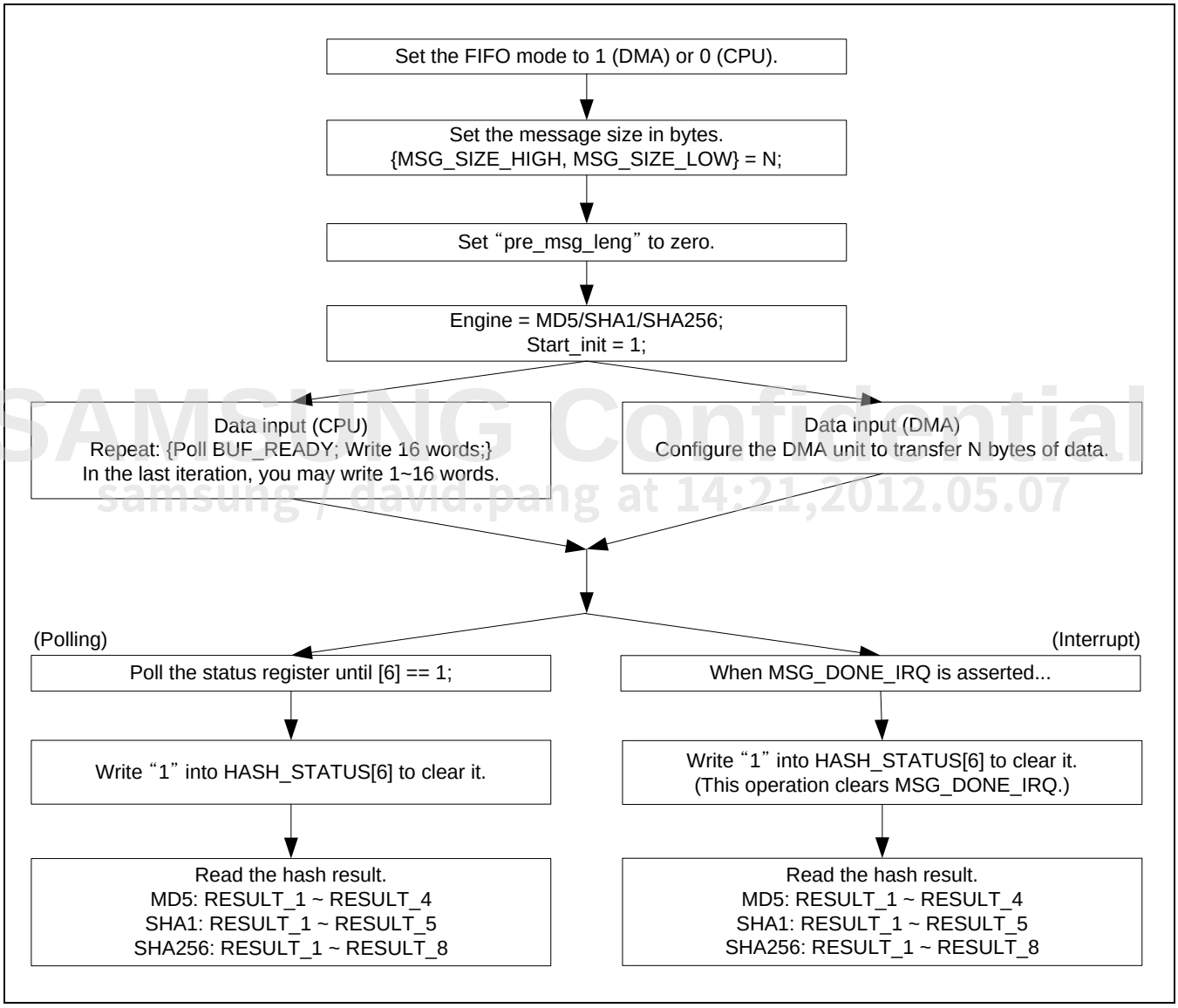


Figure 54-20 Hash Operation Flow

54.3.4.2 HMAC Operation Flow

This section includes:

- Key setup
- HMAC operation
- Hash with partial results
- HMAC with partial results
- PRNG
- Block cipher and hash together
- Special Cases

54.3.4.2.1 Key Setup

The Hash/HMAC engine in SSS_V4.0 includes an internal 512-bit key storage. The value stored in the key storage will be used automatically for both the inner hash and the outer hash.

As illustrated in *Figure 54-21*, each address in register map corresponds to each word of the key storage.

Figure 54-21 illustrates the internal key storage.

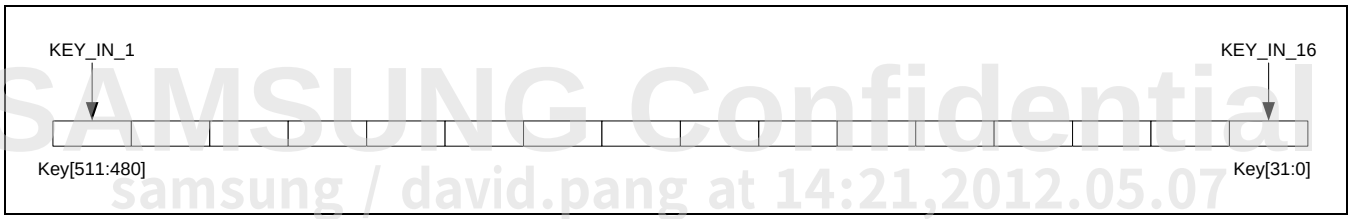


Figure 54-21 Internal Key Storage

The user should initialize key storage before an Hash-based Message Authentication Code (HMAC) operation (including the zero padding). If the same key is used for many HMAC operations, the key setup may be skipped.

Figure 54-22 illustrates the HMAC key setup.

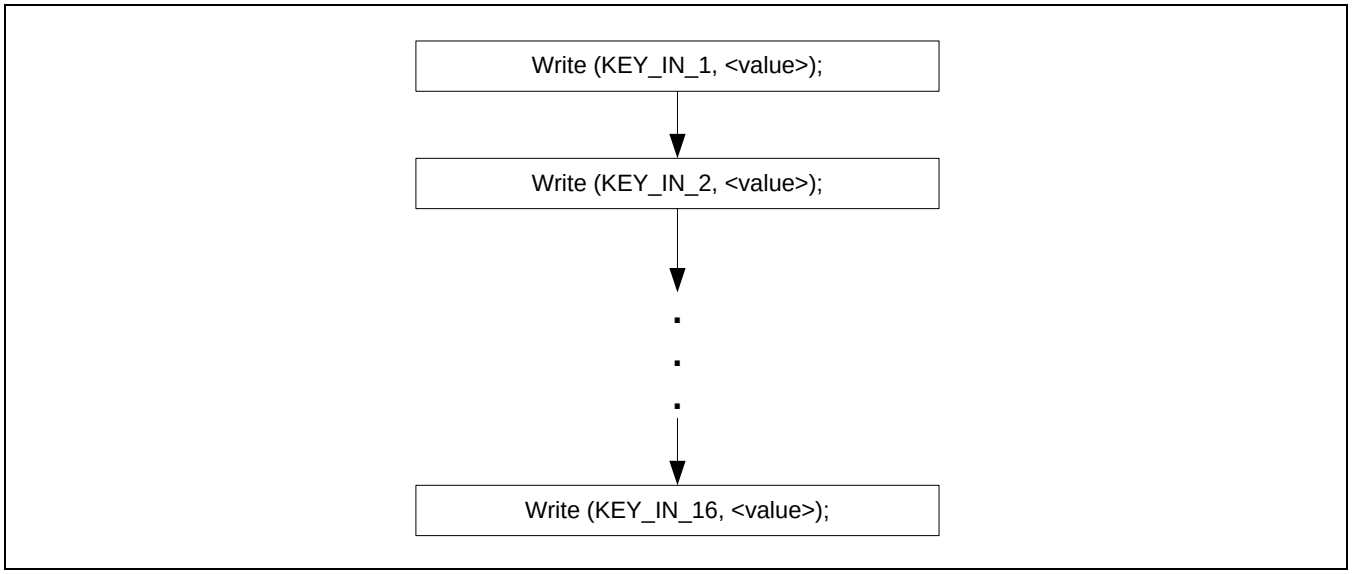


Figure 54-22 HMAC Key Setup

54.3.4.2.2 HMAC Operation

Figure 54-23 illustrates the HMAC operation.

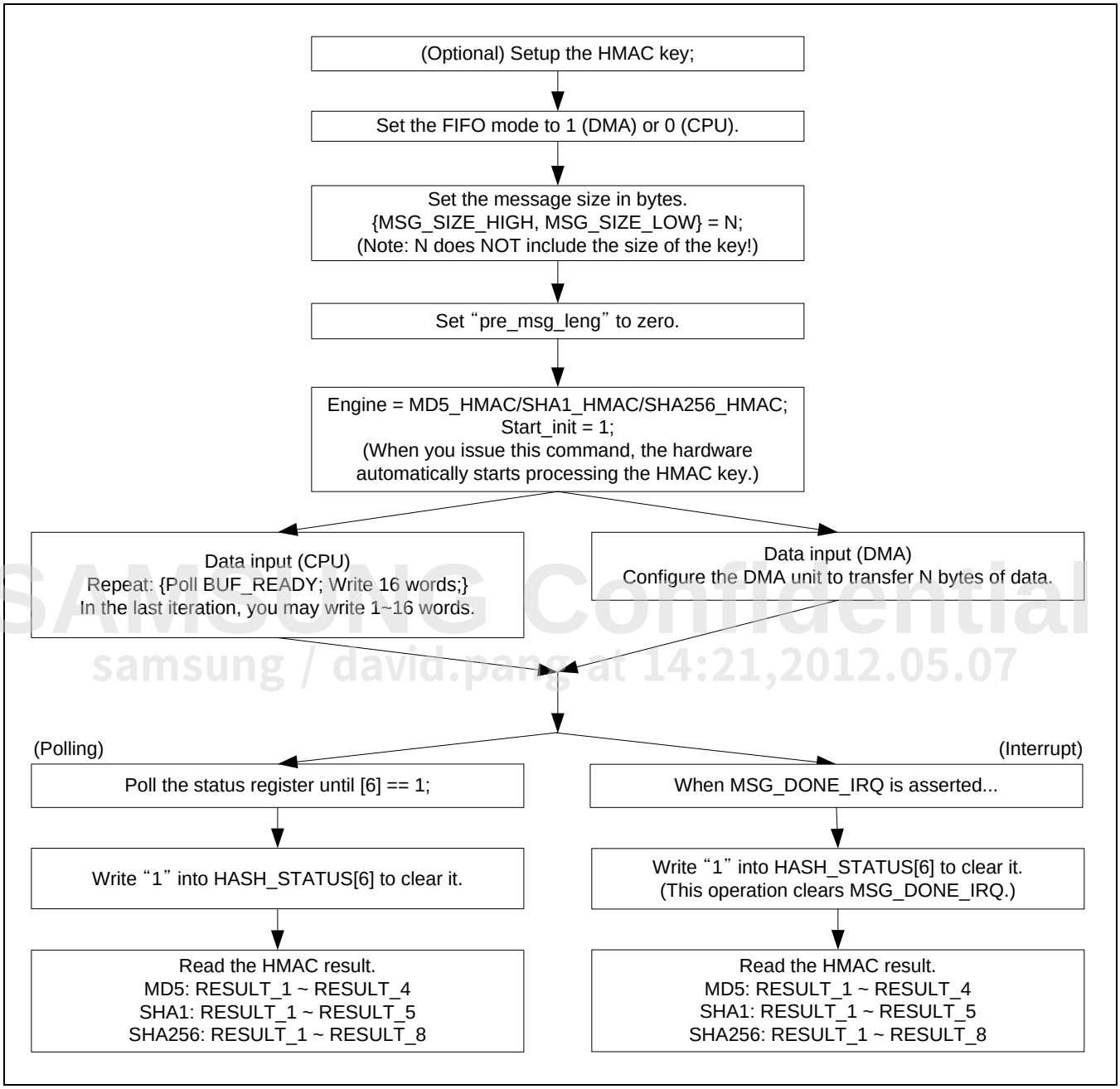


Figure 54-23 HMAC Operation

54.3.4.3 Hash with Partial Results

In *Figure 54-24*, the whole hash operation is not performed at a time. Instead, the whole message is divided into several parts (Part 1 to N) and each part is performed one by one. After hashing each part, the user has to save partial result, which will be used as IV for next part. This kind of operation pattern goes till the last part. To obtain partial result, each part (except the last one) must be a multiple of 64 bytes.

Examples of multi-part hashing include network applications such as IPSec and encrypted multimedia data.

Figure 54-24 illustrates the multi-part hashing.

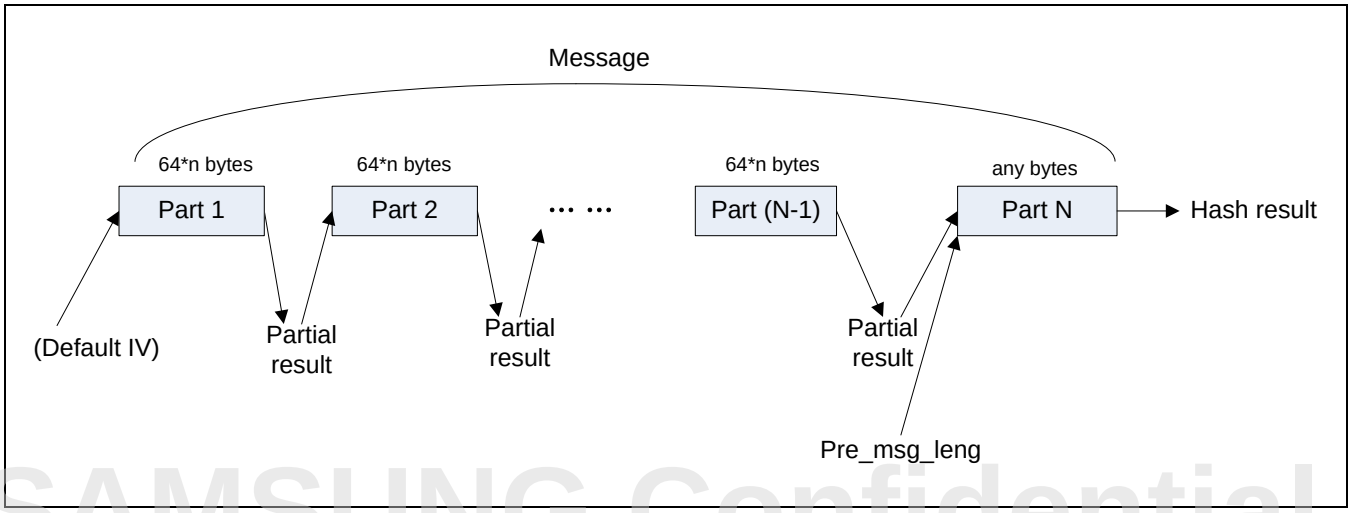


Figure 54-24 Multi-Part Hashing

Multi-part hashing is applicable when it satisfies the following conditions:

1. It is difficult to perform hash operation in one shot.
2. Each part (except the last part) of message should be a multiple of 64 bytes (64, 128, and so on). The last part may be of any bytes.

Follow these steps for Part 1 (to obtain partial result):

Figure 54-25 illustrates the steps of multi-part hashing: part 1.

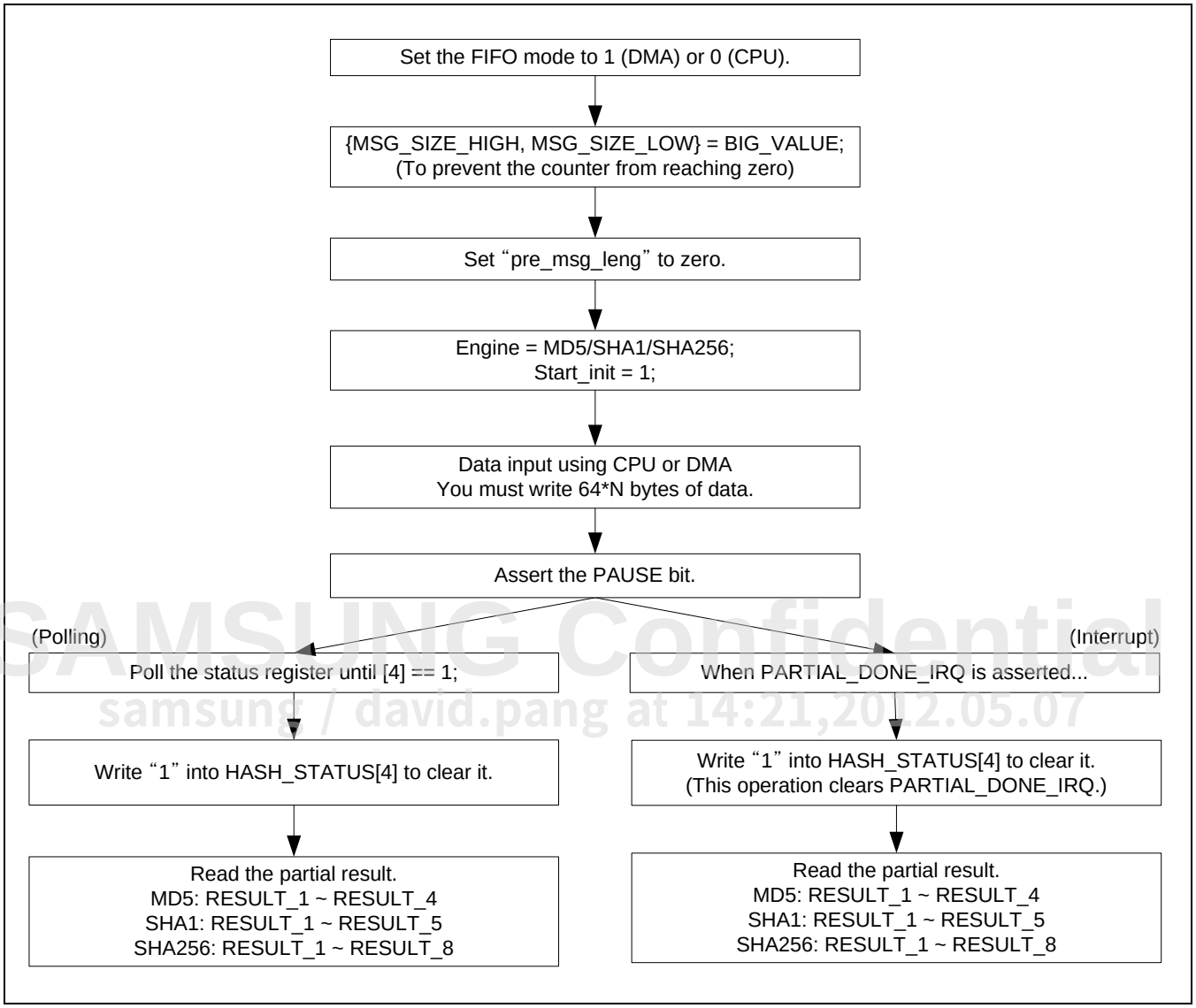


Figure 54-25 Multi-Part Hashing: Part 1

Follow these steps for Part 2 to Part (N-1) (to obtain partial result):

Figure 54-26 illustrates the steps of multi-part hashing part 2 to part (N-1).

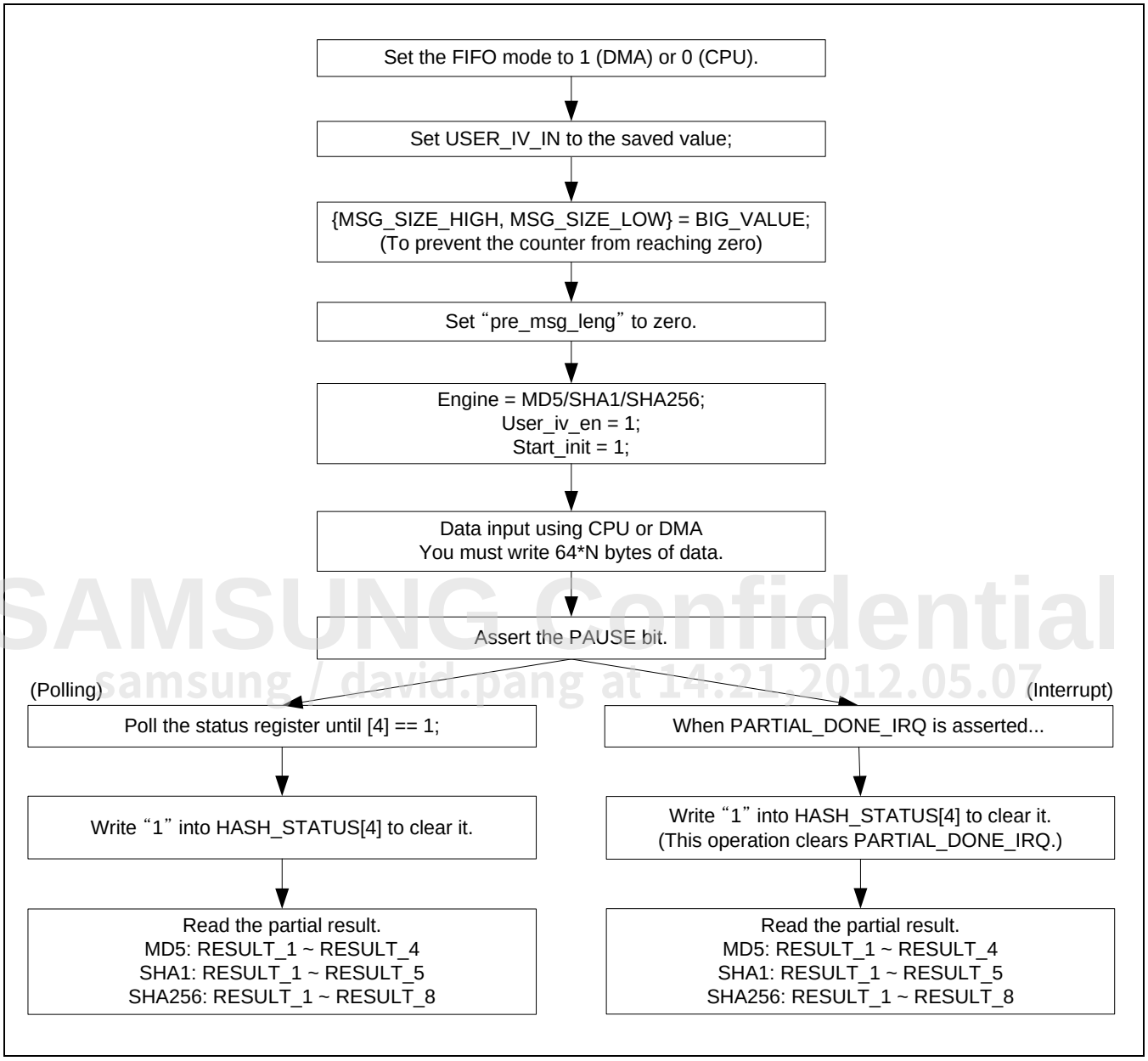


Figure 54-26 Multi-Part Hashing: Part 2 to Part (N-1)

Follow these steps for Part N (to obtain full result):

Figure 54-27 illustrates the steps of multi-part hashing part N.

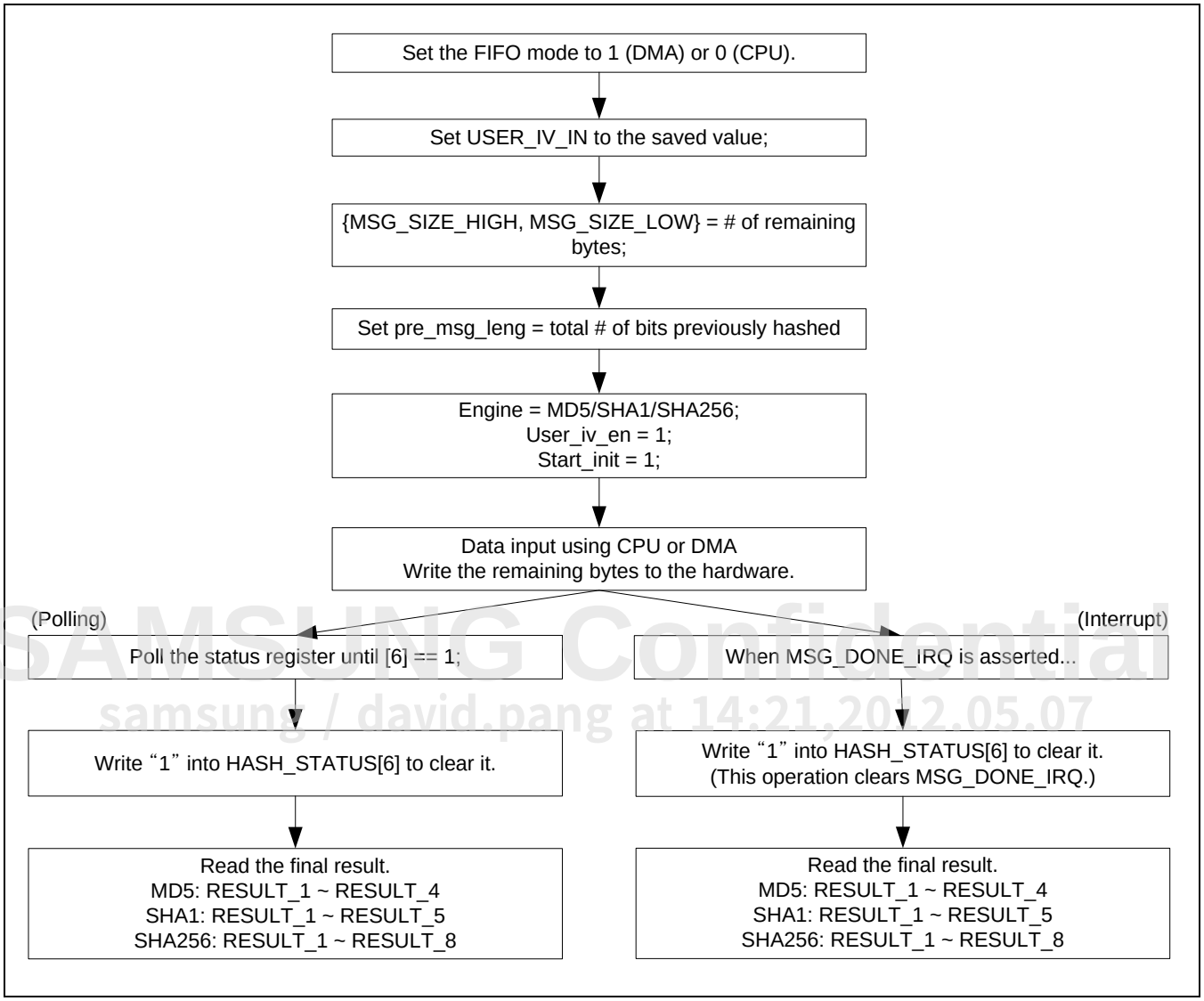


Figure 54-27 Multi-Part Hashing: Part N

54.3.4.4 HMAC with Partial Results

Similar to multi-part hashing, HMAC can be also done in a multi-part fashion, as illustrated in *Figure 54-28*. Note that:

1. In SSS_V4.0, hashing of key is automatic, which occurs as soon as the user assert START_INIT bit.
2. In SSS_V4.0, the boundary between inner hash and outer hash is hidden from the user, so that a HMAC operation is not much different from a hash operation.

Figure 54-28 illustrates the multi-part HMAC.

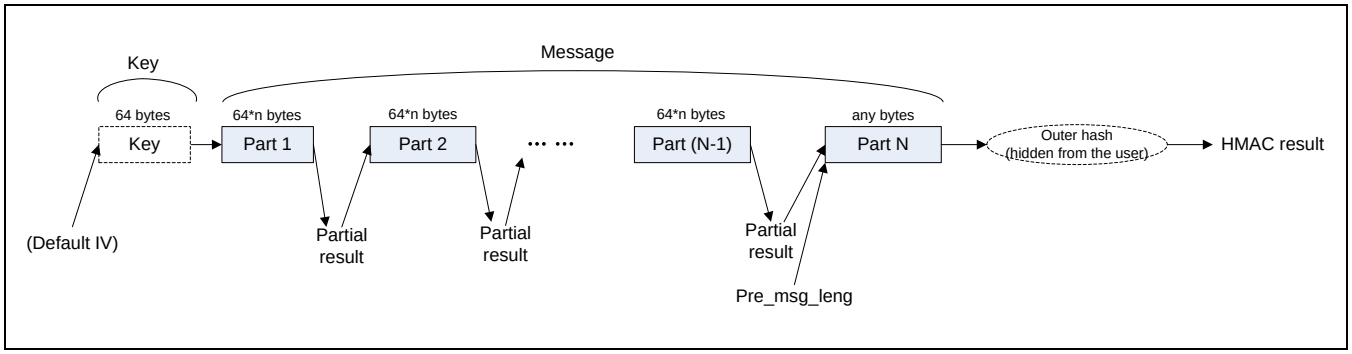


Figure 54-28 Multi-Part HMAC

Follow these steps for Part 1 (to obtain partial result):

Figure 54-29 illustrates the steps of multi-part HMAC: part 1.

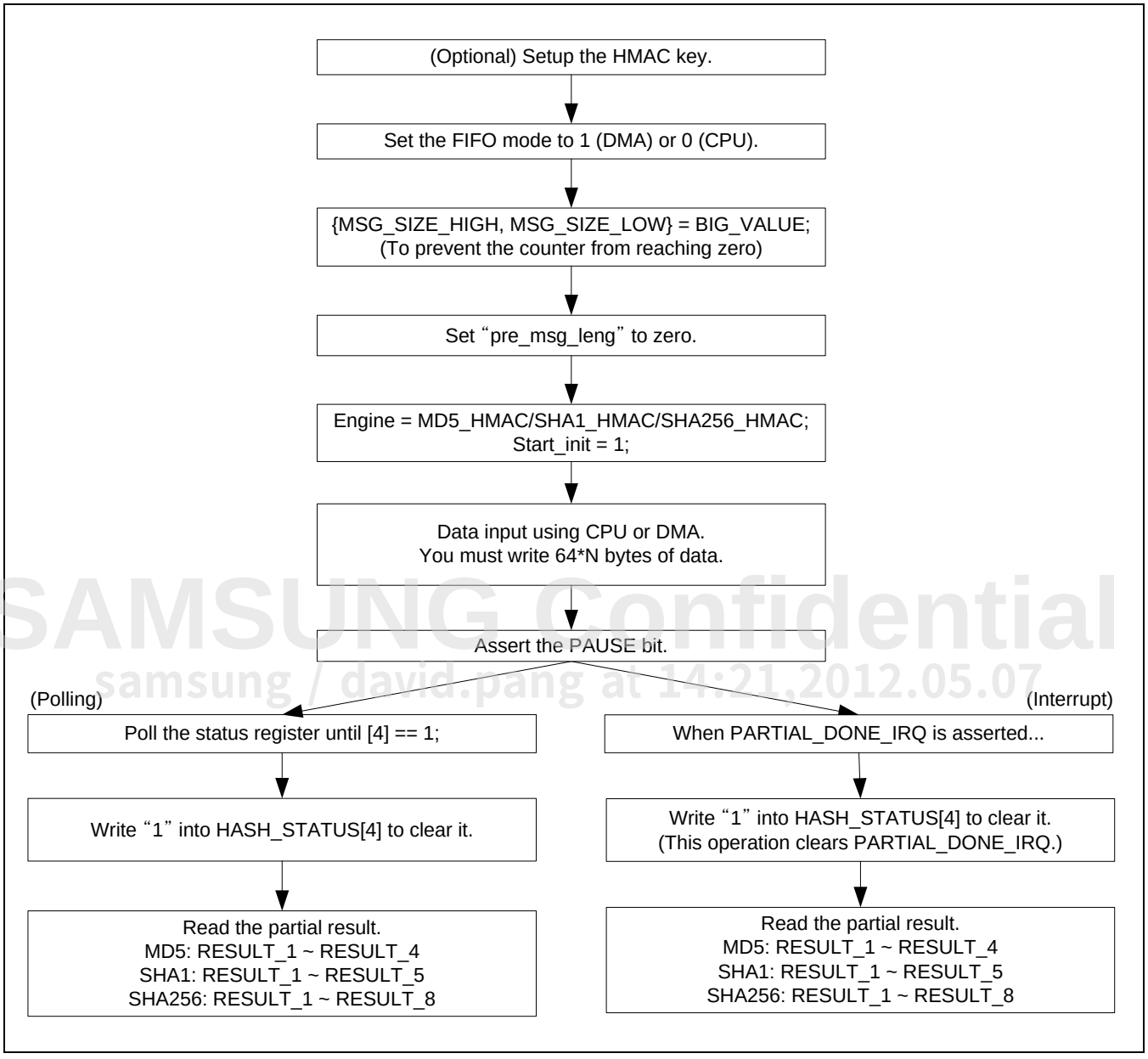


Figure 54-29 Multi-Part HMAC: Part 1

Follow these steps for Part 2 to Part (N-1) (to obtain partial result):

Figure 54-30 illustrates the steps of multi-part hashing: part 2 to part (N-1).

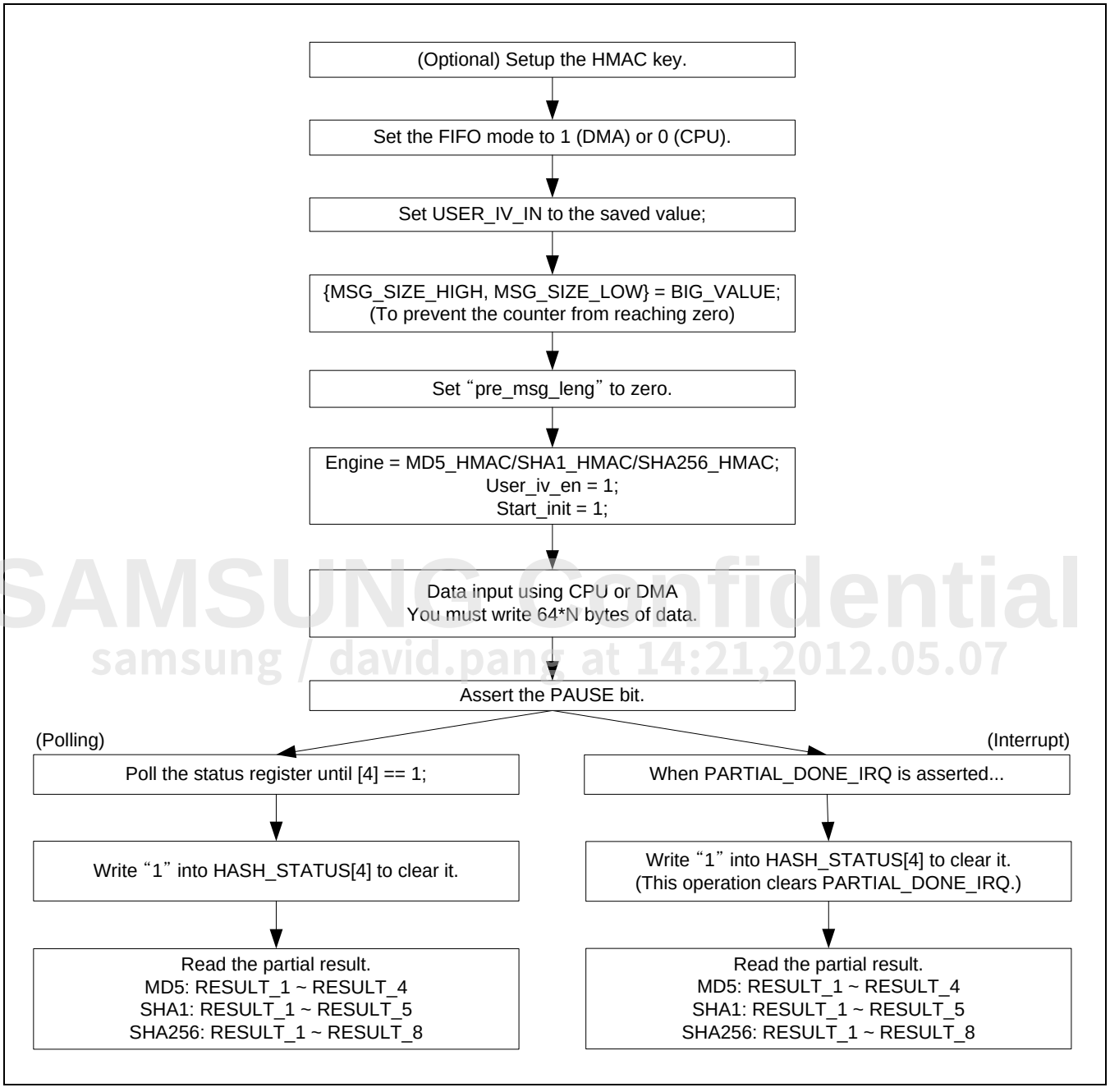


Figure 54-30 Multi-Part HMAC: Part 2 to Part (N-1)

Follow these steps for Part N (to obtain full result):

Figure 54-31 illustrates the steps of multi-hashing HMAC: part N.

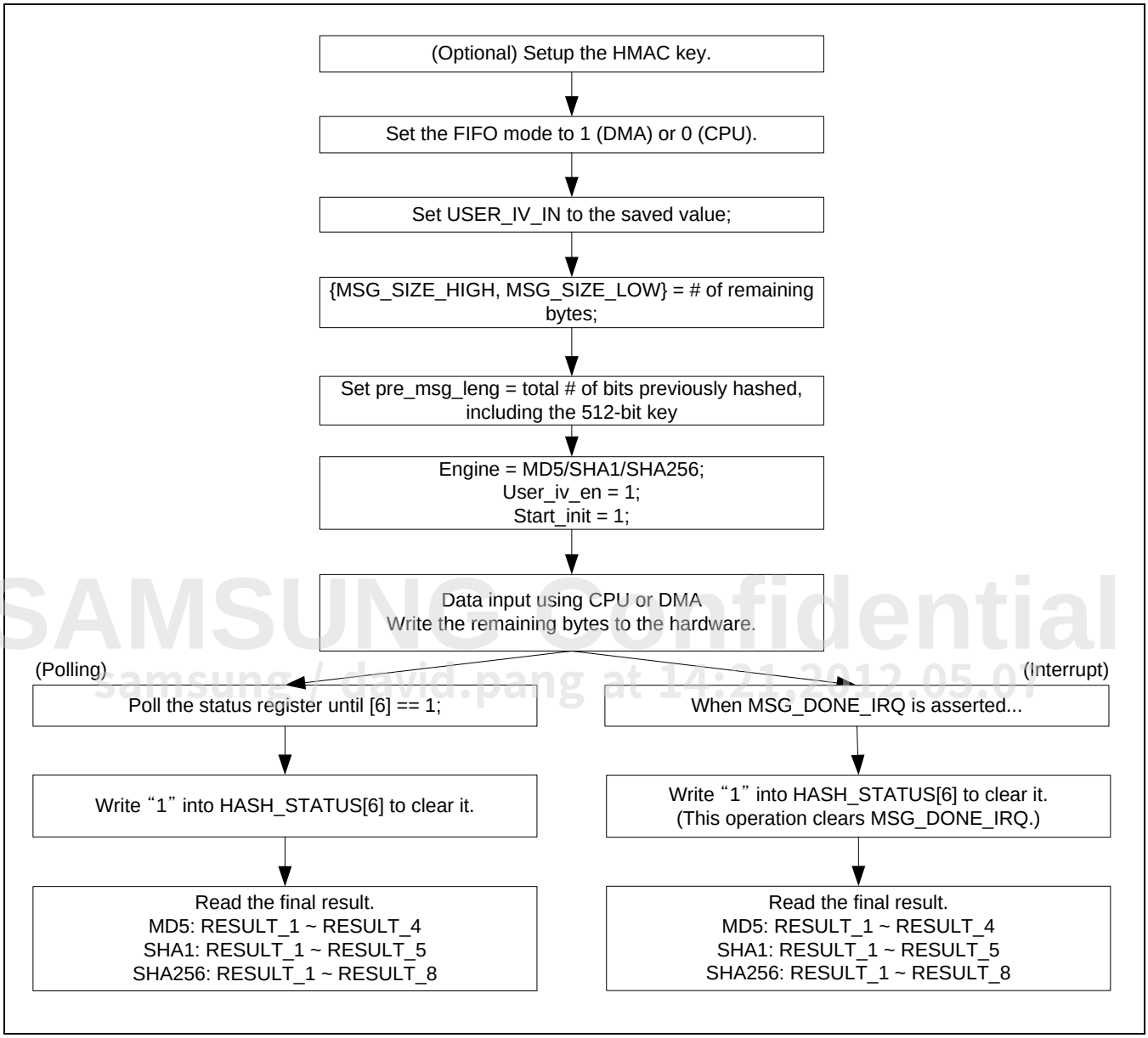


Figure 54-31 Multi-Part HMAC: Part N

54.3.4.5 PRNG

This section includes:

- Seed Initialization
- Block Cipher and Hash Together

54.3.4.5.1 Seed Initialization

Figure 54-32 illustrates the seed initialization.

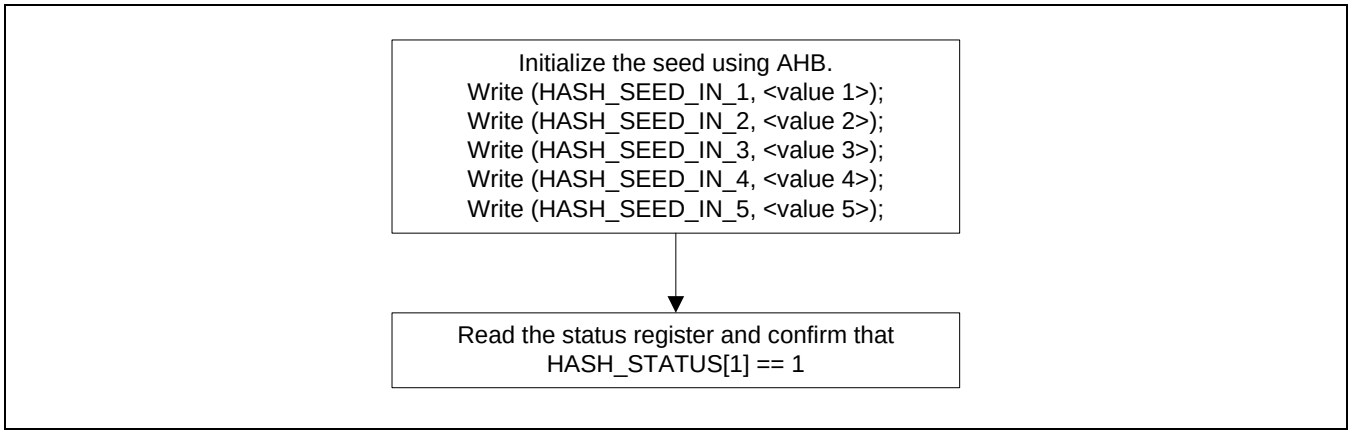


Figure 54-32 Seed Initialization

54.3.4.5.2 Generating a Random Number

Figure 54-33 illustrates the PRNG operation.

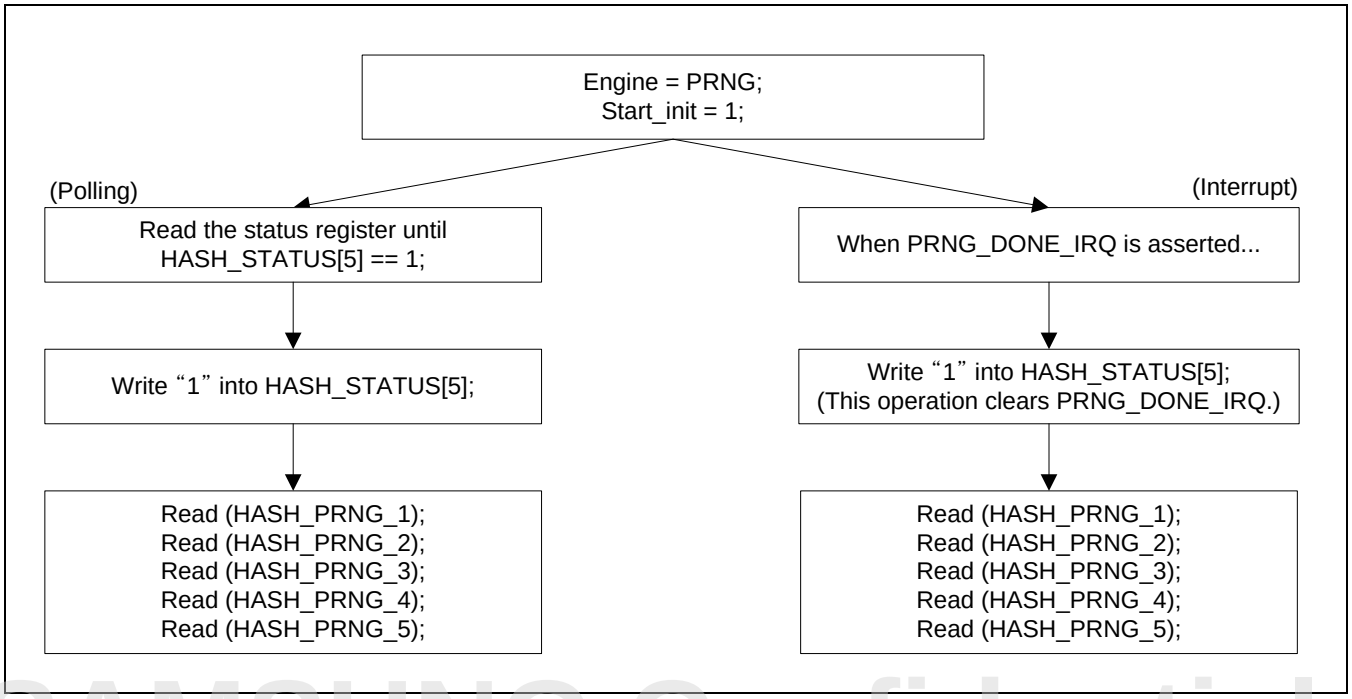


Figure 54-33 PRNG Operation

54.3.4.6 Block Cipher and Hash Together

This section includes:

- Overview
- Operation sequence

54.3.4.6.1 Overview

There are two cases where we need an encryption/decryption and a hash operation together. They are:

1. Concurrent encryption/decryption and hash
2. Encryption/decryption first, then hash

Figure 54-34 illustrates the concurrent Encryption/Decryption and Hash.

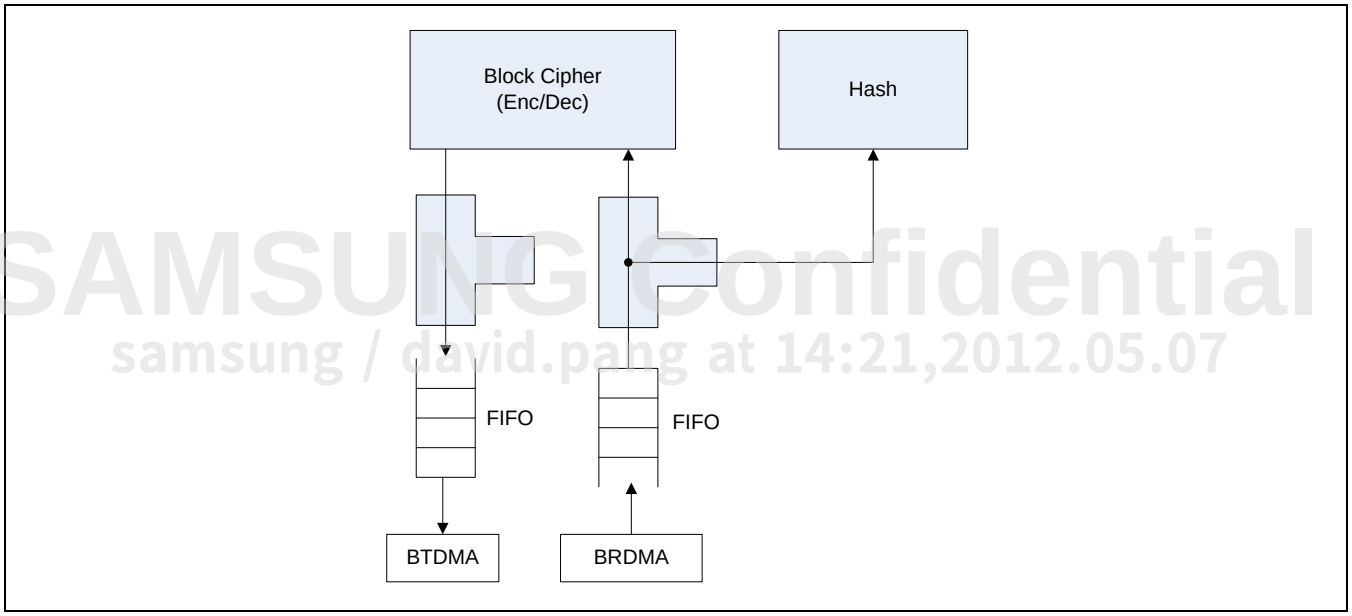


Figure 54-34 Concurrent Encryption/Decryption and Hash

Figure 54-35 illustrates the Encryption/Decryption first, then Hash.

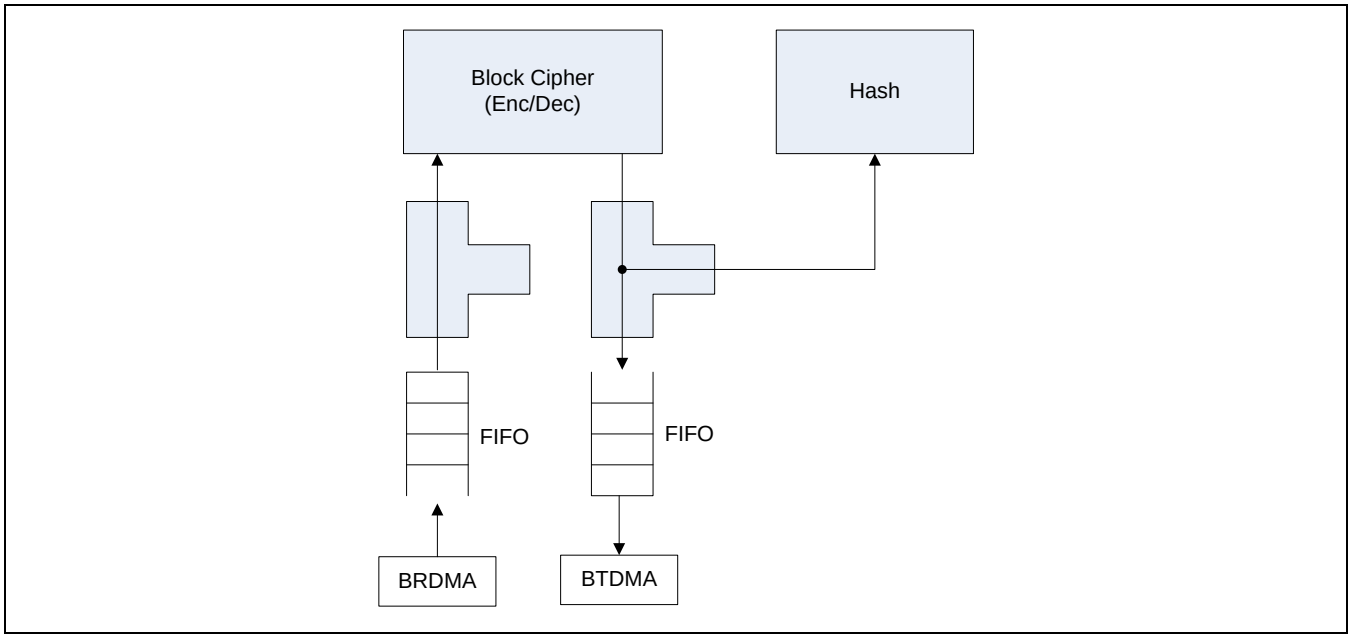


Figure 54-35 Encryption/Decryption First, Then Hash

54.3.4.6.2 Operation Sequence

Figure 54-36 illustrates the steps of operation sequence.

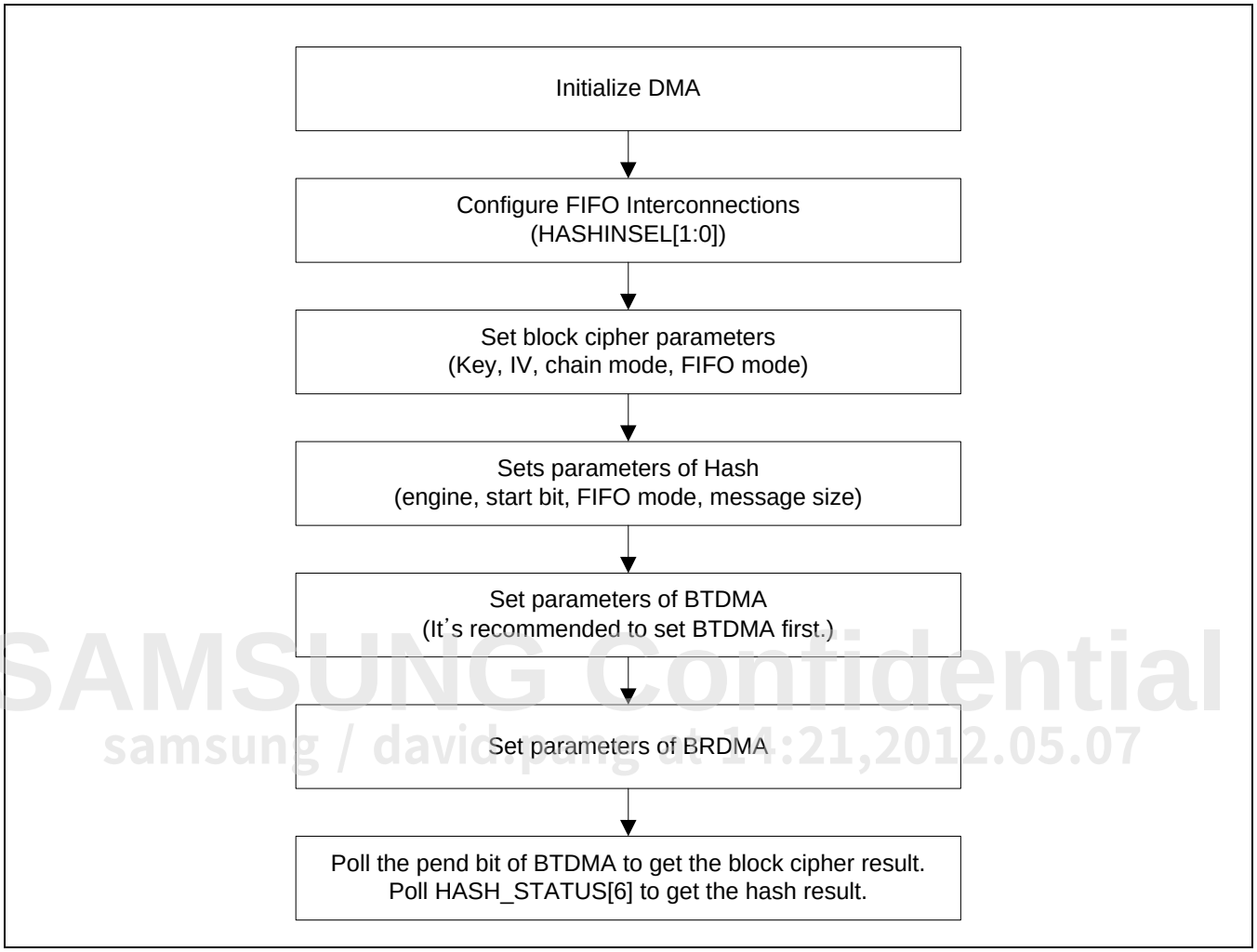


Figure 54-36 Operation Sequence

54.3.4.7 Special Cases

This section includes:

- Special cases (Hash)
- Special cases (HMAC)

54.3.4.7.1 Special Cases (Hash)

Table 54-8 describes the special cases of Hash.

Table 54-8 Special Cases (Hash)

ID	Case Description	Solution
1	NULL message	Not supported by hardware Since hash result of NULL message is constant (NOTE), the user/developer can simply hard-code these constants into software.

NOTE: SHA1 (NULL) = da39a3ee_5e6b4b0d_3255bfef_95601890_afd80709
MD5 (NULL) = d41d8cd9_8f00b204_e9800998_ecf8427e
SHA256 (NULL) = e3b0c442_98fc1c14_9afbf4c8_996fb924_27ae41e4_649b934c_a495991b_7852b855

54.3.4.7.2 Special Cases (HMAC)

Table 54-9 describes the special cases of HMAC.

Table 54-9 Special Cases (HMAC)

ID	Case Description	Solution
1	Long key (> 64 bytes)	First, process the long key using a hash engine and obtain result. • key_temp = MD5/SHA1/SHA256 (key) Then, write {key_temp, 000 … 000} into key storage.
2	NULL key only	Simply use 512'h0000_0000_0000 … 0000 as key. (In HMAC, using a NULL key is equivalent to using 512'd0.) (NOTE)
3	NULL message only	Hardware solution: Follow the steps in *Figure 54-23*. Software solution: • Key XOR 36363636 … → K1 • Hash (K1) → Temp; • Key XOR 5c5c5c5c … → K2 Hash (K2 || Temp) → Final; The speed difference is approximately the XOR operation of 512 × 2 bits of data plus α.
4	NULL key and NULL message	HMAC (NULL, NULL) = constant Hard-code these constants in software • OR Follow the method of #3 with key = 512'd0

NOTE: SHA1_HMAC (NULL, "123") = 658a0901_623568ea_5c3631cf_6193a023_d657ae4f
SHA1_HMAC (512'd0, "123") = 658a0901_623568ea_5c3631cf_6193a023_d657ae4f

Figure 54-37 illustrates the steps of operation sequence (HMAC, zero length message).

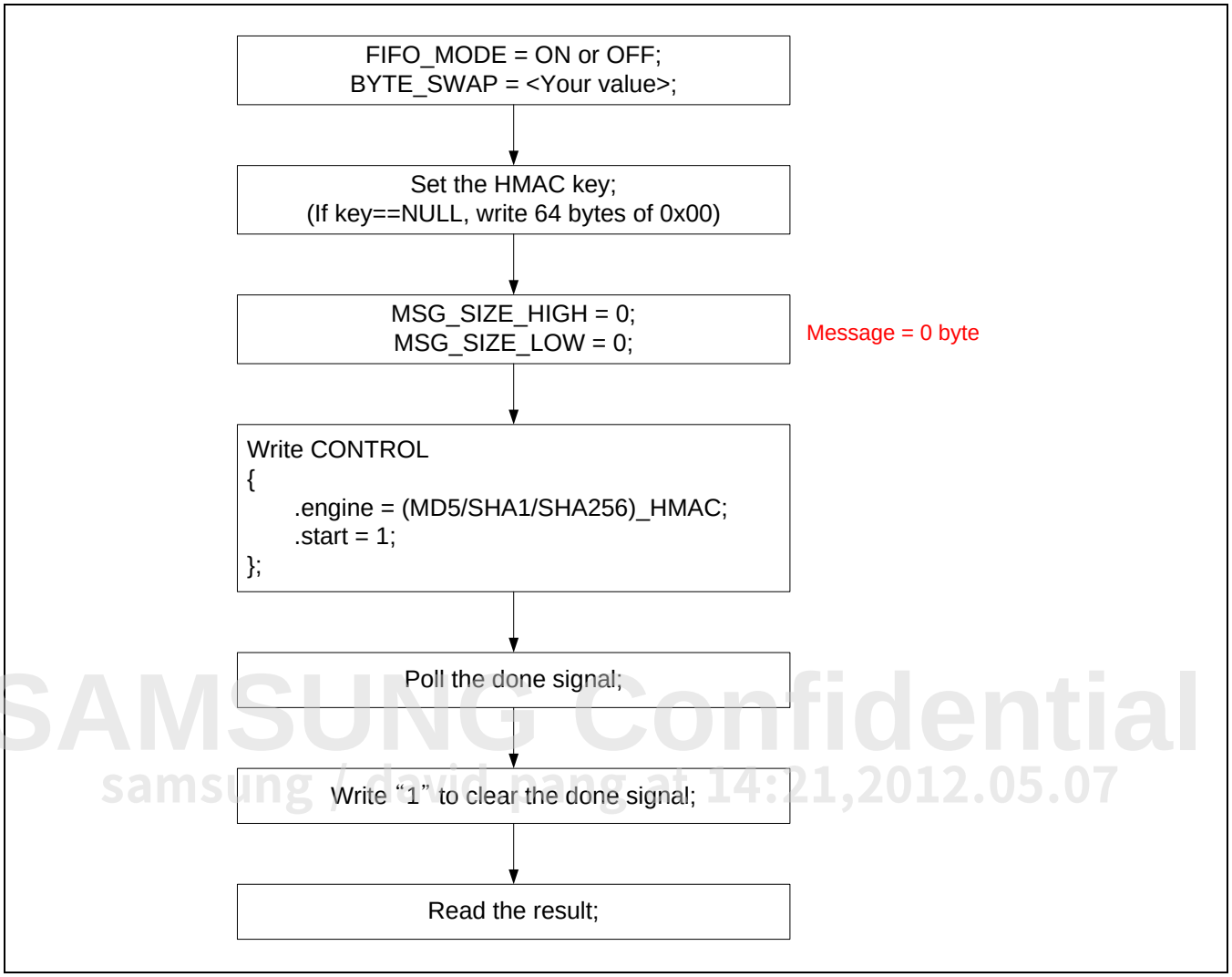


Figure 54-37 Operation Sequence (HMAC, Zero length Message)

54.3.5 PKA

This section includes:

- Register Map (SFRs of PKA)
- Operation Sequence
- Memory Map

54.3.5.1 Register Map (SFRs of PKA)

PKA has five Special Function Registers in Host interface as described in the table.

Figure 54-10 describes the PKA special function register.

Table 54-10 PKA Special Function Register (PKA_SFR) List Summary

Register	Address	RW	Description	Reset Value
PKA_SFR0	Offset + 0x00	RW	CHNK_SZ/PREC_ID	0x0000_0200
PKA_SFR1	Offset + 0x04	RW	PLDM_ON/EXEC_ON	0x0000_0000
PKA_SFR2	Offset + 0x08	RW	SEG_ID (A, B, M, S)	0x0000_0000
PKA_SFR3	Offset + 0x0C	RW	SEG_SIGN	0x0000_0000
PKA_SFR4	Offset + 0x10	RW	SEG_SIZE, FUNC_ID	0x0000_0000

Table 54-11 describes the bit field of PKA_SFR0. PKA_SFR0 controls the value of CHNK_SZ and PREC_ID.

- **CHNK_SZ:** CHNK_SZ is size of Chunk available in PKA operation.
- **PREC_ID:** PREC_ID is register that display the number of Chunk.
  If CHNK_SZ is 1111 (512 bits) and PREC_ID = 01 (double precision), the total Key Size becomes 1024-bit (512 × 2).

Table 54-11 PKA_SFR0 Register (Offset + 0x00)

Name	Bit	Type	Description
RSVD	[31:7]	–	Reserved
CHNK_SZ	[6:3]	RW	0000 = (don't use) (Default) 0001 = (don't use) 0010 = (don't use) 0011 = 128 bits 0100 = 160 bits 0101 = 192 bits 0110 = 224 bits 0111 = 256 bits 1000 = 288 bits 1001 = 320 bits 1010 = 352 bits 1011 = 384 bits 1100 = 416 bits 1101 = 448 bits 1110 = 480 bits 1111 = 512 bits
RSVD	[2]	–	Reserved
PREC_ID	[1:0]	RW	00 = Single Precision (Default) 01 = Double Precision 10 = Triple Precision 11 = Quadruple Precision

Table 54-12 describes the bit field of PKA_SFR1. PKA_SFR1 controls the value of PLDM_ON and EXEC_ON.

- **PLDM_ON:** PLDM_ON relates with pre-loading of modulus M.
- **EXEC_ON:** EXEC_ON is the register that operates PKA

Table 54-12 PKA_SFR1 Register (Offset + 0x04)

Name	Bit	Type	Description
RSVD	[31:4]	–	Reserved
PLDM_ON	[3]	RW	0 = Do not pre-load the least significant chunk of modulus M (Default) 1 = Pre-load the least significant chunk of modulus M
RSVD	[2:1]	–	Reserved
EXEC_ON	[0]	RW	0 = Do not run execution (Default) 1 = Run execution

Table 54-13 describes the bit field of PKA_SFR2.

- **A_SEG_ID:**
  - "Segment" is a partition of RAM in PKA2 operation.
  - So A_SEG_ID is memory field that stores value of input A.
  - If A_SEG_ID is 00000, input A stores in "Segment 0".
- **B_SEG_ID:**
  - B_SEG_ID stores Input B.
  - If B_SEG_ID is 00001, input B stores in Segment 1.
- **M_SEG_ID:**
  - M_SEG_ID stores Input M (modulus M).
  - If M_SEG_ID is 01111, input M stores in Segment 15.
- **S_SEG_ID:**
  - S_SEG_ID stores output S.
  - If S_SEG_ID is 10000, output S stores in Segment 16.

Table 54-13 PKA_SFR2 Register (Offset + 0x08)

Name	Bit	Type	Description
RSVD	[31:29]	–	Reserved
A_SEG_ID	[28:24]	–	00000 = Segment 0 (Default) 00001 = Segment 1 00010 = Segment 2 00011 = Segment 3 00100 = Segment 4 00101 = Segment 5 00110 = Segment 6 00111 = Segment 7 01000 = Segment 8 01001 = Segment 9 01010 = Segment 10 01011 = Segment 11 01100 = Segment 12 01101 = Segment 13 01110 = Segment 14 01111 = Segment 15 10000 = Segment 16 10001 = Segment 17 10010 = Segment 18 10011 = Segment 19 10100 = Segment 20 10101 = Segment 21 10110 = Segment 22 10111 = Segment 23 11000 = Segment 24 11001 = Segment 25 11010 = Segment 26 11011 = Segment 27

Name	Bit	Type	Description
			11100 = Segment 28 11101 = Segment 29 11110 = (don't use) 11111 = (don't use)
RSVD	[23:21]	–	Reserved
B_SEG_ID	[20:16]	–	00000 = Segment 0 (Default) 00001 = Segment 1 00010 = Segment 2 00011 = Segment 3 00100 = Segment 4 00101 = Segment 5 00110 = Segment 6 00111 = Segment 7 01000 = Segment 8 01001 = Segment 9 01010 = Segment 10 01011 = Segment 11 01100 = Segment 12 01101 = Segment 13 01110 = Segment 14 01111 = Segment 15 10000 = Segment 16 10001 = Segment 17 10010 = Segment 18 10011 = Segment 19 10100 = Segment 20 10101 = Segment 21 10110 = Segment 22 10111 = Segment 23 11000 = Segment 24 11001 = Segment 25 11010 = Segment 26 11011 = Segment 27 11100 = Segment 28 11101 = Segment 29 11110 = (don't use) 11111 = (don't use)
RSVD	[15:13]	–	Reserved
M_SEG_ID	[12:8]	–	00000 = Segment 0 (Default) 00001 = Segment 1 00010 = Segment 2 00011 = Segment 3 00100 = Segment 4 00101 = Segment 5 00110 = Segment 6 00111 = Segment 7 01000 = Segment 8 01001 = Segment 9 01010 = Segment 10 01011 = Segment 11

Name	Bit	Type	Description
			01100 = Segment 12 01101 = Segment 13 01110 = Segment 14 01111 = Segment 15 10000 = Segment 16 10001 = Segment 17 10010 = Segment 18 10011 = Segment 19 10100 = Segment 20 10101 = Segment 21 10110 = Segment 22 10111 = Segment 23 11000 = Segment 24 11001 = Segment 25 11010 = Segment 26 11011 = Segment 27 11100 = Segment 28 11101 = Segment 29 11110 = (don't use) 11111 = (don't use)
RSVD	[7:5]	–	Reserved
S_SEG_ID	[4:0]	–	00000 = Segment 0 (Default) 00001 = Segment 1 00010 = Segment 2 00011 = Segment 3 00100 = Segment 4 00101 = Segment 5 00110 = Segment 6 00111 = Segment 7 01000 = Segment 8 01001 = Segment 9 01010 = Segment 10 01011 = Segment 11 01100 = Segment 12 01101 = Segment 13 01110 = Segment 14 01111 = Segment 15 10000 = Segment 16 10001 = Segment 17 10010 = Segment 18 10011 = Segment 19 10100 = Segment 20 10101 = Segment 21 10110 = Segment 22 10111 = Segment 23 11000 = Segment 24 11001 = Segment 25 11010 = Segment 26 11011 = Segment 27 11100 = Segment 28

Name	Bit	Type	Description
			11101 = Segment 29 11110 = (don't use) 11111 = (don't use)

Table 54-14 describes the bit field of PKA_SFR3.

- **SEG_SIGN:** SEG_SIGN displays sign of memory segment that stores data.

Table 54-14 PKA_SFR3 Register (Offset + 0x0c)

Name	Bit	Type	Description
RSVD	[31:30]	–	Reserved
SEG_SIGN	[29:0]	RW	xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx0: Segment 0 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx1: Segment 0 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xx0x: Segment 1 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xx1x: Segment 1 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_x0xx: Segment 2 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_x1xx: Segment 2 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_0xxx: Segment 3 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_1xxx: Segment 3 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx0_xxxx: Segment 4 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx1_xxxx: Segment 4 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xx0x_xxxx: Segment 5 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xx1x_xxxx: Segment 5 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_x0xx_xxxx: Segment 6 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_x1xx_xxxx: Segment 6 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_0xxx_xxxx: Segment 7 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_1xxx_xxxx: Segment 7 is negative xx_xxxx_xxxx_xxxx_xxxx_xxx0_xxxx_xxxx: Segment 8 is positive xx_xxxx_xxxx_xxxx_xxxx_xxx1_xxxx_xxxx: Segment 8 is negative xx_xxxx_xxxx_xxxx_xxxx_xx0x_xxxx_xxxx: Segment 9 is positive xx_xxxx_xxxx_xxxx_xxxx_xx1x_xxxx_xxxx: Segment 9 is negative xx_xxxx_xxxx_xxxx_xxxx_x0xx_xxxx_xxxx: Segment 10 is positive xx_xxxx_xxxx_xxxx_xxxx_x1xx_xxxx_xxxx: Segment 10 is negative xx_xxxx_xxxx_xxxx_xxxx_0xxx_xxxx_xxxx: Segment 11 is positive xx_xxxx_xxxx_xxxx_xxxx_1xxx_xxxx_xxxx: Segment 11 is negative xx_xxxx_xxxx_xxxx_xxx0_xxxx_xxxx_xxxx: Segment 12 is positive xx_xxxx_xxxx_xxxx_xxx1_xxxx_xxxx_xxxx: Segment 12 is negative xx_xxxx_xxxx_xxxx_xx0x_xxxx_xxxx_xxxx: Segment 13 is positive xx_xxxx_xxxx_xxxx_xx1x_xxxx_xxxx_xxxx: Segment 13 is negative xx_xxxx_xxxx_xxxx_x0xx_xxxx_xxxx_xxxx: Segment 14 is positive xx_xxxx_xxxx_xxxx_x1xx_xxxx_xxxx_xxxx: Segment 14 is negative xx_xxxx_xxxx_xxxx_0xxx_xxxx_xxxx_xxxx: Segment 15 is positive xx_xxxx_xxxx_xxxx_1xxx_xxxx_xxxx_xxxx: Segment 15 is negative

Name	Bit	Type	Description
			xx_xxxx_xxxx_xxx0_xxxx_xxxx_xxxx_xxxx: Segment 16 is positive xx_xxxx_xxxx_xxx1_xxxx_xxxx_xxxx_xxxx: Segment 16 is negative xx_xxxx_xxxx_xx0x_xxxx_xxxx_xxxx_xxxx: Segment 17 is positive xx_xxxx_xxxx_xx1x_xxxx_xxxx_xxxx_xxxx: Segment 17 is negative xx_xxxx_xxxx_x0xx_xxxx_xxxx_xxxx_xxxx: Segment 18 is positive xx_xxxx_xxxx_x1xx_xxxx_xxxx_xxxx_xxxx: Segment 18 is negative xx_xxxx_xxxx_0xxx_xxxx_xxxx_xxxx_xxxx: Segment 19 is positive xx_xxxx_xxxx_1xxx_xxxx_xxxx_xxxx_xxxx: Segment 19 is negative xx_xxxx_xxx0_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 20 is positive xx_xxxx_xxx1_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 20 is negative xx_xxxx_xx0x_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 21 is positive xx_xxxx_xx1x_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 21 is negative xx_xxxx_x0xx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 22 is positive xx_xxxx_x1xx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 22 is negative xx_xxxx_0xxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 23 is positive xx_xxxx_1xxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 23 is negative xx_xxx0_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 24 is positive xx_xxx1_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 24 is negative xx_xx0x_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 25 is positive xx_xx1x_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 25 is negative xx_x0xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 26 is positive xx_x1xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 26 is negative xx_0xxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 27 is positive xx_1xxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 27 is negative x0_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 28 is positive x1_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 28 is negative 0x_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 29 is positive 1x_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 29 is negative

Table 54-15 describes the bit field of PKA_SFR4.

- **SEG_SIZE:**
  - SEG_SIZE displays segment size.
  - According to a SEG_SIZE, the Architecture of Memory map differs.
- **FUNC_ID:** FUNC_ID decides A × B operation or A × 1 operation.

Table 54-15 PKA_SFR4 Register (Offset + 0x10)

Name	Bit	Type	Description
RSVD	[31:7]	–	Reserved
SEG_SIZE	[6:5]	RW	00 = 256 bytes (= 2048 bits) (Default) 01 = 128 bytes (= 1024 bits) 10 = 64 bytes (= 512 bits)
RSVD	[4:1]	–	Reserved
FUNC_ID	[0]	RW	0 = Montgomery Multiplication (A by B) (Default) 1 = Montgomery Multiplication (A by 1)

54.3.5.2 Operation Sequence

This section includes:

- Register setting and Input data write
- Run

54.3.5.2.1 Register Setting and Input Data Write

PKA Control Register has five registers of PKA_SFR0 to PKA_SFR4. The sequences for operating PKA are like Step 1.

Step 1: Register Setting

The Step for Register Setting is:

1. PKA_SFR0: CHNK_SZ and PREC_ID value setting. (key length = chunk size × precision)
2. PKA_SFR1: PLDM_ON value setting. (But, EXEC_ON = 0)
3. PKA_SFR2: A_SEG_ID, B_SEG_ID, M_SEG_ID, and S_SEG_ID value setting.
4. PKA_SFR3: SEG_SIGN value setting.
5. PKA_SFR4: SEG_SIZE and FUNC_ID value setting.

Step 2: Select Segment Size

The Step for Selecting Segment Size is:

1. Select segment size within 256 byte, 128 byte, and 64 byte according to SEG_SIZE[6:5] value

Step 3: Load Input Data

The Steps to Load Input Data are:

1. Load input A into memory. If CHUNK_SIZE is 128-bit, PREC_ID is Single and SEG_SIZE is 128 byte then total key size is 128-bit.
2. Load input B into memory.
3. Load input M into memory.

Figure 54-38 illustrates the load input A to memory (CHUNK_SZ = 0011 and PREC_ID = 00).

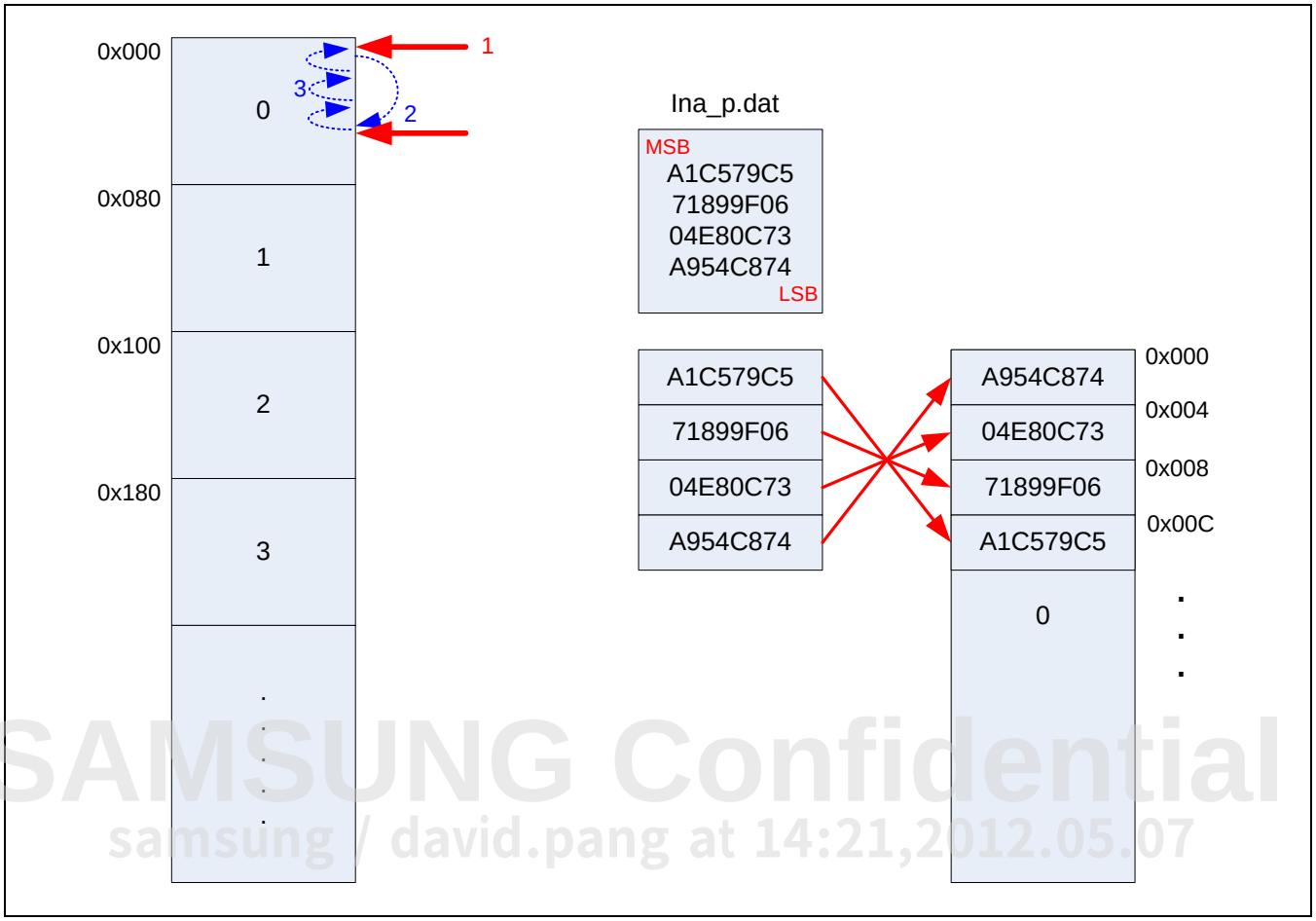


Figure 54-38 Load Input A to Memory (CHNK_SZ = 0011, PREC_ID = 00)

54.3.5.2.2 Run

PKA_SFR1: EXEC_ON[0] value setting with 1. EXEC_ON makes RUN_PKA signal value to 1 and PKA operation.

54.3.5.2.3 After Run (Result Reading)

Step 1: Result S's Sign Check

Verify SEG_SIGN of S_SEG_ID to know the output sign is positive or negative.

Step 2: Change Negative Output to Positive Value

If the output sign is negative, it has to be converted to positive.

Step 3: Compare Result S with Expected Output Value

Compare the operation result S with the Expected Output value S.

54.3.5.3 Memory Map

The memory map of PKSRAM (2 KB) is like as shown in *Figure 54-33*.

- The colored segment shows the usage during PKA internal operation. So, SEG_ID of A, B, M, and S cannot use these colored segments.

Figure 54-39 illustrates the memory map definition.

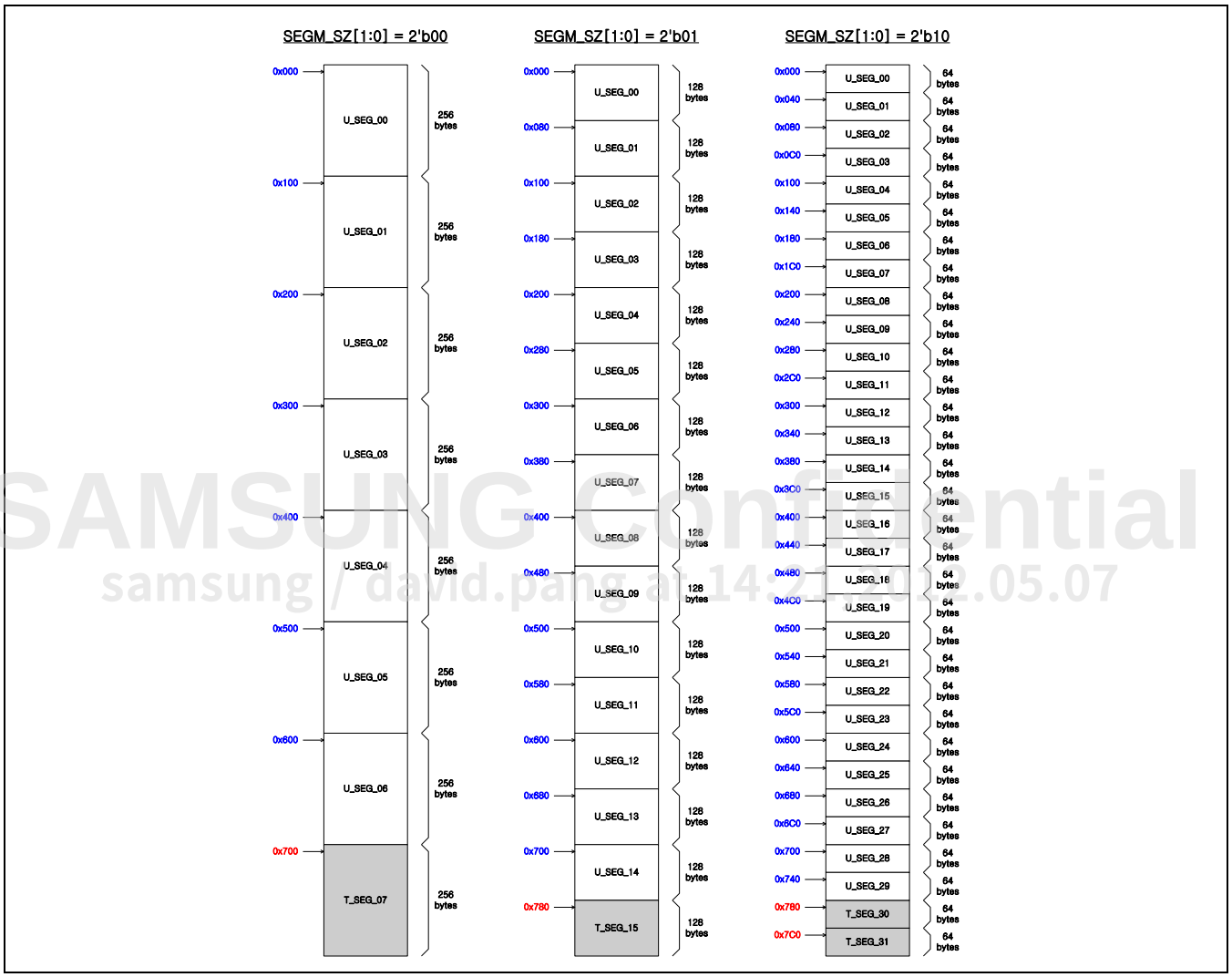


Figure 54-39 Memory Map Definition

NOTE:

1. The size of data memory is 2 Kbyte.
2. The least significant word of every operand should be stored at the lowest address of the corresponding memory segment.
3. PKA uses **little-endian byte ordering** and **little-endian word ordering**. Thus, if user's system uses big-endian byte ordering, the memory access arbiter has to perform big-to-little byte ordering conversion. The word ordering, however, cannot be changed and has to be little-endian.

54.4 Register Description

54.4.1 Register Map Summary

- Base Address: 0x1083_0000

Register	Offset	Description	Reset Value
FEED_REG_BASE			
FCINTSTAT	0x0000	Interrupt status of feeder	32'd0
FCINTENSET	0x0004	Interrupt enable set register of feeder	32'd0
FCINTENCLR	0x0008	Interrupt enable clear register of feeder	32'd0
FCINTPEND	0x000C	Pending interrupts of feeder	32'd0
FCFIFOSTAT	0x0010	FIFO status of feeder	32'h54
FCFIFOCTRL	0x0014	FIFO control of feeder	32'h0
FCGLOBAL	0x0018	Bus endian and soft reset	32'h0
FCBRDMAS	0x0020	Start address of block cipher receiving DMA	32'd0
FCBRDMAL	0x0024	Length of block cipher receiving DMA	32'd0
FCBRDMAC	0x0028	Control of block cipher receiving DMA	32'd0
FCBTDMAS	0x0030	Start address of block cipher transmitting DMA	32'd0
FCBTDMAL	0x0034	Length of block cipher transmitting DMA	32'd0
FCBTDMAC	0x0038	Control of block cipher transmitting DMA	32'd0
FCHRDMAS	0x0040	Start address of hash receiving DMA	32'd0
FCHRDMAL	0x0044	Length of hash receiving DMA	32'd0
FCHRDMAC	0x0048	Control of hash receiving DMA	32'd0

- Base Address: 0x1083_0200

Register	Offset	Description	Reset Value
AES_REG_BASE			
AES_control	0x0000	AES control register	32'h0f80
AES_status	0x0004	AES status register	32'd2
AES_indata_01	0x0010	Input data to be used in encryption/decryption Input Data[127:96]	32'd0
AES_indata_02	0x0014	Input data to be used in encryption/decryption Input Data[95:64]	32'd0
AES_indata_03	0x0018	Input data to be used in encryption/decryption Input Data[63:32]	32'd0
AES_indata_04	0x001C	Input data to be used in encryption/decryption Input Data[31:0]	32'd0
AES_outdata_01	0x0020	Output data to be used in encryption/decryption Output Data[127:96]	32'd0
AES_outdata_02	0x0024	Output data to be used in encryption/decryption Output Data[95:64]	32'd0
AES_outdata_03	0x0028	Output data to be used in encryption/decryption Output Data[63:32]	32'd0
AES_outdata_04	0x002C	Output data to be used in encryption/decryption Output Data[31:0]	32'd0
AES_ivdata_01	0x0030	Initialization vector to be used in encryption/decryption IV Data[127:96]	32'd0
AES_ivdata_02	0x0034	Initialization vector to be used in encryption/decryption IV Data[95:64]	32'd0
AES_ivdata_03	0x0038	Initialization vector to be used in encryption/decryption IV Data[63:32]	32'd0
AES_ivdata_04	0x003C	Initialization vector to be used in encryption/decryption IV Data[31:0]	32'd0
AES_cntdata_01	0x0040	Counter data to be used in encryption/decryption Counter Data[127:96]	32'd0
AES_cntdata_02	0x0044	Counter data to be used in encryption/decryption Counter Data[95:64]	32'd0
AES_cntdata_03	0x0048	Counter data to be used in encryption/decryption Counter Data[63:32]	32'd0
AES_cntdata_04	0x004C	Counter data to be used in encryption/decryption Counter Data[31:0]	32'd0
AES_keydata_01	0x0080	Key data to be used in encryption/decryption Key Data[255:224]	32'd0
AES_keydata_02	0x0084	Key data to be used in encryption/decryption Key Data[223:192]	32'd0

Register	Offset	Description	Reset Value
AES_keydata_03	0x0088	Key data to be used in encryption/decryption Key Data[191:160]	32'd0
AES_keydata_04	0x008C	Key data to be used in encryption/decryption Key Data[159:128]	32'd0
AES_keydata_05	0x0090	Key data to be used in encryption/decryption Key Data[127:96]	32'd0
AES_keydata_06	0x0094	Key data to be used in encryption/decryption Key Data[95:64]	32'd0
AES_keydata_07	0x0098	Key data to be used in encryption/decryption Key Data[63:32]	32'd0
AES_keydata_08	0x009C	Key data to be used in encryption/decryption Key Data[31:0]	32'd0

NOTE: AES_keydata_01 to AES_keydata_08:

1. In case of 128-bit key size (AES_control[5:4] == 2'b00), you must write AES_keydata_05, AES_keydata_06, AES_keydata_07, and AES_keydata_08.
2. In case of 192-bit key size (AES_control[5:4] == 2'b01), you must write AES_keydata_03, AES_keydata_04, AES_keydata_05, AES_keydata_06, AES_keydata_07, and AES_keydata_08.
3. In case of 256-bit key size (AES_control[5:4] == 2'b10), you must write AES_keydata_01, AES_keydata_02, AES_keydata_03, AES_keydata_04, AES_keydata_05, AES_keydata_06, AES_keydata_07, and AES_keydata_08.

The key size determines which key registers need to be populated. Unused keys may be ignored. The table below indicates the key requirements. "O" indicates the register must be populated.

Key Size	keydata_01	keydata_02	keydata_03	keydata_04	keydata_05	keydata_06	keydata_07	keydata_08
128-bit	–	–	–	–	O	O	O	O
192-bit	–	–	O	O	O	O	O	O
256-bit	O	O	O	O	O	O	O	O

During executing an operation, you must not write AES_control, AES_ivdata01 to 04, AES_cntdata_01 to 04, and AES_keydata_01 to 08 registers. Also, it is not possible to configure the next AES operation (packet) in advance.

- Base Address: 0x1083_0300

Register	Offset	Description	Reset Value
TDES_REG_BASE			
TDES_CONF	0x0000	TDES configuration register	32'h3C0
TDES_STAT	0x0004	TDES status register	32'd2
TDES_KEY1_0	0x0010	TDES key 1: [63:32]	32'd0
TDES_KEY1_1	0x0014	TDES key 1: [31:0]	32'd0
TDES_KEY2_0	0x0018	TDES key 2: [63:32]	32'd0
TDES_KEY2_1	0x001C	TDES key 2: [31:0]	32'd0
TDES_KEY3_0	0x0020	TDES key 3: [63:32]	32'd0
TDES_KEY3_1	0x0024	TDES key 3: [31:0]	32'd0
TDES_IV_0	0x0028	TDES initial vector: [63:32]	32'd0
TDES_IV_1	0x002C	TDES initial vector: [31:0]	32'd0
TDES_INPUT_0	0x0030	TDES input data: [63:32]	32'd0
TDES_INPUT_1	0x0034	TDES input data: [31:0]	32'd0
TDES_OUTPUT_0	0x0038	TDES output data: [63:32]	32'd0
TDES_OUTPUT_1	0x003C	TDES output data: [31:0]	32'd0

- Base Address: 0x1083_0400

Register	Offset	Description	Reset Value
HASH_REG_BASE			
HASH_CONTROL_1	0x0000	Hash control register 1	32'd0
HASH_CONTROL_2	0x0004	Hash control register 2	32'd0
HASH_FIFO_MODE_EN	0x0008	FIFO mode enable/disable	32'd0
HASH_BYTE_SWAP	0x000C	Byte swap configuration register	32'h0000_000F
HASH_STATUS	0x0010	Status register	32'h0000_0001
HASH_MSG_SIZE_LOW	0x0020	Message size in bytes (lower 32 bits)	32'd0
HASH_MSG_SIZE_HIGH	0x0024	Message size in bytes (higher 32 bits)	32'd0
HASH_PRE_MSG_LENG_LOW	0x0028	Pre message length in bits (lower 32 bits)	32'd0
HASH_PRE_MSG_LENG_HIGH	0x002C	Pre message length in bits (higher 32 bits)	32'd0
HASH_DATA_IN_1	0x0030	Message input register 1	32'd0
HASH_DATA_IN_2	0x0034	Message input register 2	32'd0
HASH_DATA_IN_3	0x0038	Message input register 3	32'd0
HASH_DATA_IN_4	0x003C	Message input register 4	32'd0
HASH_DATA_IN_5	0x0040	Message input register 5	32'd0
HASH_DATA_IN_6	0x0044	Message input register 6	32'd0
HASH_DATA_IN_7	0x0048	Message input register 7	32'd0

Register	Offset	Description	Reset Value
HASH_DATA_IN_8	0x004C	Message input register 8	32'd0
HASH_DATA_IN_9	0x0050	Message input register 9	32'd0
HASH_DATA_IN_10	0x0054	Message input register 10	32'd0
HASH_DATA_IN_11	0x0058	Message input register 11	32'd0
HASH_DATA_IN_12	0x005C	Message input register 12	32'd0
HASH_DATA_IN_13	0x0060	Message input register 13	32'd0
HASH_DATA_IN_14	0x0064	Message input register 14	32'd0
HASH_DATA_IN_15	0x0068	Message input register 15	32'd0
HASH_DATA_IN_16	0x006C	Message input register 16	32'd0
HASH_HMAC_KEY_IN_1	0x0070	HMAC key input register 1 (KEY[511:480])	32'd0
HASH_HMAC_KEY_IN_2	0x0074	HMAC key input register 2	32'd0
HASH_HMAC_KEY_IN_3	0x0078	HMAC key input register 3	32'd0
HASH_HMAC_KEY_IN_4	0x007C	HMAC key input register 4	32'd0
HASH_HMAC_KEY_IN_5	0x0080	HMAC key input register 5	32'd0
HASH_HMAC_KEY_IN_6	0x0084	HMAC key input register 6	32'd0
HASH_HMAC_KEY_IN_7	0x0088	HMAC key input register 7	32'd0
HASH_HMAC_KEY_IN_8	0x008C	HMAC key input register 8	32'd0
HASH_HMAC_KEY_IN_9	0x0090	HMAC key input register 9	32'd0
HASH_HMAC_KEY_IN_10	0x0094	HMAC key input register 10	32'd0
HASH_HMAC_KEY_IN_11	0x0098	HMAC key input register 11	32'd0
HASH_HMAC_KEY_IN_12	0x009C	HMAC key input register 12	32'd0
HASH_HMAC_KEY_IN_13	0x00A0	HMAC key input register 13	32'd0
HASH_HMAC_KEY_IN_14	0x00A4	HMAC key input register 14	32'd0
HASH_HMAC_KEY_IN_15	0x00A8	HMAC key input register 15	32'd0
HASH_HMAC_KEY_IN_16	0x00AC	HMAC key input register 16 (KEY[31:0])	32'd0
HASH_USER_IV_IN_1	0x00B0	User IV input register 1	32'd0
HASH_USER_IV_IN_2	0x00B4	User IV input register 2	32'd0
HASH_USER_IV_IN_3	0x00B8	User IV input register 3	32'd0
HASH_USER_IV_IN_4	0x00BC	User IV input register 4	32'd0
HASH_USER_IV_IN_5	0x00C0	User IV input register 5	32'd0
HASH_USER_IV_IN_6	0x00C4	User IV input register 6	32'd0
HASH_USER_IV_IN_7	0x00C8	User IV input register 7	32'd0
HASH_USER_IV_IN_8	0x00CC	User IV input register 8	32'd0
HASH_RESULT_1	0x0100	Hash/HMAC/Partial result 1	32'd0
HASH_RESULT_2	0x0104	Hash/HMAC/Partial result 2	32'd0
HASH_RESULT_3	0x0108	Hash/HMAC/Partial result 3	32'd0
HASH_RESULT_4	0x010C	Hash/HMAC/Partial result 4	32'd0

Register	Offset	Description	Reset Value
HASH_RESULT_5	0x0110	Hash/HMAC/Partial result 5	32'd0
HASH_RESULT_6	0x0114	Hash/HMAC/Partial result 6	32'd0
HASH_RESULT_7	0x0118	Hash/HMAC/Partial result 7	32'd0
HASH_RESULT_8	0x011C	Hash/HMAC/Partial result 8	32'd0
HASH_SEED_IN_1	0x0140	PRNG seed input 1	32'd0
HASH_SEED_IN_2	0x0144	PRNG seed input 2	32'd0
HASH_SEED_IN_3	0x0148	PRNG seed input 3	32'd0
HASH_SEED_IN_4	0x014C	PRNG seed input 4	32'd0
HASH_SEED_IN_5	0x0150	PRNG seed input 5	32'd0
HASH_PRNG_1	0x0160	PRNG result 1	32'd0
HASH_PRNG_2	0x0164	PRNG result 2	32'd0
HASH_PRNG_3	0x0168	PRNG result 3	32'd0
HASH_PRNG_4	0x016C	PRNG result 4	32'd0
HASH_PRNG_5	0x0170	PRNG result 5	32'd0

- Base Address: 0x1083_0700

Register	Offset	Description	Reset Value
PKA_REG_BASE			
PKA_SFR0	0x0000	CHNK_SZ/PREC_ID	32'd0
PKA_SFR1	0x0004	PLDM_ON/EXEC_ON	32'd0
PKA_SFR2	0x0008	SEG_ID (A, B, M, S)	32'd0
PKA_SFR3	0x000C	SEG_SIGN	32'd0
PKA_SFR4	0x0010	SEG_SIZE, FUNC_ID	32'd0

- Base Address: 0x1083_0800

Register	Offset	Description	Reset Value
PKA_SRAM_BASE			
PKA_SRAM	0x0000 to 0x0800	PKA Memory	Undefined

54.4.2 Feeder Hardware Register Base

54.4.2.1 FCINTSTAT

- Base Address: 0x1083_0000
- Address = Base Address + 0x0000, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	24'd0
PARTIAL_DONE	[7]	R	Interrupt signal of Hash that informs end of partial operation.	1'b0
PRNG_DONE	[6]	R	Interrupt signal of PRNG that informs end of PRNG operation.	1'b0
MSG_DONE	[5]	R	Interrupt signal of Hash that informs end of normal hashing operation.	1'b0
PRNG_ERROR	[4]	R	Interrupt signal of PRNG that informs abnormal access such as getting random number before seed setting.	1'b0
BRDMA_DONE	[3]	R	Interrupt signal of Block cipher Receiving DMA that informs end of DMA operation.	1'd0
BTDMA_DONE	[2]	R	Interrupt signal of Block cipher Transmitting DMA that informs end of DMA operation.	1'd0
HRDMA_DONE	[1]	R	Interrupt signal of Hash Receiving DMA that informs end of DMA operation.	1'd0
RSVD	[0]	–	Reserved	1'd0

54.4.2.2 FCINTENSET

- Base Address: 0x1083_0000
- Address = Base Address + 0x0004, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	24'd0
PARTIAL_DONE	[7]	RW	Enables Interrupt bit set signal of PARTIAL_DONE	1'b0
PRNG_DONE	[6]	RW	Enables Interrupt bit set signal of PRNG_DONE	1'b0
MSG_DONE	[5]	RW	Enables Interrupt bit set signal of MSG_DONE	1'b0
PRNG_ERROR	[4]	RW	Enables Interrupt bit set signal of PRNG_ERROR	1'b0
BRDMA_DONE	[3]	RW	Enables Interrupt bit set signal of BRDMA_DONE	1'd0
BTDMA_DONE	[2]	RW	Enables Interrupt bit set signal of BTDMA_DONE	1'd0
HRDMA_DONE	[1]	RW	Enables Interrupt bit set signal of HRDMA_DONE	1'd0
RSVD	[0]	–	Reserved	1'd0

54.4.2.3 FCINTENCLR

- Base Address: 0x1083_0000
- Address = Base Address + 0x0008, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	24'd0
PARTIAL_DONE	[7]	RW	Enables Interrupt bit clear signal of PARTIAL_DONE	1'b0
PRNG_DONE	[6]	RW	Enables Interrupt bit clear signal of PRNG_DONE	1'b0
MSG_DONE	[5]	RW	Enables Interrupt bit clear signal of MSG_DONE	1'b0
PRNG_ERROR	[4]	RW	Enables Interrupt bit clear signal of PRNG_ERROR	1'b0
BRDMA_DONE	[3]	RW	Enables Interrupt bit clear signal of BRDMA_DONE	1'd0
BTDMA_DONE	[2]	RW	Enables Interrupt bit clear signal of BTDMA_DONE	1'd0
HRDMA_DONE	[1]	RW	Enables Interrupt bit clear signal of HRDMA_DONE	1'd0
RSVD	[0]	–	Reserved	1'd0

54.4.2.4 FCINTPEND

- Base Address: 0x1083_0000
- Address = Base Address + 0x000C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	24'd0
PARTIAL_DONE	[7]	R	Raw interrupt signal of PARTIAL_DONE. This bit can only be cleared by writing "1" to the bit of HASH_STATUS register.	1'b0
PRNG_DONE	[6]	R	Raw interrupt signal of PRNG_DONE. This bit can only be cleared by writing "1" to the bit of HASH_STATUS register.	1'b0
MSG_DONE	[5]	R	Raw interrupt signal of MSG_DONE. This bit can only be cleared by writing "1" to the bit of HASH_STATUS register.	1'b0
PRNG_ERROR	[4]	R	Raw interrupt signal of PRNG_ERROR. This bit can only be cleared by a complete seed setup operation.	1'b0
BRDMA_DONE	[3]	RW	Raw interrupt signal of BRDMA_DONE. This bit can only be cleared by writing "1" to this bit.	1'd0
BTDMA_DONE	[2]	RW	Raw interrupt signal of BTDMA_DONE. This bit can only be cleared by writing "1" to this bit.	1'd0
HRDMA_DONE	[1]	RW	Raw interrupt signal of HRDMA_DONE. This bit can only be cleared by writing "1" to this bit.	1'd0
RSVD	[0]	–	Reserved	1'd0

54.4.2.5 FCFIFOSTAT

- Base Address: 0x1083_0000
- Address = Base Address + 0x0010, Reset Value = 32'h54

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	24'd0
BRFIFOFUL	[7]	R	Full state of Block cipher Receiving FIFO	1'd0
BRFIFOEMP	[6]	R	Empty state of Block cipher Receiving FIFO	1'd1
BTFIFOFUL	[5]	R	Full state of Block cipher Transmitting FIFO	1'd0
BTFIFOEMP	[4]	R	Empty state of Block cipher Transmitting FIFO	1'd1
HRFIFOFUL	[3]	R	Full state of Hash Receiving FIFO	1'd0
HRFIFOEMP	[2]	R	Empty state of Hash Receiving FIFO	1'd1
RSVD	[1:0]	–	Reserved	2'd0

54.4.2.6 FCFIFOCTRL

- Base Address: 0x1083_0000
- Address = Base Address + 0x0014, Reset Value = 32'h0

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	29'd0
DESSEL	[2]	RW	Destination block cipher of FIFO. AES (= 0)/DES (= 1)	1'd0
HASHINSEL	[1:0]	RW	Data from independent source (= 0) Data from block cipher input (= 1) Data from block cipher output (= 2) Reserved (= 3)	2'd0

54.4.2.7 FCGLOBAL

- Base Address: 0x1083_0000
- Address = Base Address + 0x0018, Reset Value = 32'h0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	24'd0
AXI_ENDIAN	[7:6]	RW	Endianness of AXI interface. 2'b00 = Little Endian (Default) 2'b01 = Byte-Invariant Big Endian 2'b10 = Word-Invariant Big Endian	2'd0
RSVD	[5]	–	Reserved	1'd0
HASH_RESET	[4]	RW	Soft reset for Hash and PRNG	1'd0
DES_RESET	[3]	RW	Soft reset for DES and 3DES	1'd0
AES_RESET	[2]	RW	Soft reset for AES	1'd0
DMA_RESET	[1]	RW	Soft reset for DMA	1'd0
SSS_RESET	[0]	RW	Soft reset for whole SSS. This soft reset asserts all above soft resets.	1'd0

54.4.2.8 FCBRDMAS

- Base Address: 0x1083_0000
- Address = Base Address + 0x0020, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
STARTADDR	[31:0]	RW	Start Address of DMA. The address does not need to be aligned by 32 bit. Its value increases by 8 after every transfer.	32'd0

54.4.2.9 FCBRDMAL

- Base Address: 0x1083_0000
- Address = Base Address + 0x0024, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
LENGTH	[31:0]	RW	Block length of DMA. The length needs to be aligned by 64-bit (DES) or 128 bit (AES). The three least significant bits are ignored. Its value decreases by 8 after every transfer. If you set this register with a number (>=8), DMA starts immediately.	32'd0

54.4.2.10 FCBRDMAC

- Base Address: 0x1083_0000
- Address = Base Address + 0x0028, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	23'd0
ARCACHE	[8:5]	RW	ARCACHE value of AR channel of AXI	4'd0
ARPROT	[4:2]	RW	ARPROT value of AR channel of AXI	3'd0
BYTESWAP	[1]	RW	If this bit is set to high, then the data read from the bus is byte-swapped in a 64-bit boundary. If this bit is low (default), then the data is handed over to FIFO without byte-swap. Normally this bit should be "0".	1'b0
FLUSH	[0]	RW	If this bit is set to high, then data flushes from FIFO and DMA. This bit clears automatically.	1'd0

54.4.2.11 FCBTDMAS

- Base Address: 0x1083_0000
- Address = Base Address + 0x0030, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
STARTADDR	[31:0]	RW	Start Address of DMA. The address does not need to be aligned by 32-bit. Its value increases by 8 after every transaction.	32'd0

54.4.2.12 FCBTDMAL

- Base Address: 0x1083_0000, Address = Base Address + 0x0034, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
LENGTH	[31:0]	RW	Block length of DMA. The length needs to be aligned by 64-bit (DES) or 128-bit (AES). The three least significant bits are ignored. Its value decreases by 8 after every transfer. If you set this register with a number (>=8), DMA starts immediately.	32'd0

54.4.2.13 FCBTDMAC

- Base Address: 0x1083_0000
- Address = Base Address + 0x0038, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	23'd0
ARCACHE	[8:5]	RW	AWCACHE value of Write-Address (AW) channel of AXI	4'd0
ARPROT	[4:2]	RW	AWPROT value of AW channel of AXI	3'd0
BYTESWAP	[1]	RW	If this bit is set to high, then the data read from the bus is byte-swapped in a 64-bit boundary. If this bit is low (default), then the data is handed over to the FIFO without byte-swap. Normally this bit should be "0".	1'b0
FLUSH	[0]	RW	If this bit is set to high, then data flushes from FIFO and DMA. This bit clears automatically.	1'd0

54.4.2.14 FCHRDMAS

- Base Address: 0x1083_0000
- Address = Base Address + 0x0040, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
STARTADDR	[31:0]	RW	Start Address of DMA. The address does not need to be aligned by 32-bit. Its value increases by 8 after every transaction.	32'd0

54.4.2.15 FCHRDMAL

- Base Address: 0x1083_0000
- Address = Base Address + 0x0044, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
LENGTH	[31:0]	RW	Block length of DMA. The length does not need to be aligned. If the three least significant bits are not zero, the LSB 3 bits are rounded up to 8. For example, if we set it with 13, it will become 16 automatically. Its value decreases by 8 after every transfer. If you set this register with a number (>=1), DMA starts immediately.	32'd0

54.4.2.16 FCHRDMAC

- Base Address: 0x1083_0000
- Address = Base Address + 0x0048, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:9]	–	Reserved	23'd0
ARCACHE	[8:5]	RW	ARCACHE value of AR channel of AXI	4'd0
ARPROT	[4:2]	RW	ARPROT value of AR channel of AXI	3'd0
BYTESWAP	[1]	RW	If this bit is set to high, then the data read from the bus is byte-swapped in a 64-bit boundary. If this bit is low (default), then the data is handed over to the FIFO without byte-swap. Normally this bit should be "0".	1'b0
FLUSH	[0]	RW	If this bit is set to high, then data flushes from FIFO and DMA. This bit is cleared automatically.	1'd0

54.4.3 AES Engine Hardware Register Base

54.4.3.1 AES_control

- Base Address: 0x1083_0200
- Address = Base Address + 0x0000, Reset Value = 32'h0f80

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	–	Reserved	18'b0
AES Counter Size	[13:12]	RW	AES counter size selection signal 00 = Selects 128-bit counter 01 = Selects 64-bit counter 10 = Selects 32-bit counter 11 = Selects 16-bit counter	2'b00
AES_ByteSwap_DI	[11]	RW	0 = Disables input data byte swap 1 = Enables input data byte swap	1'b1
AES_ByteSwap_DO	[10]	RW	0 = Disables output data byte swap 1 = Enables output data byte swap	1'b1
AES_ByteSwap_IV	[9]	RW	0 = Disables initial value byte swap 1 = Enables initial value byte swap	1'b1
AES_ByteSwap_CNT	[8]	RW	0 = Disables counter data byte swap 1 = Enables counter data byte swap	1'b1
AES_ByteSwap_Key	[7]	RW	0 = Disables key byte swap 1 = Enables key byte swap	1'b1
RSVD	[6]	–	Reserved	1'b0
AES Key Size	[5:4]	RW	AES key size selection signal 00 = Selects 128-bit key 01 = Selects 192-bit key 10 = Selects 256-bit key	2'b00
FIFO Mode	[3]	RW	ARM/FIFO mode selection signal 0 = Selects ARM mode (ARM Slave) 1 = Selects FIFO mode	1'b0
AES Chain Mode	[2:1]	RW	AES chain mode selection signal 00 = Selects ECB mode 01 = Selects CBC mode 10 = Selects CTR mode	2'b00
AES Mode	[0]	RW	Encryption/Decryption mode selection signal 0 = Selects Encryption mode 1 = Selects Decryption mode	1'b0

NOTE: AES_control[0]: In case of CTR mode, AES core should always work in encryption mode even in decryption. Therefore AES_control[0] should always be "0".

54.4.3.2 AES_status

- Base Address: 0x1083_0200
- Address = Base Address + 0x0004, Reset Value = 32'd2

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	29'b0
Busy	[2]	R	AES busy signal 0 = Idle 1 = Busy	1'b0
Input Ready	[1]	R	AES input ready signal 0 = AES input buffer is not empty 1 = AES input buffer is empty, and the host get permission to write the next block of data to write the next block of data	1'b1
Output Ready	[0]	RW	AES output ready signal 0 = AES output is not available 1 = AES output is available for host to retrieve	1'b0

NOTE: To clear Output Ready bit, write 0x1 at that bit, AES_status[0].

54.4.3.3 AES_indata_01

- Base Address: 0x1083_0200
- Address = Base Address + 0x0010, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_indata_01	[31:0]	RW	Input data[127:96]	32'd0

54.4.3.4 AES_indata_02

- Base Address: 0x1083_0200
- Address = Base Address + 0x0014, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_indata_02	[31:0]	RW	Input data[95:64]	32'd0

54.4.3.5 AES_indata_03

- Base Address: 0x1083_0200
- Address = Base Address + 0x0018, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_indata_03	[31:0]	RW	Input data[63:32]	32'd0

54.4.3.6 AES_indata_04

- Base Address: 0x1083_0200
- Address = Base Address + 0x001C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_data_04	[31:0]	RW	Input data[31:0]	32'd0

54.4.3.7 AES_outdata_01

- Base Address: 0x1083_0200
- Address = Base Address + 0x0020, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_outdata_01	[31:0]	R	Output data[127:96]	32'd0

54.4.3.8 AES_outdata_02

- Base Address: 0x1083_0200
- Address = Base Address + 0x0024, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_outdata_02	[31:0]	R	Output data[95:64]	32'd0

54.4.3.9 AES_outdata_03

- Base Address: 0x1083_0200
- Address = Base Address + 0x0028, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_outdata_03	[31:0]	R	Output data[63:32]	32'd0

54.4.3.10 AES_outdata_04

- Base Address: 0x1083_0200
- Address = Base Address + 0x002C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_outdata_04	[31:0]	R	Output data[31:0]	32'd0

54.4.3.11 AES_ivdata_01

- Base Address: 0x1083_0200
- Address = Base Address + 0x0030, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_ivdata_01	[31:0]	RW	Initialization vector[127:96]	32'd0

54.4.3.12 AES_ivdata_02

- Base Address: 0x1083_0200
- Address = Base Address + 0x0034
- Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_ivdata_02	[31:0]	RW	Initialization vector[95:64]	32'd0

54.4.3.13 AES_ivdata_03

- Base Address: 0x1083_0200
- Address = Base Address + 0x0038, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_ivdata_03	[31:0]	RW	Initialization vector[63:32]	32'd0

54.4.3.14 AES_ivdata_04

- Base Address: 0x1083_0200
- Address = Base Address + 0x003C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_ivdata_04	[31:0]	RW	Initialization vector[31:0]	32'd0

54.4.3.15 AES_cntdata_01

- Base Address: 0x1083_0200
- Address = Base Address + 0x0040, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_cntdata_01	[31:0]	RW	Counter data[127:96]	32'd0

54.4.3.16 AES_cntdata_02

- Base Address: 0x1083_0200
- Address = Base Address + 0x0044, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_cntdata_02	[31:0]	RW	Counter data[95:64]	32'd0

54.4.3.17 AES_cntdata_03

- Base Address: 0x1083_0200
- Address = Base Address + 0x0048, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_cntdata_03	[31:0]	RW	Counter data[63:32]	32'd0

54.4.3.18 AES_cntdata_04

- Base Address: 0x1083_0200
- Address = Base Address + 0x004C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_cntdata_04	[31:0]	RW	Counter data[31:0]	32'd0

54.4.3.19 AES_keydata_01

- Base Address: 0x1083_0200
- Address = Base Address + 0x0080, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_keydata_01	[31:0]	W	Input key data[255:224]	32'd0

54.4.3.20 AES_keydata_02

- Base Address: 0x1083_0200
- Address = Base Address + 0x0084, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_keydata_02	[31:0]	W	Input key data[223:192]	32'd0

54.4.3.21 AES_keydata_03

- Base Address: 0x1083_0200
- Address = Base Address + 0x0088, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_keydata_03	[31:0]	W	Input key data[191:160]	32'd0

54.4.3.22 AES_keydata_04

- Base Address: 0x1083_0200
- Address = Base Address + 0x008C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_keydata_04	[31:0]	W	Input key data[159:128]	32'd0

54.4.3.23 AES_keydata_05

- Base Address: 0x1083_0200
- Address = Base Address + 0x0090, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_keydata_05	[31:0]	W	Input key data[127:96]	32'd0

54.4.3.24 AES_keydata_06

- Base Address: 0x1083_0200
- Address = Base Address + 0x0094, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_keydata_06	[31:0]	W	Input key data[95:64]	32'd0

54.4.3.25 AES_keydata_07

- Base Address: 0x1083_0200
- Address = Base Address + 0x0098, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_keydata_07	[31:0]	W	Input key data[63:32]	32'd0

54.4.3.26 AES_keydata_08

- Base Address: 0x1083_0200
- Address = Base Address + 0x009C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
AES_keydata_08	[31:0]	W	Input key data[31:0]	32'd0

54.4.4 TDES Engine Hardware Register Base

54.4.4.1 TDES_CONF

- Base Address: 0x1083_0300
- Address = Base Address + 0x0000, Reset Value = 32'h3C0

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	22'b0
TDES_ByteSwap_DI	[9]	RW	Data input Byte Swap enable/disable signal 0 = Disables data input byte swap signal 1 = Enables data input byte swap signal	1'b1
TDES_ByteSwap_DO	[8]	RW	Data output Byte Swap enable/disable signal 0 = Disables data output byte swap signal 1 = Enables data output byte swap signal al	1'b1
TDES_ByteSwap_IV	[7]	RW	Initialization vector Byte Swap enable/disable signal 0 = Disables initialization vector byte swap signal 1 = Enables initialization vector byte swap signal	1'b1
TDES_ByteSwap_Key	[6]	RW	Key Byte Swap enable/disable signal 0 = Disables key byte swap signal 1 = Enables key byte swap signal	1'b1
TDES_DMA	[5]	RW	CPU/DMA mode selection signal 0 = Selects CPU mode 1 = Selects DMA mode	1'b0
TDES_EEE	[4]	RW	TDES mode selection signal 0 = Selects TDES EDE mode 1 = Selects TDES EEE mode	1'b0
TDES_Select	[3]	RW	DES/TDES algorithm selection signal 0 = Selects DES algorithm signal 1 = Selects TDES algorithm signal	1'b0
RSVD	[2]	–	Reserved	1'b0
TDES_Mode	[1]	RW	Chain mode selection signal 0 = Selects ECB mode 1 = Selects CBC mode	1'b0
TDES_Enc	[0]	RW	Encryption/Decryption mode selection signal 0 = Selects Encryption mode 1 = Selects Decryption mode	1'b0

54.4.4.2 TDES_STAT

- Base Address: 0x1083_0300
- Address = Base Address + 0x0004, Reset Value = 32'd2

Name	Bit	Type	Description	Reset Value
RSVD	[31:3]	–	Reserved	29'b0
TDES_Busy	[2]	R	TDES/DES busy signal 0 = Idle 1 = Busy	1'b0
TDES_Ready	[1]	R	TDES/DES input ready signal 0 = TDES/DES Input buffer is full 1 = TDES/DES Input buffer is empty, and the host get permission to write the next block of data to write the next block of data	1'b1
TDES_Valid	[0]	RW	TDES/DES output valid signal 0 = TDES/DES output is invalid 1 = TDES/DES output is valid	1'b0

NOTE: To clear TDES_Valid bit, 0x1 has to be written to TDES_STAT[0] bit .

54.4.4.3 TDES_KEY1_0

- Base Address: 0x1083_0300
- Address = Base Address + 0x0010, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
TDES_KEY1_0	[31:0]	W	Key 1[63:32]	32'd0

54.4.4.4 TDES_KEY1_1

- Base Address: 0x1083_0300
- Address = Base Address + 0x0014, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
TDES_KEY1_1	[31:0]	W	Key 1[31:0]	32'd0

54.4.4.5 TDES_KEY2_0

- Base Address: 0x1083_0300
- Address = Base Address + 0x0018, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
TDES_KEY2_0	[31:0]	W	Key 2[63:32]	32'd0

54.4.4.6 TDES_KEY2_1

- Base Address: 0x1083_0300
- Address = Base Address + 0x001C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
TDES_KEY2_1	[31:0]	W	Key 2[31:0]	32'd0

54.4.4.7 TDES_KEY3_0

- Base Address: 0x1083_0300
- Address = Base Address + 0x0020, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
TDES_KEY3_0	[31:0]	W	Key 3[63:32]	32'd0

54.4.4.8 TDES_KEY3_1

- Base Address: 0x1083_0300
- Address = Base Address + 0x0024, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
TDES_KEY3_1	[31:0]	W	Key 3[31:0]	32'd0

54.4.4.9 TDES_IV_0

- Base Address: 0x1083_0300
- Address = Base Address + 0x0028, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
TDES_IV_0	[31:0]	RW	Input initialization vector[63:32]	32'd0

54.4.4.10 TDES_IV_1

- Base Address: 0x1083_0300
- Address = Base Address + 0x002C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
TDES_IV_1	[31:0]	RW	Input initialization vector[31:0]	32'd0

54.4.4.11 TDES_INPUT_0

- Base Address: 0x1083_0300
- Address = Base Address + 0x0030, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
TDES_INPUT_0	[31:0]	RW	Input data[63:32]	32'd0

54.4.4.12 TDES_INPUT_1

- Base Address: 0x1083_0300
- Address = Base Address + 0x0034, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
TDES_INPUT_1	[31:0]	RW	Input data[31:0]	32'd0

54.4.4.13 TDES_OUTPUT_0

- Base Address: 0x1083_0300
- Address = Base Address + 0x0038, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
TDES_OUTPUT_0	[31:0]	R	Output data[63:32]	32'd0

54.4.4.14 TDES_OUTPUT_1

- Base Address: 0x1083_0300
- Address = Base Address + 0x003C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
TDES_OUTPUT_1	[31:0]	R	Output data[31:0]	32'd0

54.4.5 HASH Engine Hardware Register Base

54.4.5.1 HASH_CONTROL_1

- Base Address: 0x1083_0400
- Address = Base Address + 0x0000, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	24'd0
RSVD	[7]	–	Reserved	1'b0
RSVD	[6]	–	Reserved	1'b0
USER_IV_EN	[5]	RW	Use customized IV. Automatically cleared by hardware	1'b0
START_INIT_BIT	[4]	RW	Starts/initializes the hash/HMAC/PRNG Automatically cleared by hardware	1'b0
Engine_Selection	[3:0]	RW	4'b0000 = SHA1_HASH 4'b0001 = SHA1_HMAC 4'b0010 = MD5_HASH 4'b0011 = MD5_HMAC 4'b0100 = SHA256_HASH 4'b0101 = SHA256_HMAC 4'b1000 = PRNG	4'b0000

54.4.5.2 HASH_CONTROL_2

- Base Address: 0x1083_0400
- Address = Base Address + 0x0004, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	31'd0
HASH_PAUSE	[0]	RW	Pauses a Hash/HMAC operation so that the user can read the partial result. Automatically cleared by hardware The operations including partial results use this bit.	1'b0

54.4.5.3 HASH_FIFO_MODE_EN

- Base Address: 0x1083_0400
- Address = Base Address + 0x0008, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	31'd0
HASH_FIFO_ MODE_EN	[0]	RW	0 = Disables FIFO mode (default) 1 = Enables FIFO mode	1'b0

54.4.5.4 HASH_BYTE_SWAP

- Base Address: 0x1083_0400
- Address = Base Address + 0x000C, Reset Value = 32'h0000_000F

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	28'd0
HASH_SWAP_DI	[3]	RW	Byte swap of data input 0 = Does not swaps data input 1 = Swaps data input (default)	1'b1
HASH_SWAP_DO	[2]	RW	Byte swap of data output (hash result) 0 = Does not swaps data output 1 = Swaps data output (default)	1'b1
HASH_SWAP_IV	[1]	RW	Byte swap of custom IVs 0 = Does not swaps custom IVs 1 = Swaps custom IVs (default)	1'b1
HASH_SWAP_KEY	[0]	RW	Byte swap of HMAC key input 0 = Does not swaps HMAC key input 1 = Swaps HMAC key input (default)	1'b1

54.4.5.5 HASH_STATUS

- Base Address: 0x1083_0400
- Address = Base Address + 0x0010, Reset Value = 32'h0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	24'd0
PRNG_ERROR	[7]	R	PRNG error bit This bit goes to HIGH if a PRNG request occurs without a complete seed setup. To clear this bit, you should perform a complete seed setup operation.	1'b0
MSG_DONE	[6]	RW	Hash/HMAC done signal You should write 1 into this bit to clear it.	1'b0
PRNG_DONE	[5]	RW	PRNG done signal You should write 1 into this bit to clear it.	1'b0
PARTIAL_DONE	[4]	RW	The partial result done signal You should write 1 into this bit to clear it.	1'b0
RSVD	[3]	–	Reserved	1'b0
PRNG_BUSY	[2]	R	PRNG engine status 1 = Busy 0 = Idle	1'b0
SEED_SETTING_DONE	[1]	R	1 = Seed setup is done. 0 = Seed setup is not done.	1'b0
BUFFER_READY	[0]	R	Specifies the status of internal SHA1 buffer 0 = The buffer is full. 1 = The buffer still has empty spaces (not full).	1'b1

NOTE: Status bits[4], [5], [6] and [7] are also used to generate the interrupt signal.

54.4.5.6 HASH_MSG_SIZE_LOW

- Base Address: 0x1083_0400
- Address = Base Address + 0x0020, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
MSG_SIZE_LOW	[31:8]	RW	Message size in bytes (lower 32 bits)	32'd0

54.4.5.7 HASH_MSG_SIZE_HIGH

- Base Address: 0x1083_0400
- Address = Base Address + 0x0024, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
MSG_SIZE_HIGH	[31:0]	RW	Message size in bytes (higher 32 bits)	32'd0

54.4.5.8 HASH_PRE_MSG_LENG_LOW

- Base Address: 0x1083_0400
- Address = Base Address + 0x0028, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_PRE_MSG_LENG_LOW	[31:0]	RW	Pre-message length[31:0]	32'd0

54.4.5.9 HASH_PRE_MSG_LENG_HIGH

- Base Address: 0x1083_0400
- Address = Base Address + 0x002C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_PRE_MSG_LENG_HIGH	[31:0]	RW	Pre-message length[63:32]	32'd0

54.4.5.10 HASH_DATA_IN_1 to HASH_DATA_IN_16

- Base Address: 0x1083_0400
- Address = Base Address + 0x0030 to 0x006C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_DATA_IN_1 to HASH_DATA_IN_16	[31:0]	W	Message input register 1 to 16 Only effective when the FIFO mode is disabled Supports burst up to 16 words.	32'd0

54.4.5.11 HASH_HMAC_KEY_IN_1 to HASH_HMAC_KEY_IN_16

- Base Address: 0x1083_0400
- Address = Base Address + 0x0070 to 0x00AC, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_HMAC_KEY_ IN_1 to HASH_HMAC_KEY_ IN_16	[31:0]	W	HMAC key input register 1 to 16 Each address corresponds to each word of the internal 512-bit key storage. HASH_HMAC_KEY_IN_1: Key[511:480] HASH_HMAC_KEY_IN_2: Key[479:448] HASH_HMAC_KEY_IN_3: Key[447:416] … HASH_HMAC_KEY_IN_16: Key[31:0]	32'd0

54.4.5.12 HASH_USER_IV_IN_1

- Base Address: 0x1083_0400
- Address = Base Address + 0x00B0, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_USER_IV_IN_1	[31:0]	RW	Custom IV input 1	32'd0

NOTE:

1. MD5: HASH_USER_IV_IN_1 to HASH_USER_IV_IN_4
2. SHA1: HASH_USER_IV_IN_1 to HASH_USER_IV_IN_5
3. SHA256: HASH_USER_IV_IN_1 to HASH_USER_IV_IN_8

54.4.5.13 HASH_USER_IV_IN_2

- Base Address: 0x1083_0400
- Address = Base Address + 0x00B4, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_USER_IV_IN_2	[31:0]	RW	Custom IV input 2	32'd0

54.4.5.14 HASH_USER_IV_IN_3

- Base Address: 0x1083_0400
- Address = Base Address + 0x00B8, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_USER_IV_IN_3	[31:0]	RW	Custom IV input 3	32'd0

54.4.5.15 HASH_USER_IV_IN_4

- Base Address: 0x1083_0400
- Address = Base Address + 0x00BC, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_USER_IV_IN_4	[31:0]	RW	Custom IV input 4	32'd0

54.4.5.16 HASH_USER_IV_IN_5

- Base Address: 0x1083_0400
- Address = Base Address + 0x00C0, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_USER_IV_IN_5	[31:0]	RW	Custom IV input 5	32'd0

54.4.5.17 HASH_USER_IV_IN_6

- Base Address: 0x1083_0400
- Address = Base Address + 0x00C4, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_USER_IV_IN_6	[31:0]	RW	Custom IV input 6	32'd0

54.4.5.18 HASH_USER_IV_IN_7

- Base Address: 0x1083_0400
- Address = Base Address + 0x00C8, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_USER_IV_IN_7	[31:0]	RW	Custom IV input 7	32'd0

54.4.5.19 HASH_USER_IV_IN_8

- Base Address: 0x1083_0400
- Address = Base Address + 0x00CC, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_USER_IV_IN_8	[31:0]	RW	Custom IV input 8	32'd0

54.4.5.20 HASH_RESULT_1

- Base Address: 0x1083_0400
- Address = Base Address + 0x0100, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_RESULT_1	[31:0]	R	Hash/HMAC/Partial result 1	32'd0

NOTE:

1. MD5: HASH_RESULT_1 to HASH_RESULT_4
2. SHA1: HASH_RESULT_1 to HASH_RESULT_5
3. SHA256: HASH_RESULT_1 to HASH_RESULT_8

54.4.5.21 HASH_RESULT_2

- Base Address: 0x1083_0400
- Address = Base Address + 0x0104, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_RESULT_2	[31:0]	R	Hash/HMAC/Partial result 2	32'd0

54.4.5.22 HASH_RESULT_3

- Base Address: 0x1083_0400
- Address = Base Address + 0x0108, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_RESULT_3	[31:0]	R	Hash/HMAC/Partial result 3	32'd0

54.4.5.23 HASH_RESULT_4

- Base Address: 0x1083_0400
- Address = Base Address + 0x010C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_RESULT_4	[31:0]	R	Hash/HMAC/Partial result 4	32'd0

54.4.5.24 HASH_RESULT_5

- Base Address: 0x1083_0400
- Address = Base Address + 0x0110, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_RESULT_5	[31:0]	R	Hash/HMAC/Partial result 5	32'd0

54.4.5.25 HASH_RESULT_6

- Base Address: 0x1083_0400
- Address = Base Address + 0x0114, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_RESULT_6	[31:0]	R	Hash/HMAC/Partial result 6	32'd0

54.4.5.26 HASH_RESULT_7

- Base Address: 0x1083_0400
- Address = Base Address + 0x0118, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_RESULT_7	[31:0]	R	Hash/HMAC/Partial result 7	32'd0

54.4.5.27 HASH_RESULT_8

- Base Address: 0x1083_0400
- Address = Base Address + 0x011C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_RESULT_8	[31:0]	R	Hash/HMAC/Partial result 8	32'd0

54.4.5.28 HASH_SEED_IN_1

- Base Address: 0x1083_0400
- Address = Base Address + 0x0140, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_SEED_IN_1	[31:0]	W	PRNG seed buffer[159:128]	32'd0

54.4.5.29 HASH_SEED_IN_2

- Base Address: 0x1083_0400
- Address = Base Address + 0x0144, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_SEED_IN_2	[31:0]	W	PRNG seed buffer[127:96]	32'd0

54.4.5.30 HASH_SEED_IN_3

- Base Address: 0x1083_0400
- Address = Base Address + 0x0148, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_SEED_IN_3	[31:0]	W	PRNG seed buffer[95:64]	32'd0

54.4.5.31 HASH_SEED_IN_4

- Base Address: 0x1083_0400
- Address = Base Address + 0x014C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_SEED_IN_4	[31:0]	W	PRNG seed buffer[63:32]	32'd0

54.4.5.32 HASH_SEED_IN_5

- Base Address: 0x1083_0400
- Address = Base Address + 0x0150, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_SEED_IN_5	[31:0]	W	PRNG seed buffer[31:0]	32'd0

54.4.5.33 HASH_PRNG_1

- Base Address: 0x1083_0400
- Address = Base Address + 0x0160, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_PRNG_1	[31:0]	R	PRNG output 1	32'd0

54.4.5.34 HASH_PRNG_2

- Base Address: 0x1083_0400
- Address = Base Address + 0x0164, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_PRNG_2	[31:0]	R	PRNG output 2	32'd0

54.4.5.35 HASH_PRNG_3

- Base Address: 0x1083_0400
- Address = Base Address + 0x0168, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_PRNG_3	[31:0]	R	PRNG output 3	32'd0

54.4.5.36 HASH_PRNG_4

- Base Address: 0x1083_0400
- Address = Base Address + 0x016C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_PRNG_4	[31:0]	R	PRNG output 4	32'd0

54.4.5.37 HASH_PRNG_5

- Base Address: 0x1083_0400
- Address = Base Address + 0x0170, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
HASH_PRNG_5	[31:0]	R	PRNG output 5	32'd0

54.4.5.38 Notes

- About reserved bits

Accessing reserved bits results in UNDEFINED behaviors.

- HASH_CONTROL_1

You can assert ENGINE_SELECTION, START_INIT_BIT, and USER_IV_EN simultaneously. If custom IVs are used, initialize HASH_IV_1 to HASH_IV_8 before asserting USER_IV_EN.

- HASH_MSG_SIZE_LOW and HASH_MSG_SIZE_HIGH

Figure 54-40 illustrates the HASH_MSG_SIZE_LOW and HASH_MSG_SIZE_HIGH.

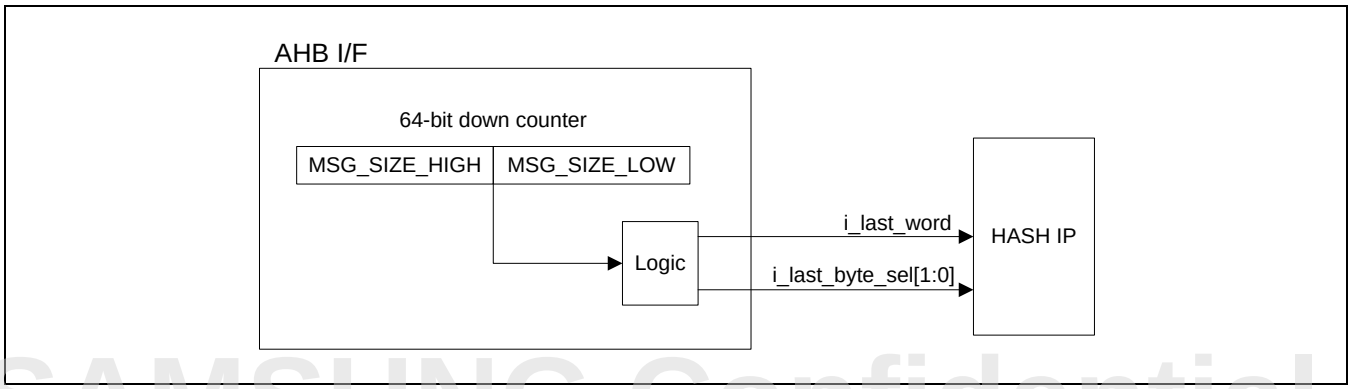


Figure 54-40 HASH_MSG_SIZE_LOW and HASH_MSG_SIZE_HIGH

As illustrated in *Figure 54-40*, MSG_SIZE_LOW and MSG_SIZE_HIGH forms a 64-bit counter. The counter is initialized by the user through SFR write transfers. If data words are written through SFR or FIFO, the counter decreases itself. When the counter is about to become zero, the internal logic generates correct "i_last_word" and "i_last_byte_sel" signals for IP.

Note that unit of this counter is byte. The maximum counting range is (264-1) bytes, which is larger than the range specified in SHA1 specification.

In certain cases, you may use HASH_MSG_SIZE_HIGH and HASH_MSG_SIZE_LOW in a different way. A typical example is multi-part hashing (partial result is involved) without knowing the total message size in advance. In this case, you can initialize the counter with a "big" number (such as 64'h80000000_00000000) for all parts (except the last one). If you are going to process last part, by which the message is known, you should initialize this counter with real message size.

- HASH_BYTE_SWAP

If HASH_SWAP_DI, HASH_SWAP_IV or HASH_SWAP_KEY is set to 0, data will enter hash core in the same order as HWDATA[31:0]. Otherwise, it byte-swaps the 32-bit word before entering hash core. Note that the hash core is designed with "big endian" in mind, so you should turn on byte swapping if the bus is little endian.

For HASH_SWAP_DO, see the example:

SHA1 ("abcd") = 81fe8bfe_87576c3e_cb22426f_8e578473_82917acf
SFR_READ (HASH_RESULT_1) → HRDATA[31:0] = 0x81fe8bfe (when HASH_SWAP_DO = 0)
SFR_READ (HASH_RESULT_1) → HRDATA[3:0] = 0xfe8bfe81 (when HASH_SWAP_DO = 1)

To avoid byte order issues, you should configure HASH_BYTE_SWAP properly.

- HASH_USER_IV_IN_1 to HASH_USER_IV_IN_8

The values in these registers are sampled and saved by the hardware only when both USER_IV_EN (HASH_CONTROL_1[5]) and START_INIT_BIT (HASH_CONTROL_1[4]) are set to high. Since USER_IV_EN and START_INIT_BIT are automatically cleared by hardware, these registers do not need to be cleared after they are used.

- HASH_PRE_MSG_LENG_LOW and HASH_PRE_MSG_LENG_HIGH

In contrast to HASH_USER_IV_IN_x, these two registers always affect hardware. Therefore, they must be set to zero when "Pre-message length" is not used. Note that, the unit of the value is bit.

54.4.6 PKA Engine Hardware Register Base

54.4.6.1 PKA_SFR0

- Base Address: 0x1083_0700
- Address = Base Address + 0x0000, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	RW	Reserved	–
CHNK_SZ	[6:3]	RW	Sets the size of the chunk 0000 = (don't use)[Default] 0001 = (don't use) 0010 = (don't use) 0011 = 128 bits 0100 = 160 bits 0101 = 192 bits 0110 = 224 bits 0111 = 256 bits 1000 = 288 bits 1001 = 320 bits 1010 = 352 bits 1011 = 384 bits 1100 = 416 bits 1101 = 448 bits 1110 = 480 bits 1111 = 512 bits	4'b0
PREC_ID	[1:0]	RW	Sets the precision 00 = Single precision, i.e., x1[Default] 01 = Double precision, i.e., x2 10 = Triple precision, i.e., x3 11 = Quadruple precision, i.e., x4	2'b00
RSVD	[2]	RW	Reserved	–

NOTE: Operand's bit length = (Chunk's size) × (Precision)
For example: (160 bits) × Single = 160 bits, (512 bits) × Double=1024 bits, (512 bits) × Quadruple = 2048 bits

54.4.6.2 PKA_SFR1

- Base Address: 0x1083_0700
- Address = Base Address + 0x0004, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
PLDM_ON (NOTE)	[3]	RW	Controls pre-loading of least significant chunk of modulus M 0 = Does not pre-load least significant chunk of modulus M[Default] 1 = Pre-loads least significant chunk of modulus M	1'b0
RSVD	[2:1]	RW	Reserved	–
EXEC_ON	[0]	RW	Controls and monitor execution of PKA 0 = PKA stays in idle state[Default] 1 = PKA starts to run and is being in running state	1'b0

NOTE: If PLDM_ON is set to "1", PKA loads least significant chunk of modulus M data from memory to internal register of PKA at initial time of multiplication. For the whole modular exponentiation, only first modular multiplication needs to pre-load the least significant chunk of modulus M. When PKA performs a number of modular multiplications except first one, the least significant chunk of modulus M is pre-loaded during previous modular multiplication. When PLDM_ON is "0", the processing time for multiplication becomes shorter. The saved clock cycles by setting PLDM_ON to "0" is c/32 + 5.

54.4.6.3 PKA_SFR2

- Base Address: 0x1083_0700
- Address = Base Address + 0x0008, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
A_SEG_ID	[28:24]	RW	Memory Segment ID for input operand A in PKA -2 mode 00000 = Segment 0[Default] 00001 = Segment 1 00010 = Segment 2 00011 = Segment 3 00100 = Segment 4 00101 = Segment 5 00110 = Segment 6 00111 = Segment 7 01000 = Segment 8 01001 = Segment 9 01010 = Segment 10 01011 = Segment 11 01100 = Segment 12 01101 = Segment 13 01110 = Segment 14 01111 = Segment 15 10000 = Segment 16 10001 = Segment 17 10010 = Segment 18 10011 = Segment 19 10100 = Segment 20 10101 = Segment 21 10110 = Segment 22 10111 = Segment 23 11000 = Segment 24 11001 = Segment 25 11010 = Segment 26 11011 = Segment 27 11100 = Segment 28 11101 = Segment 29 11110 = (dedicated to the hardware's internal usage) 11111 = (dedicated to the hardware's internal usage)	5'b00000
RSVD	[23:21]	RW	Reserved	–
B_ SEG_ID	[20:16]	RW	Memory Segment ID for input operand B in PKA-2 mode 00000 = Segment 0[Default] 00001 = Segment 1 00010 = Segment 2 00011 = Segment 3 00100 = Segment 4 00101 = Segment 5 00110 = Segment 6 00111 = Segment 7	5'b00000

Name	Bit	Type	Description	Reset Value
			01000 = Segment 8 01001 = Segment 9 01010 = Segment 10 01011 = Segment 11 01100 = Segment 12 01101 = Segment 13 01110 = Segment 14 01111 = Segment 15 10000 = Segment 16 10001 = Segment 17 10010 = Segment 18 10011 = Segment 19 10100 = Segment 20 10101 = Segment 21 10110 = Segment 22 10111 = Segment 23 11000 = Segment 24 11001 = Segment 25 11010 = Segment 26 11011 = Segment 27 11100 = Segment 28 11101 = Segment 29 11110 = (dedicated to hardware's internal usage) 11111 = (dedicated to hardware's internal usage)	
RSVD	[15:13]	RW	Reserved	–
M_ SEG_ID	[12:8]	RW	Memory Segment ID for input operand M in PKA-2 mode 00000 = Segment 0[Default] 00001 = Segment 1 00010 = Segment 2 00011 = Segment 3 00100 = Segment 4 00101 = Segment 5 00110 = Segment 6 00111 = Segment 7 01000 = Segment 8 01001 = Segment 9 01010 = Segment 10 01011 = Segment 11 01100 = Segment 12 01101 = Segment 13 01110 = Segment 14 01111 = Segment 15 10000 = Segment 16 10001 = Segment 17 10010 = Segment 18 10011 = Segment 19 10100 = Segment 20 10101 = Segment 21 10110 = Segment 22	5'b00000

Name	Bit	Type	Description	Reset Value
			10111 = Segment 23 11000 = Segment 24 11001 = Segment 25 11010 = Segment 26 11011 = Segment 27 11100 = Segment 28 11101 = Segment 29 11110 = (dedicated to hardware's internal usage) 11111 = (dedicated to hardware's internal usage)	
RSVD	[7:5]	RW	Reserved	–
S_ SEG_ID	[4:0]	RW	Memory Segment ID for input operand S in PKA-2 mode 00000 = Segment 0[Default] 00001 = Segment 1 00010 = Segment 2 00011 = Segment 3 00100 = Segment 4 00101 = Segment 5 00110 = Segment 6 00111 = Segment 7 01000 = Segment 8 01001 = Segment 9 01010 = Segment 10 01011 = Segment 11 01100 = Segment 12 01101 = Segment 13 01110 = Segment 14 01111 = Segment 15 10000 = Segment 16 10001 = Segment 17 10010 = Segment 18 10011 = Segment 19 10100 = Segment 20 10101 = Segment 21 10110 = Segment 22 10111 = Segment 23 11000 = Segment 24 11001 = Segment 25 11010 = Segment 26 11011 = Segment 27 11100 = Segment 28 11101 = Segment 29 11110 = (dedicated to hardware's internal usage) 11111 = (dedicated to hardware's internal usage)	5'b00000

54.4.6.4 PKA_SFR3

- Base Address: 0x1083_0700
- Address = Base Address + 0x000C, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:30]	RW	Reserved	–
SEG_SIGN	[29:0]	RW	Signs of the numbers stored in the segments 0 at ith bit: The number in ith segment is positive [Default] 1 at ith bit: The number in ith segment is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx0: Segment 0 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx1: Segment 0 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xx0x: Segment 1 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xx1x: Segment 1 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_x0xx: Segment 2 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_x1xx: Segment 2 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_0xxx: Segment 3 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_1xxx: Segment 3 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx0_xxxx: Segment 4 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx1_xxxx: Segment 4 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xx0x_xxxx: Segment 5 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xx1x_xxxx: Segment 5 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_x0xx_xxxx: Segment 6 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_x1xx_xxxx: Segment 6 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_0xxx_xxxx: Segment 7 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_1xxx_xxxx: Segment 7 is negative xx_xxxx_xxxx_xxxx_xxxx_xxx0_xxxx_xxxx: Segment 8 is positive xx_xxxx_xxxx_xxxx_xxxx_xxx1_xxxx_xxxx: Segment 8 is negative	–

Name	Bit	Type	Description	Reset Value
			xx_xxxx_xxxx_xxxx_xxxx_xx0x_xxxx_xxxx: Segment 9 is positive xx_xxxx_xxxx_xxxx_xxxx_xx1x_xxxx_xxxx: Segment 9 is negative xx_xxxx_xxxx_xxxx_xxxx_x0xx_xxxx_xxxx: Segment 10 is positive xx_xxxx_xxxx_xxxx_xxxx_x1xx_xxxx_xxxx: Segment 10 is negative xx_xxxx_xxxx_xxxx_xxxx_0xxx_xxxx_xxxx: Segment 11 is positive xx_xxxx_xxxx_xxxx_xxxx_1xxx_xxxx_xxxx: Segment 11 is negative xx_xxxx_xxxx_xxxx_xxx0_xxxx_xxxx_xxxx: Segment 12 is positive xx_xxxx_xxxx_xxxx_xxx1_xxxx_xxxx_xxxx: Segment 12 is negative xx_xxxx_xxxx_xxxx_xx0x_xxxx_xxxx_xxxx: Segment 13 is positive xx_xxxx_xxxx_xxxx_xx1x_xxxx_xxxx_xxxx: Segment 13 is negative xx_xxxx_xxxx_xxxx_x0xx_xxxx_xxxx_xxxx: Segment 14 is positive xx_xxxx_xxxx_xxxx_x1xx_xxxx_xxxx_xxxx: Segment 14 is negative xx_xxxx_xxxx_xxxx_0xxx_xxxx_xxxx_xxxx: Segment 15 is positive xx_xxxx_xxxx_xxxx_1xxx_xxxx_xxxx_xxxx: Segment 15 is negative xx_xxxx_xxxx_xxx0_xxxx_xxxx_xxxx_xxxx: Segment 16 is positive xx_xxxx_xxxx_xxx1_xxxx_xxxx_xxxx_xxxx: Segment 16 is negative xx_xxxx_xxxx_xx0x_xxxx_xxxx_xxxx_xxxx: Segment 17 is positive xx_xxxx_xxxx_xx1x_xxxx_xxxx_xxxx_xxxx: Segment 17 is negative xx_xxxx_xxxx_x0xx_xxxx_xxxx_xxxx_xxxx: Segment 18 is positive xx_xxxx_xxxx_x1xx_xxxx_xxxx_xxxx_xxxx: Segment 18 is negative xx_xxxx_xxxx_0xxx_xxxx_xxxx_xxxx_xxxx: Segment 19 is positive xx_xxxx_xxxx_1xxx_xxxx_xxxx_xxxx_xxxx: Segment 19 is negative xx_xxxx_xxx0_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 20 is positive xx_xxxx_xxx1_xxxx_xxxx_xxxx_xxxx_xxxx:	

Name	Bit	Type	Description	Reset Value
			Segment 20 is negative xx_xxxx_xx0x_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 21 is positive xx_xxxx_xx1x_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 21 is negative xx_xxxx_x0xx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 22 is positive xx_xxxx_x1xx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 22 is negative xx_xxxx_0xxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 23 is positive xx_xxxx_1xxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 23 is negative xx_xxx0_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 24 is positive xx_xxx1_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 24 is negative xx_xx0x_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 25 is positive xx_xx1x_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 25 is negative xx_x0xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 26 is positive xx_x1xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 26 is negative xx_0xxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 27 is positive xx_1xxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 27 is negative x0_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 28 is positive x1_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 28 is negative 0x_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 29 is positive 1x_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 29 is negative	

54.4.6.5 PKA_SFR4

- Base Address: 0x1083_0700
- Address = Base Address + 0x0010, Reset Value = 32'd0

Name	Bit	Type	Description	Reset Value
RSVD	[31:7]	RW	Reserved	–
SEG_SIZE	[6:5]	RW	Size of memory segments 00 = Full-size (that is , 256 bytes) [Default] 01 = Half-size (that is , 128 bytes) 10 = Quarter-size (that is , 64 bytes)	2'b00
SVD	[4:1]	RW	Reserved	–
FUNC_ID	[0]	RW	Select the function to be executed 0 = Montgomery multiplication (A by B) [Default] 1 = Montgomery multiplication (A by 1)	1'b0

NOTE: Selecting half-size and quarter-size segments is only possible at PKA-2 mode.

54.4.7 PKA Engine Local Memory

54.4.7.1 PKA_SRAM

- Base Address: 0x1083_0800
- Address = Base Address + 0x0010 to 0x0800, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
PKA_SRAM	–	RW	PKA Memory	–

54.5 Example Codes

This section includes:

- SFR definition
- Feeder
- AES
- DES/3DES
- HASH/PRNG
- PKA
- Interrupt Service Routine
- Useful Routines

54.5.1 SFR Definition

The following code is an example of SFR definition.

```
#define       FEED_REG_BASE          0x????????
#define       FCINTSTAT              (*(volatile unsigned int*)(FEED_REG_BASE + 0x0000)
#define       FCINTENSET             (*(volatile unsigned int*)(FEED_REG_BASE + 0x0004)
...
```


54.5.2 Feeder

The following code is an example of Feeder.

```
The following code is an example of AES FIFO mode.
void AES_ENC_FIFO (unsigned int key[4], unsigned int* SrcAddr, unsigned int* DstAddr, unsigned int
Length) {

      // DMA Initialization
      FCBRDMAC = 0x1;               // Flushing
      FCBTDMAC = 0x1;               // Flushing

      // FIFO Interconnection Configuration
      FCFIFOCTRL = 0x0;             // AES Selection

      // AES Setting
      AES_keydata_05 = key[0];
      AES_keydata_06 = key[1];
      AES_keydata_07 = key[2];
      AES_keydata_08 = key[3];
      AES_control = 0x88;           // ECB mode, FIFO mode, 128 bit key, little endian
                                    // Setting it to FIFO mode starts processing automatically

      // DMA Setting
```

```
        FCBRDMAS = SrcAddr;
        FCBRDMAL = Length;
        FCBTDMAS = DstAddr;
        FCBTDMAL = Length;

        // Polling the end of processing
        while(!(FCINTPEND & 0x4));    // Waits for the end of BTDMA transfers
}
```

The following code is an example of DES decoding before Hashing.

```
void DES_DEC_before_HASH (unsigned int key[2], unsigned int* SrcAddr, unsigned int* DstAddr, unsigned
int Length)
{

	// DMA Initialization
	FCBRDMAC = 0x1;			// Flushing
	FCBTDMAC = 0x1;			// Flushing

	// FIFO Interconnection Configuration
	FCFIFOCTRL = 0x6;		// DES Selection and Hashing the output of Block Cipher

	// DES Setting
	TDES_KEY1_0 = key[0];
	TDES_KEY1_1 = key[1];
	TDES_CONF = 0x9;		// ECB mode, FIFO mode, decryption
				// Setting it to FIFO mode starts processing automatically
	// Hash Setting
	HASH_MSG_SIZE_LOW = Length;
	HASH_ENGINE = 0x10;		// Start bit = 1, engine = 0000
	HASH_FIFO_MODE_EN = 0x1;	// FIFO mode is enabled

	// DMA Setting
	FCBTDMAS = DstAddr;
	FCBTDMAL = Length;
	FCBRDMAS = SrcAddr;
	FCBRDMAL = Length;

	// Polling the end of processing
	while (!(FCINTPEND & 0x4));	// Waits for the end of BTDMA transfers

	// Polling the end of hash operation
	while (! (HASH_STATUS & 0x40));
}
```

54.5.3 AES

This section includes:

- AES CPU mode using one buffer
- AES CPU mode using two buffer
- AES DMA mode

54.5.3.1 AES CPU Mode Using One Buffer

The following code is an example of AES CPU mode using one buffer.

```
//-------------------------------------------------------------------------
// Secure Block Initialization
//-------------------------------------------------------------------------
SBLK_Init(AES, eAesOperMode, ENC)
/* The following is an example implementation of the above function.
      Copy ((u32)uAesKey, AES_keydata_05, 4);                    // install aes key
      Copy ((u32)uAesIV, AES_ivdata_01, 4);                      // install aes iv
      Copy ((u32)uAesInitCounter, AES_cntdata_01, 4);            // install ctr data

      uReg =  ((eDirSel == ENC)? 0 : 1) |                        //AES Enc/Dec
              ((oSblk.m_eOperMode == ECB)? (0 << 1):
              ((oSblk.m_eOperMode == CBC)? (1 << 1):(2 << 1)))|  // AES Chain Mode
              ((eMode == FIFO)? (1 << 3) : (0 << 3))|            // FIFO Mode
              (0 << 4)|                                          // AES Key Size (128bit)
              (0 << 12);                                         // AES Counter Size (128bit)

      Outp32(AES_control, uReg);
*/

for (i = 0; i < 4; i++)
{

//-------------------------------------------------------------------------
// Secure Block Input Write
//-------------------------------------------------------------------------
      SBLK_PutDataToInReg ((u32) uAesPlainText + i*4*4, 4, LASTBYTE_1ST)
      /* The following is an example implementation of the above function.
      pDstAddr = (u32 *)AES_indata_01;
      for (i = uSize; i > 0; i--)
            Outp32 (pDstAddr++, *pSrcAddr++);
      */

//-------------------------------------------------------------------------
// Secure Block Output Ready Polling
//-------------------------------------------------------------------------
      while(!SBLK_IsOutputReady());
```

```
//-------------------------------------------------------------------------------
// Secure Block Output Read
//-------------------------------------------------------------------------------
      SBLK_GetDataFromOutReg (OUTPUT_BUF + i*4*4, 4)
      /* The following is an example implementation of the above function.
      pSrc = (u32 *)AES_outdata_01;
      for (i = 0; i < uSize; i++) {
            *pDst++ = *pSrc++;
}
      */

//-------------------------------------------------------------------------------
// Secure Block Valid signal zeroization
//-------------------------------------------------------------------------------
      SBLK_ValidZero()
      /* The following is an example implementation of the above function.
            Outp32 (AES_status, 0x1);
      */
}
```

54.5.3.2 AES CPU Mode Using Two Buffer

The following code is an example of AES CPU mode using two buffers.

```
//-------------------------------------------------------------------------------
// Secure Block Initialization
//-------------------------------------------------------------------------------
SBLK_Init (AES, eAesOperMode, ENC)
/* The following is an example implementation of the above function.

      Copy ((u32) uAesKey, AES_keydata_05, 4);                    // install aes key
      Copy ((u32) uAesIV, AES_ivdata_01, 4);                      // install aes iv
      Copy ((u32) uAesInitCounter, AES_cntdata_01, 4);            // install ctr data

      uReg = ((eDirSel == ENC)? 0 : 1)|                           //AES Enc/Dec
             ((oSblk.m_eOperMode == ECB)? (0 << 1):
             ((oSblk.m_eOperMode == CBC)? (1 << 1):(2 << 1)))|    // AES Chain Mode
             ((eMode == FIFO)? (1 << 3):(0 << 3))|                // FIFO Mode
             (0 << 4)|                                            // AES Key Size (128 bit)
             (0 << 12);                                           // AES Counter Size (128 bit)

      Outp32(AES_control, uReg);
*/

i = 0;
//-------------------------------------------------------------------------------
// Secure Block Input Write
//-------------------------------------------------------------------------------
SBLK_PutDataToInReg ((u32) uAesPlainText + i*4*4, 4, LASTBYTE_1ST)
/* The following is an example implementation of the above function.
      pDstAddr = (u32 *)AES_indata_01;
      for (i = uSize; i > 0; i--)
             Outp32 (pDstAddr++, *pSrcAddr++);
*/

for (i = 1; i < 4; i++)
{
//-------------------------------------------------------------------------------
// Secure Block Input Write
//-------------------------------------------------------------------------------
      SBLK_PutDataToInReg ((u32) uAesPlainText + i*4*4, 4, LASTBYTE_1ST)
      /* The following is an example implementation of the above function.
      pDstAddr = (u32 *) AES_indata_01;
      for (i = uSize; i > 0; i--)
             Outp32 (pDstAddr++, *pSrcAddr++);
*/

//-------------------------------------------------------------------------------
// Secure Block Output Ready Polling
```

```
//-------------------------------------------------------------------------------
      while (!SBLK_IsOutputReady());

//-------------------------------------------------------------------------------
// Secure Block Output Read
//-------------------------------------------------------------------------------
      SBLK_GetDataFromOutReg (OUTPUT_BUF+ (i - 1) *4*4, 4)
      /* The following is an example implementation of the above function.
      pSrc = (u32 *)AES_outdata_01;
      for (i = 0; i < uSize; i++) {
             *pDst++ = *pSrc++;
}
      */

//-------------------------------------------------------------------------------
// Secure Block Valid signal zeroization
//-------------------------------------------------------------------------------
      SBLK_ValidZero()
      /* The following is an example implementation of the above function.
             Outp32 (AES_status, 0x1);
      */
}

//-------------------------------------------------------------------------------
// Secure Block Output Ready Polling
//-------------------------------------------------------------------------------
while (!SBLK_IsOutputReady());

//-------------------------------------------------------------------------------
// Secure Block Output Read
//-------------------------------------------------------------------------------
SBLK_GetDataFromOutReg (OUTPUT_BUF+ (i - 1)*4*4, 4);

//-------------------------------------------------------------------------------
// Secure Block Valid signal zeroization
//-------------------------------------------------------------------------------
SBLK_ValidZero ();
```

54.5.3.3 AES DMA Mode

The following code is an example of AES DMA mode.

```
//-----------------------------------------------------------------------------
// Secure Block DMA Flush
//-----------------------------------------------------------------------------
SBLK_DMA_Flush()
/* The following is an example implementation of the above function.
      Outp32 (FCBRDMAC, 0x1);                                   // Block Cipher Rx DMA Flush
      Outp32 (FCBTDMAC, 0x1);                                   // Block Cipher Tx DMA Flush
      Outp32 (FCHRDMAC, 0x1);                                   // Hash Rx DMA Flush
*/

//-----------------------------------------------------------------------------
// Secure Block Initialization
//-----------------------------------------------------------------------------
SBLK_Init (AES, eAesOperMode, ENC, FIFO)
/* The following is an example implementation of the above function.
      Copy ((u32) uAesKey, AES_keydata_05, 4                    // install aes key
      Copy ((u32) uAesIV, AES_ivdata_01, 4);                    // install aes iv
C     opy ((u32) uAesInitCounter, AES_cntdata_01, 4);           // install ctr data

      uReg = ((eDirSel == ENC)? 0:1)|                           //AES Enc/Dec
             ((oSblk.m_eOperMode == ECB)? (0 << 1):
             ((oSblk.m_eOperMode == CBC)? (1 << 1):(2 << 1)))|  // AES Chain Mode
             ((eMode == FIFO)? (1 << 3):(0 << 3))|              // FIFO Mode
             (0 << 4)|                                          // AES Key Size (128 bit)
             (0 << 12);                                         // AES Counter Size (128 bit)
      Outp32 (AES_control, uReg);
*/

//-----------------------------------------------------------------------------
// Secure Block DMA Setting
//-----------------------------------------------------------------------------
SBLK_DMA_Set ((u32) uAesPlainText, OUTPUT_BUF, 64)
/* The following is an example implementation of the above function.

      Outp32 (FCFIFOCTRL, 0x0);                                 // AES select

      Outp32 (FCBTDMAS, uDstAddr);                              // Block Cipher Tx Start Addr
      Outp32 (FCBTDMAL, uSize);                                 // Block Cipher Tx Length

      Outp32 (FCBRDMAS, uSrcAddr);                              // Block Cipher Rx Start Addr
      Outp32 (FCBRDMAL, uSize);                                 // Block Cipher Rx Length
*/

//-----------------------------------------------------------------------------
// Secure Block DMA Done
```

```
//-----------------------------------------------------------------------------
```

SBLK_IsDMADone ()

```
/* The following is an example implementation of the above function.
      u32 uRead;

      do {
              Inp32 (FCINTPEND, uRead);
} while (! ((uRead >> 2) & 0x1));                                  // BC Tx Pending Interrupt Sig
      Outp32 (FCINTPEND, uRead);
*/
```

54.5.4 DES/3DES

This section includes:

- DES/3DES CPU mode using one buffer
- DES/3DES CPU mode using two buffer
- DES/3DES DMA mode

54.5.4.1 DES/3DES CPU Mode Using One Buffer

The following code is an example of DES/3DES CPU mode using one buffer.

```
//------------------------------------------------------------------------------
// Secure Block Initialization
//------------------------------------------------------------------------------
SBLK_Init(DES, ECB, ENC, CPU)
/* The following is an example implementation of the above function.

      Copy ((u32) uDesKey, DES_ADDR_KEY1_0, 6);                  // install des key
      Copy ((u32) uDesIV, DES_ADDR_IV_0, 2);                     // install des iv

      uReg = ((eDirSel == ENC)? 0:1)|                            // Enc/Dec
             ((oSblk.m_eOperMode == ECB)? (0 << 1):(1 << 1))|    // Mode of operation
             (0 << 3)|                                           // TDES select
             (0 << 4)|                                           // TDES eee select
             ((eMode == DMA)? (1 << 5) : (0 << 5));              // DMA select

      Outp32 (DES_ADDR_CONF, uReg);                              // install des conf
*/

for (i = 0; i < 8; i++)
{

//------------------------------------------------------------------------------
// Secure Block Input Write
//------------------------------------------------------------------------------
      SBLK_PutDataToInReg (SrcAddr + i*4*2, 2)
      /* The following is an example implementation of the above function.

      pDstAddr = (u32*)DES_ADDR_IDAT_0;
      for (i = uSize; i > 0; i--)
             Outp32 (pDstAddr++, *pSrcAddr++);
      */

//------------------------------------------------------------------------------
// Secure Block Output Ready Polling
//------------------------------------------------------------------------------
      while (!SBLK_IsOutputReady());
```

```
//-------------------------------------------------------------------------------
// Secure Block Output Read
//-------------------------------------------------------------------------------
```

SBLK_GetDataFromOutReg (DstAddr + I × 4 × 2, 2)

```
      /* The following is an example implementation of the above function.
      pSrc = (u32 *) DES_ADDR_ODAT_0;
      for (i = 0; i < uSize; i++) {
             *pDst++ = *pSrc++;
      }
      */

//-------------------------------------------------------------------------------
// Secure Block Valid signal zeroization
//-------------------------------------------------------------------------------
```

SBLK_ValidZero ()

```
      /* The following is an example implementation of the above function.
      Outp32 (DES_ADDR_STAT, 0x1);
      */
}
```

54.5.4.2 DES/3DES CPU Mode Using Two Buffer

The following code is an example of DES/3DES CPU mode using two buffers.

```
//--------------------------------------------------------------------------
// Secure Block Initialization
//--------------------------------------------------------------------------
SBLK_Init (DES, ECB, ENC, CPU)
/* The following is an example implementation of the above function.

      Copy ((u32) uDesKey, DES_ADDR_KEY1_0, 6);                  // install des key
      Copy ((u32) uDesIV, DES_ADDR_IV_0, 2);                     // install des iv

      uReg = ((eDirSel == ENC)? 0:1)|                            // Enc/Dec select
             ((oSblk.m_eOperMode == ECB)? (0 << 1):(1 << 1))|    // Mode of operation
             (0 << 3)|                                           // TDES select
             (0 << 4)|                                           // TDES eee select
             ((eMode == DMA)? (1 << 5):(0 << 5));                // DMA select

      Outp32 (DES_ADDR_CONF, uReg);                              // install des conf
*/

i = 0;

//--------------------------------------------------------------------------
// Secure Block Input Write
//--------------------------------------------------------------------------
SBLK_PutDataToInReg (SrcAddr + i*4*2, 2)
/* The following is an example implementation of the above function.
      pDstAddr = (u32*)DES_ADDR_IDAT_0;
      for (i = uSize; i > 0; i--)
             Outp32(pDstAddr++, *pSrcAddr++);
*/

for (i = 1; i < 8; i++)
{
//--------------------------------------------------------------------------
// Secure Block Input Write
//--------------------------------------------------------------------------
      SBLK_PutDataToInReg(SrcAddr + i*4*2, 2)
      /* The following is an example implementation of the above function.
             pDstAddr = (u32 *)DES_ADDR_IDAT_0;
             for (i = uSize; i > 0; i--)
                    Outp32 (pDstAddr++, *pSrcAddr++);
      */

//--------------------------------------------------------------------------
// Secure Block Output Ready Polling
//--------------------------------------------------------------------------
```

```
while (!SBLK_IsOutputReady ());

//------------------------------------------------------------------------------
// Secure Block Output Read
//------------------------------------------------------------------------------
      SBLK_GetDataFromOutReg (DstAddr+ (i - 1) *4*2, 2)
      /* The following is an example implementation of the above function.
      pSrc = (u32*) DES_ADDR_ODAT_0;
      for (i = 0; i < uSize; i++) {
            *pDst++ = *pSrc++;
}
      */

//------------------------------------------------------------------------------
// Secure Block Valid signal zeroization
//------------------------------------------------------------------------------
      SBLK_ValidZero()
      /* The following is an example implementation of the above function.
            Outp32 (DES_ADDR_STAT, 0x1);
      */
}

//------------------------------------------------------------------------------
// Secure Block Output Ready Polling
//------------------------------------------------------------------------------
while (!SBLK_IsOutputReady ());

//------------------------------------------------------------------------------
// Secure Block Output Read
//------------------------------------------------------------------------------
SBLK_GetDataFromOutReg (DstAddr+ (i - 1) *4*2, 2);

//------------------------------------------------------------------------------
// Secure Block Valid signal zeroization
//------------------------------------------------------------------------------
SBLK_ValidZero ();
```

54.5.4.3 DES/3DES DMA Mode

The following code is an example of DES/3DES DMA mode.

```
//-----------------------------------------------------------------------------
// Secure Block DMA Flush
//-----------------------------------------------------------------------------
SBLK_DMA_Flush ()
/* The following is an example implementation of the above function.
      Outp32 (FCBRDMAC, 0x1);                                   // Block Cipher Rx DMA Flush
      Outp32 (FCBTDMAC, 0x1);                                   // Block Cipher Tx DMA Flush
      Outp32 (FCHRDMAC, 0x1);                                   // Hash Rx DMA Flush
*/

//-----------------------------------------------------------------------------
// Secure Block Initialization
//-----------------------------------------------------------------------------
SBLK_Init (DES, ECB, ENC, DMA)
/* The following is an example implementation of the above function.

      Copy ((u32) uDesKey, DES_ADDR_KEY1_0, 6);                 // install des key
      Copy ((u32) uDesIV, DES_ADDR_IV_0, 2);                    // install des iv

      uReg = ((eDirSel == ENC)? 0:1)|                           // Enc/Dec select
             ((oSblk.m_eOperMode == ECB)? (0 << 1):(1 << 1))|   // Mode of operation
             (0 << 3)|                                          // TDES select
             (0 << 4)|                                          // TDES eee select
             ((eMode == DMA)? (1 << 5):(0 << 5));               // DMA select

      Outp32 (DES_ADDR_CONF, uReg);                             // install des conf
*/

//-----------------------------------------------------------------------------
// Secure Block DMA Setting
//-----------------------------------------------------------------------------
SBLK_DMA_Set (SrcAddr, DstAddr, 16)
/* The following is an example implementation of the above function.
      Outp32 (FCFIFOCTRL, 0x4);                                 // DES select
      Outp32 (FCBTDMAS, uDstAddr);                              // Block Cipher Tx Start Addr
      Outp32 (FCBTDMAL, uSize);                                 // Block Cipher Tx Length

      Outp32 (FCBRDMAS, uSrcAddr);                              // Block Cipher Rx Start Addr
      Outp32 (FCBRDMAL, uSize);                                 // Block Cipher Rx Length
*/

//-----------------------------------------------------------------------------
// Secure Block DMA Done
//-----------------------------------------------------------------------------
SBLK_IsDMADone ()
```

```
/* The following is an example implementation of the above function.
      u32 uRead;
      do{
              Inp32 (FCINTPEND, uRead);
      } while (! ((uRead >> 2) & 0x1));                          // BC Tx Pending Interrupt Sig
      Outp32 (FCINTPEND, uRead);
*/
```

54.5.5 HASH/PRNG

The following code is an example of HASH/PRNG.

```
//--------------------------------------------------------------------------
// A utility function
//--------------------------------------------------------------------------
void calc_blocks_words (u32 msg_size, u32 * x, u32 * y)
{
       u32 total_words;

       total_words = (msg_size + 3)/4;

       *x = total_words/16;
       *y = total_words % 16;

}
```

54.5.5.1 Hash (CPU)

The following code is an example of Hash (CPU) mode.

```
//--------------------------------------------------------------------
// SHA1 hash, MD5 hash, SHA256 hash
// CPU mode
// msg_size: in bytes
// msg_addr is a word pointer.
//--------------------------------------------------------------------
void basic_hash_cpu (ALG_TYPE alg, u32 * msg_addr, u32 msg_size, u32* OUPUT_BUF)
{
	u32 num_blocks;
	u32 remain_words;
	u32 i;

	assert (msg_size! = 0);

	// Calculate the number of blocks and the number of remaining words
	calc_blocks_words (msg_size, &num_blocks, &remain_words);

	// Disable the FIFO mode
	HASH_FIFO_MODE_EN = 0x0;

	// Set the message size in bytes
	HASH_MSG_SIZE_LOW = msg_size;
	HASH_MSG_SIZE_HIGH = 0x0;

	// Set "pre_msg_leng" to zero
	HASH_PRE_MSG_LENG_LOW = 0x0;
	HASH_PRE_MSG_LENG_HIGH = 0x0;


	// Set the engine and the start/init bit together
	if (alg == ALG_SHA1)
	{
	HASH_CONTROL_1 = 0x10;
	}
	else if (alg == ALG_MD5)
	{
	HASH_CONTROL_1 = 0x12;
	}
	else if (alg == ALG_SHA256)
	{
	HASH_CONTROL_1 = 0x14;
	}

	// Data input
	for (i = 0; i < num_blocks; i++)
```

```
        {
                // Poll "buf_ready"
                while (!(HASH_STATUS & 0x1));

                // Data input: 16 words
                Copy ((u32)(msg_addr + i*16), HASH_DATA_IN_1, 16);
        }

        if (remain_words! = 0)
        {
                // Poll "buf_ready"
                while (!(HASH_STATUS & 0x1));

                // Data input: 1~16 words
                Copy ((u32)(msg_addr + num_blocks*16), HASH_DATA_IN_1, remain_words);
        }

        // Poll the "done" signal
        while (!(HASH_STATUS & 0x40));

        // Write "1" to clear it
        HASH_STATUS = 0x40;

        // Read the result
        if (alg == ALG_SHA1)
        {
                Copy (HASH_RESULT_1, OUTPUT_BUF, 5);
        }
        else if (alg == ALG_MD5)
        {
                Copy (HASH_RESULT_1, OUTPUT_BUF, 4);
        }
        else if (alg == ALG_SHA256)
        {
        Copy (HASH_RESULT_1, OUTPUT_BUF, 8);
        }
}
```

54.5.5.2 Hash (DMA)

The following is an example of Hash (DMA) mode.

```
//-------------------------------------------------------------
// SHA1 hash, MD5 hash, SHA256 hash
// DMA mode
//-------------------------------------------------------------
void basic_hash_dma (ALG_TYPE alg, u32 * msg_addr, u32 msg_size, u32* OUTPUT_BUF)
{
	assert(msg_size != 0);

	// Flush HRDMA to be safe
	FCHRDMAC = 0x1;

	// HASHINSEL = 2'b00 (independent source)
	FCFIFOCTRL = FCFIFOCTRL & 0x4;
	// HASHINSEL: keep bit[2] and set bit[1:0] to 2'b00

	// Enable the FIFO mode
	HASH_FIFO_MODE_EN = 0x1;

	// Set the message size in bytes
	HASH_MSG_SIZE_LOW = msg_size;
	HASH_MSG_SIZE_HIGH = 0x0;

	// Set "pre_msg_leng" to zero
	HASH_PRE_MSG_LENG_LOW = 0x0;
	HASH_PRE_MSG_LENG_HIGH = 0x0;

	// Set the engine and the start/init bit together
	if (alg == ALG_SHA1)
	{
		HASH_CONTROL_1 = 0x10;
	}
	else if (alg == ALG_MD5)
	{
		HASH_CONTROL_1 = 0x12;
	}
	else if (alg == ALG_SHA256)
	{
		HASH_CONTROL_1 = 0x14;
	}

	// Congifure HRDMA
	FCHRDMAS = (u32) msg_addr;
	FCHRDMAL = msg_size;		//--> DMA starts.

	// Poll the done signal
```

```
        while (!(HASH_STATUS & 0x40));

        // Write "1" to clear it
        HASH_STATUS = 0x40;

        // Read the result
        if (alg == ALG_SHA1)
        {
                Copy (HASH_RESULT_1, OUTPUT_BUF, 5);
        }
        else if (alg == ALG_MD5)
        {
                Copy (HASH_RESULT_1, OUTPUT_BUF, 4);
        }
        else if (alg == ALG_SHA256)
        {
                Copy(HASH_RESULT_1, OUTPUT_BUF, 8);
        }

        // Manually clear HRDMA pending bit, which is FCINTPEND[1]
        FCINTPEND = 0x2;
}
```

54.5.5.3 HMAC (CPU)

The following code is an example of HMAC (CPU).

```
//------------------------------------------------------------------------------
// Setup the HMAC key
// You may skip this step if you want to reuse the previous key.
//------------------------------------------------------------------------------
void set_hmac_key (u32 * key_data)
{
      Copy ((u32) key_data, HASH_HMAC_KEY_IN_1, 16);
}


//------------------------------------------------------------------------------
//SHA1-HMAC, MD5-HMAC, SHA256-HMAC
//CPU mode
//------------------------------------------------------------------------------
void basic_hmac_cpu (ALG_TYPE alg, u32 *msg_addr, u32 msg_size, u32 * OUTPUT_BUF)
{
      u32 num_blocks;
      u32 remain_words;
      u32 i;

      if (it is necessary)
      {
          set_hmac_key();
      }

      // calculate the number of blocks and the number of remaining words
      // Note that the key is not included.
      calc_blocks_words(msg_size, &num_blocks, &remain_words);

      // Disable the FIFO mode
      HASH_FIFO_MODE_EN = 0x0;

      // Set the message size in bytes
      // The key is NOT included!
      HASH_MSG_SIZE_LOW = msg_size;
      HASH_MSG_SIZE_HIGH = 0x0;

      // Set "pre_msg_leng" to zero
      HASH_PRE_MSG_LENG_LOW = 0x0;
      HASH_PRE_MSG_LENG_HIGH = 0x0;

      // Set the engine and the start/init bit together
      // Hashing of the key will start automatically.
      if (alg == ALG_SHA1)
      {
```

```
        HASH_CONTROL_1 = 0x11;
}
else if (alg == ALG_MD5)
{
        HASH_CONTROL_1 = 0x13;
}
else if (alg == ALG_SHA256)
{
        HASH_CONTROL_1 = 0x15;
}

// Data input
for (i = 0; i < num_blocks; i++)
{
        // Poll "buf_ready"
        while (!(HASH_STATUS & 0x1));

        // Data input: 16 words
        Copy ((u32)(msg_addr + i*16), HASH_DATA_IN_1, 16);
}

if (remain_words! = 0)
{
        // Poll "buf_ready"
        while (!(HASH_STATUS & 0x1));

        // Data input: 1~16 words
        Copy ((u32)(msg_addr + num_blocks*16), HASH_DATA_IN_1, remain_words);
}

// Poll the "done" signal
while (!(HASH_STATUS & 0x40));

// Write "1" to clear it
HASH_STATUS = 0x40;

// Read the final result
if (alg == ALG_SHA1)
{
        Copy (HASH_RESULT_1, OUTPUT_BUF, 5);
}
else if (alg == ALG_MD5)
{
        Copy (HASH_RESULT_1, OUTPUT_BUF, 4);
}
else if (alg == ALG_SHA256)
{
        Copy(HASH_RESULT_1, OUTPUT_BUF, 8);
```

```
        }
}
```

54.5.5.4 HMAC (DMA)

The following code is an example of HMAC (DMA).

```
//-------------------------------------------------------------
// SHA1-HMAC, MD5-HMAC, SHA256-HMAC
// DMA mode
//-------------------------------------------------------------
void basic_hmac_dma (ALG_TYPE alg, u32 *msg_addr, u32 msg_size, u32 * OUTPUT_BUF)
{
	if (it is necessary)
	{
		set_hmac_key();
	}

	//-----------------------------------------------
	// DMA initialization
	//-----------------------------------------------

	// Flush HRDMA to be safe
	Outp32 (FCHRDMAC, 0x1);

	// HASHINSEL = 2'b00 (independent source)
	FCFIFOCTRL = FCFIFOCTRL & 0x4;
	// HASHINSEL: keep bit[2] and set bit[1:0] to 2'b00

	// Enable the FIFO mode
	HASH_FIFO_MODE_EN = 0x1;

	// Set the message size in bytes
	// The size of the key is NOT included!
	HASH_MSG_SIZE_LOW = msg_size;
	HASH_MSG_SIZE_HIGH = 0x0;

	// Set "pre_msg_leng" to zero
	HASH_PRE_MSG_LENG_LOW = 0x0;
	HASH_PRE_MSG_LENG_HIGH = 0x0;

	// Set the engine and the start/init bit together
	// Hashing of the key will start automatically.
	if (alg == ALG_SHA1)
	{
		HASH_CONTROL_1 = 0x11;
	}
	else if (alg == ALG_MD5)
	{
		HASH_CONTROL_1 = 0x13;
	}
	else if (alg == ALG_SHA256)
```

```
        {
                HASH_CONTROL_1 = 0x15;
        }

        // Setup DMA for the message: (msg_addr, msg_size)
        FCHRDMAS = (u32) msg_addr;
        FCHRDMAL = msg_size;                                    // --> DMA starts.

        // Poll the "done" signal
        while (!(HASH_STATUS & 0x40));

        // Write "1" to clear it
        HASH_STATUS = 0x40;

        // Read the final result
        if (alg == ALG_SHA1)
        {
                Copy(HASH_RESULT_1, OUTPUT_BUF, 5);
        }
        else if (alg == ALG_MD5)
        {
                Copy (HASH_RESULT_1, OUTPUT_BUF, 4);
        }
        else if (alg == ALG_SHA256)
        {
                Copy (HASH_RESULT_1, OUTPUT_BUF, 8);
        }

        // Clear HRDMA pending bit
        FCINTPEND = 0x2;
}
```

54.5.5.5 Multi-part Hash

The following code is an example of Multi-part Hash.

```
//-------------------------------------------------------------------------------
// multi-part SHA1 hashing (MD5 and SHA256 are similar)
// msg_size: size of the current part, not the total message size
// Each part (except the last one) must be a multiple of 64 bytes.
// flag = FIRST, MIDDLE, LAST
//-------------------------------------------------------------------------------void
partial_hash_sha1 (u32 *msg_addr, u32 msg_size, u32 iv[5], u32 pre_msg_leng, u32 result_buf[5], u32
flag)
{
      u32 i;
      u32 num_blocks;
      u32 remain_words;

      calc_blocks_words (msg_size, &num_blocks, &remain_words);

      // write IV values
      if (flag == MIDDLE||flag == LAST)
      {
            Copy (iv, HASH_IV_1, 5);
      }
      // Disable the FIFO mode
      HASH_FIFO_MODE_EN = 0x0;

      // Set the message size
      if (flag == FIRST||flag == MIDDLE)
      {
      HASH_MSG_SIZE_LOW = 0x0;
      HASH_MSG_SIZE_HIGH = 0x80000000;                    //big value
      }
      else
      {
      HASH_MSG_SIZE_LOW = msg_size;                       // real size
      HASH_MSG_SIZE_HIGH = 0x0;
      }

      // set "pre_msg_leng"
      if (flag == LAST)
      {
      HASH_PRE_MSG_LENG_HIGH = 0;
      HASH_PRE_MSG_LENG_LOW = pre_msg_leng;
      }
      else
      {
      HASH_PRE_MSG_LENG_HIGH = 0;
```

```
    HASH_PRE_MSG_LENG_LOW = 0;
    }

    // Set the engine, the start/init bit and user_iv bit
    if (flag == FIRST)     HASH_CONTROL_1 = 0x10;        // engine = 0000, start = 1, user_iv = 0
    else                   HASH_CONTROL_1 = 0x30;        // engine = 0000, start = 1, user_iv = 1

    // Data input:

    // Data input for the whole blocks
    for (i = 0; i < num_blocks; i++)
    {
            // Poll "buf_ready"
            while (!(HASH_STATUS & 0x1));

            // Data input: 16 words
            Copy ((u32)(msg_addr + i*16), HASH_DATA_IN_1, 16);
    }

if (remain_words! = 0)
{
            // Poll "buf_ready"
            while (!(HASH_STATUS & 0x1));

            // Data input: 1~16 words
            Copy ((u32)(msg_addr + num_blocks*16), HASH_DATA_IN_1, remain_words);
}


    // Assert the "pause" signal
    if (flag == FIRST||flag == MIDDLE)
    {
    HASH_CONTROL_2 = 0x1;
    }

    // Poll the "partial_done" signal or the "msg_done" signal
    If (flag == FIRST||flag == MIDDLE)
    {
            while (!(HASH_STATUS & 0x10));
            HASH_STATUS = 0x10;
    }
    else
    {
    while (!(HASH_STATUS & 0x40));
    HASH_STATUS = 0x40;
    }

    // save the partial/final result
```

```
        Copy (HASH_RESULT_1, (u32) result_buf, 5);
}
```

54.5.5.6 Multi-part HMAC

For simplification it omits the code. Refer to programmer's model and previous section for more information.

54.5.5.7 PRNG

The following code is an example of PRNG.

```
void seed_setup (u32 seed[5])
{
	// Write seed data (5 words)
	Copy ((u32)seed, HASH_SEED_IN_1, 5);

	// Confirm that HASH_STATUS[1] == 1
	if (!(HASH_STATUS & 0x2))
	{
		// Error handling
	}
}

void generate_random_number (u32 prng_result[5])
{
	// set the engine and the start bit together
	HASH_CONTROL_1 = 0x18;

	// Poll the done signal
	while (!(HASH_STATUS & 0x20));

	HASH_STATUS = 0x20;

	// Read the PRNG result
	Copy (HASH_PRNG_1, (u32) prng_result, 5);
}
```

54.5.5.8 Concurrent Encoding/Decoding and Hashing

The following code is an example of concurrent DES decoding and hashing.

```
void DES_DEC_and_HASH (unsigned int key[2], unsigned int* SrcAddr, unsigned int* DstAddr, unsigned int
Length)
{

	// DMA Initialization
	FCBRDMAC = 0x1;                         // Flushing ON
	FCBTDMAC = 0x1;                         // Flushing ON

	// FIFO Interconnection Configuration
	FCFIFOCTRL = 0x5;                       // DES Selection and Hashing the input of Block Cipher

	// DES Setting
	DES_ADDR_KEY1_0 = key[0];
	DES_ADDR_KEY1_1 = key[1];
	DES_ADDR_CONF = 0x9;                    // ECB mode, FIFO mode, decryption
	                                        // Setting it to FIFO mode starts processing automatically
	// Hash Setting
	HASH_ADDR_FIFO = 0x1;                   // FIFO mode is enabled
	HASH_ADDR_MSG_SIZE_LOW = Length;
	HASH_ADDR_MSG_SIZE_HIGH = 0x0;
	HASH_ADDR_PRE_MSG_LENG_LOW = 0x0;
	HASH_ADDR_PRE_MSG_LENG_HIGH = 0x0;
	HASH_ADDR_CONTROL_1 = 0x10;             // Start bit = 1, engine = 0000 (SHA1_HASH)

	// DMA Setting
	FCBTDMAS = DstAddr;
	FCBTDMAL = Length;
	FCBRDMAS = SrcAddr;
	FCBRDMAL = Length;

	// Polling the end of processing
	while (!(FCINTPEND & 0x4));             // Waits for the end of BTDMA transfers

	// Polling the end of hashing
	while (!(HASH_STATUS & 0x40)) ;

	HASH_STATUS = 0x40;

	// clear BRDMA and BTDMA pending bits
	FCINTPEND = 0xC;                        // 1100
}
```

54.5.5.9 Encoding/Decoding First, Then Hashing

Refer to Section 4.2 for more information.

54.5.6 PKA

This test code is an example code about function test for PKA AxB operation. The file name is PKA2_FuncTest_AxB.s.

The flow of test code is like with Ch.3.5 PKA.

First, you have to set the registers to operate PKA.

```
;-------------------------------------------------------------------------------
; Setting PKA2 control registers
;-------------------------------------------------------------------------------
      LDR           r9,     = PKA2_SFR_Setting_Value      ; pointer for register setting
      LDR           r8,     = PKA2_EXPECTED_RESULT
; pointer for compare the result with expected value
      LDR           r7,     = PKA2_DATA_IN                ; pointer for input value
LOAD_PKA2_SFR_Setting
      LDR           r6,     = PKA_SFR0                    ; PKA2 cntrol register. #0 addr
      LDR           r0,     [r9],         #4
      AND           r1,     r0,           #0xFF
      CMP           r1,                   #0xFF
      BEQ           PASS                                  ; check for the end of data.

;-------------------------------------------------------------------------------
; 1. Register Setting & Input Data Writing
; <Step 1>. Register Setting
1.  PKA_SFR0: CHNK_SZ, PREC_ID value setting
;             - single precision:    PREC_ID = 1
;             - double precision:    PREC_ID = 2
;             - triple precision:    PREC_ID = 3
;             - quadruple precision: PREC_ID = 4
;-------------------------------------------------------------------------------
      MOV           r0,     r0,     LSR     #8
                                                  ; setting PKA2_ctl[0]:CHNK_SZ[6:3], PREC_ID[1:0]
      STR           r0,     [r6],   #4            ; r6 will be PKA2_CNTR1
      AND           r10,    r0,           #0x03   ; r10 save PREC_ID[1:0]
      ADD           r10,    r10,          #1
      MOV           r11,    r0,     LSR     #3
      AND           r11,    r11,          #0x0F   ; r11 save CHNK_SZ[6:3]
      MOV           r1,                   #32
      MUL           r2,     r11,    r1
      ADD           r11,    r2,           #32     ; r11 = chunk_size

2.  PKA_SFR1: PLDM_ON value setting
```

```
      MOV           r0,     r0,     LSR     #8
; setting PKA2_ctrl[1]:PLDM_ON[3], EXEC_ON[0]
      STR           r0,     [r6],           #4      ; r6 will be PKA2_CNTR2

3. PKA_SFR2: A_SEG_ID, B_SEG_ID, M_SEG_ID, S_SEG_ID value setting
      LDR           r0,     [r9],           #4
; loading the register setting value of PKA2_ctrl[2]
      STR           r0,     [r6],           #4      ; setting PKA2_ctrl[2]:A,B,M,S_SEG_ID,
      MOV           r1,     r0                      ; r1 contains the information about address.

4.  PKA_SFR3: SEG_SIGN value setting
      LDR           r0,     [r9],           #4
; loading the register setting value of PKA_2_ctrl[3]
      STR           r0,     [r6],           #4      ; setting PKA2_ctrl[3]:SEG_SIGN[28:0]

5. PKA_SFR4: SEG_SIZE and FUNC_ID value setting
      LDR           r0,     [r9],           #4
; setting PKA2_ctrl[4] : SEG_SIZE[6:5], FUNC_ID[0]
      STR           r0,     [r6]
      MOV           r2,                     #0
      MOV           r3,     r0,     LSR     #5
      AND           r3,     r3,             #0x03   ; r3 save SEG_SIZE[6:3]

;-------------------------------------------------------------------------------
; <Step 2>. Selecting Segment Size : Select the segment size
;      - 0x0 : Full-Size (256byte:Default)
;      - 0x1 : Half-Size (128byte)
;      - 0x2 : Quater-Size (64byte)
;-------------------------------------------------------------------------------
      CMP           r3,                     #2
      BEQ           segment_64                      ; segment size = 64-bit
      CMP           r3,                     #1
      BEQ           segment_128                     ; segment size = 128-bit

segment_256
      ADD           r2,     r2,             #128

segment_128
      ADD           r2,     r2,             #64

segment_64
      ADD           r2,     r2,             #64     ; r2 = contains segment size

      MUL           r4,     r10,    r11
; r10 = PREC_ID, r11 = CHNK_SZ => r4 = key_size
      MOV           r11,    r4,     LSR     #5      ; r11 = key_size/32
      MOV           r12,    r4,     LSR     #3      ; r12 = key_size/8
```

```
      SUB           r12,    r12,          #4
; r12 = (key_size/8) - 4 = end address of segment_X


;-------------------------------------------------------------------------------
; <Step 3>. Load Input Data
; Send operand form MEMORY to Segment 0, 1, 2...
;-------------------------------------------------------------------------------
1.  Load Input A into Memory.
load_a
      MOV           r3,     r1,     LSR     #24
; r1 contains the information about segment address.
      AND           r3,     r3,           #0x1F   ; r3 = A_SEG_ID[28:24]
      MUL           r4,     r3,     r2            ; r4 = A_SEG_ID * SEG_SZ
      LDR           r5,     = pSsS_PKRAM_BASE
      ADD           r5,     r5,     r4
      ADD           r5,     r5,     r12
; r5 contains the information about A_SEG's end address.
      MOV           r6,     r11
; Set auto-load loop value (8-word=256bit at a time)
      BL            auto_load                   ; Loop Load data operation from r0 Num of r1

2.  Load Input B into Memory
load_b
      MOV           r3,     r1,     LSR     #16
                                                ; r1 contains the information about segment
                                                address.
      AND           r3,     r3,           #0x1F   ; r3 = B_SEG_ID[20:16]
      MUL           r4,     r3,     r2            ; r4 = SEG_ID * SEG_SZ
      LDR           r5,     = pSsS_PKRAM_BASE
      ADD           r5,     r5,     r4
      ADD           r5,     r5,     r12           ; r5 = end addr of B_SEG
      MOV           r6,     r11                   ; Set auto-load loop value (8 - word = 256 bit at a
                                                time)
      BL            auto_load                   ; Loop Load data operation from r0 Num of r1

3.  Load Input M into Memory
load_m
      MOV           r3,     r1,     LSR     #8
                                                ; r1 contains the information about segment
                                                address.
      AND           r3,     r3,           #0x1F   ; r3 = M_SEG_ID[12:8]
      MUL           r4,     r3,     r2            ; SEG_ID * SEG_SZ
      LDR           r5,     =pSsS_PKRAM_BASE
      ADD           r5,     r5,     r4
      ADD           r5,     r5,     r12           ; r5 = end addr of M_SEG
      MOV           r6,     r11
                                                ; Set auto-load loop value (8-word=256bit at a
                                                time)
```

```
      BL             auto_load                    ; Loop Load data operation from r0 Num of r1
      B              start_PKA2

auto_load                                         ;load & store
      _SEG_LOAD      r7,     r5,     r6
      MOV            pc,     r14


;-------------------------------------------------------------------------------
; 2. RUN : Start PKA2 Test by Setting PKA_SFR1[0] (EXEC_ON)
;-------------------------------------------------------------------------------
start_PKA2
      LDR            r5,     = PKA_SFR1
      LDR            r6,     [r5]
      ORR            r6,     r6,            #0x01   ; EXEC_ON[0] = 1 <input run signal>
      STR            r6,     [r5]

      NOP
      NOP
      NOP

RSA_POLLING
      LDR            r5,     = PKA_SFR1
      LDR            r6,     [r5]
      TST            r6,                    #0x00000001
      BNE            RSA_POLLING

      NOP
      NOP


;-------------------------------------------------------------------------------
; 3. After RUN
; <Step 1>. Result S's Sign Check after PKA2 Test
; after_PKA2
;-------------------------------------------------------------------------------
      LDR            r5,     = PKA_SFR3
      LDR            r6,     [r5]
      AND            r3,     r1,            #0x1F   ; r3 = S_SEG_ID[4:0]
      MOV            r6,     r6,     LSR r3
      ANDS           r6,     r6,            #0x01
      BEQ            chk_PKA2                     ; sign of previous S result is positive


;-------------------------------------------------------------------------------
; <Step 2>. Change Negative S to Posivive S
; change negative output to positive value
;-------------------------------------------------------------------------------
```

```
s1_add_m
      MUL           r4,     r3,     r2              ; r4 = SEG_ID * SEG_SZ
      LDR           r3,     = pSsS_PKRAM_BASE
      ADD           r3,     r3,     r4
      MOV           r4,     r1,     LSR     #8
      AND           r4,     r4,             #0x1F   ; r4 = M_SEG_ID[4:0]
      MUL           r5,     r4,     r2              ; SEG_ID * SEG_SZ
      LDR           r4,     = pSsS_PKRAM_BASE
      ADD           r4,     r4,     r5

      MOV           r0,     r11                     ; Set loop value, r11 = key_size/32
      ADDS          r5,     r5,             #0

add_reload
      LDR           r5,     [r3]                    ; load a word from S seg. to r4
      LDR           r6,     [r4],           #4
                                                    ; load a word from M seg. to r5, and then inc
                                                    $base #4
      ADCS          r5,      r5,    r6
      STR           r5,     [r3],           #4
                                                    ; store a word r4 to S seg, and then inc $dest #4
      SUB           r0,      r0,            #1
      TEQ           r0,                     #0
      BNE           add_reload

;-------------------------------------------------------------------------------
; <Step 3>. Compare Result S with Expected Result S
; Save PKA2 output value & check them
;------------------------------------------------------------------------------- chk_PKA2
      AND           r3,     r1,             #0x1F   ; r3 = S_SEG_ID[12:8]
      MUL           r4,     r3,     r2              ; r4 = SEG_ID * SEG_SZ
      LDR           r5,     = pSsS_PKRAM_BASE
      ADD           r5,     r5,     r4
      ADD           r5,     r5,     r12
                                                    ; r12 = (key_size/8) - 4 =  end addr of segment_X
      MOV           r6,     r11                     ; reload Loop_count value

NEXT_CHECK
      LDR           r3,     [r5],           # - 4
                                                    ; load a word from S seg to r3, and then dec r5 #4
      LDR           r4,     [r8],           #4
                                                    ; load a word from PKA2_EXPECTED_RESULT to r4,
                                                    and then inc r8 #4
      CMP           r3,     r4
      BNE           OUTPUT_ERROR
      SUB           r6,     r6,             #1
      CMP           r6,                     #0
```

```
      BNE             NEXT_CHECK
      B               OUTPUT_OK

OUTPUT_OK
      B                LOAD_PKA2_SFR_Setting


;-------------------------------------------------------------
; Error Routine
;-------------------------------------------------------------
OUTPUT_ERROR
      LDR             R4,      = FAIL_FLAG
      LDR             R5,      = 0xFFFFFFFF
      STR             R5,     [R4]


;-------------------------------------------------------------
; PASS Routine
;-------------------------------------------------------------
PASS
      LDR             R4,      = PASS_FLAG
      LDR             R5,      = 0x11111111
      STR             R5,     [R4]

;-------------------------------------------------------------------------------
; Operand DATA AREA
;-------------------------------------------------------------------------------
AREA  test_list, DATA , READONLY

PKA2_SFR_Setting_Value
      ; input value granularity,sign,bit size...
      ; for 1 modular multiplication,,,

      ; WORD0 => PLDM_ON[19], EXE_ON[16], CHNK_SZ[14:11], PREC_ID[9:8], STOP_SIGN
      ; WORD1 => A,B,S,M_SEG_ID, following bit field defined in PKA2_CTRL_REG2
      ;       page of PKA2 manual
      ; WORD2 => SEG_SIGN, following bit field defined in PKA2_CTRL_REG3 page of
      ;        PKA2 manual
      ; WORD3 => SEG_SIZE[6:5], FUNC_ID[0]
      ;       following bit field defined in PKA2_CTRL_REG5 page of PKA2 manual

; Test Data #1
; PLDM_ON = 1, CHNK_SZ = 0011 (128b), PREC_ID = 00 (Single)
;             1098_7654_3210_9876_5432_1098_7654_3210
; WORD0 = 0000_0000_0000_1000_0001_1000_0000_0000 (in binary)

; A_SEG_ID = 00000 (0), B_SEG_ID = 00001 (1), M_SEG_ID = 00010 (2), S_SEG_ID = 00011 (3)
```

```
;             1098_7654_3210_9876_5432_1098_7654_3210
;SEG_ID = 0000_0000_0000_0001_0000_0010_0000_0011  (in binary)

;             1098_7654_3210_9876_5432_1098_7654_3210
; SEG_SIGN = 0000_0000_0000_0000_0000_0000_0000_0000 (in binary)

; SEG_SIZE = 10 (64B), FUNC_ID = 0
;             1098_7654_3210_9876_5432_1098_7654_3210
;      WORD3 = 0000_0000_0000_0000_0000_0000_0100_0000 (in binary)

      DCD     0x00081800, 0x00010203
      DCD     0x00000000, 0x00000040
...

;-------------------------------------------------------------------------------
; Input data (A, B, M) for test vector
;
;=============================================
; Example of input data                      |
;                                            |
; Original data                              |
; [MSB]                      [LSB]           |
; a1c579c5_71899f06_04e80c73_a954c874        |
;                                            |
; Changed ARM code                           |
; [MSB]                                      |
; 0xa1c579c5, 0x71899f06                     |
;                    [LSB]                   |
; 0x04e80c73, 0xa954c874                     |
;=============================================
;
;-------------------------------------------------------------------------------
PKA2_DATA_IN

; Test Data #1 ==========================
; key_size = 128-bit
; PREC_ID = 00 (Single), CHNK_SZ = 0011 (128b)
; PLDM_ON = 1, FUNC_ID = 0 (A x B)
;=====================================

; load a (ina_p.dat)
      DCD     0xa1c579c5, 0x71899f06
      DCD     0x04e80c73, 0xa954c874

; load b (inb_p.dat)
      DCD     0x744642de, 0x0a3c773f
      DCD     0xf38d11fb, 0x9d8ef14d
```

```
; load m (inm_0.dat)
      DCD     0xc5b9235b, 0xdf3d7edc
      DCD     0x8e1fe09a, 0xf4f25f65


...

;-------------------------------------------------------------------------------
; Expected result S
;-------------------------------------------------------------------------------
PKA2_EXPECTED_RESULT

; Expected Data #1 ==============================
; Pred_ID = Single, Chunk_SZ = 128b (out_p_p_0_p.dat)
;================================================
      DCD     0x403c8f44, 0x6c623465
      DCD     0x44092e40, 0x2725eb28
...
```

54.5.7 Interrupt Service Routine

The following code is an example Interrupt Service Routine.

```
volatile bool EndOfBRDMA;
volatile bool EndOfBTDMA;
volatile bool EndOfHRDMA;
volatile bool PRNG_Done;
volatile bool MSG_Done;
volatile bool Partial_Done;
volatile bool PRNG_Error;

__irq void IRQ_Handler (void) {
      ......
      If (FCINTSTAT & 0x8) {
              EndOfBRDMA = true;
              FCINTPEND = 0x8;                                  // Clears the pending interrupt
              // Do more operation here
      }
      If (FCINTSTAT & 0x4)   {
              EndOfBTDMA = true;
              FCINTPEND = 0x4;                                  // Clears the pending interrupt
              // Do more operation here
      }
      If (FCINTSTAT & 0x2)   {
              EndOfHRDMA = true;
              FCINTPEND = 0x2;                                  // Clears the pending interrupt
              // Do more operation here
      }
      ......
}
```

54.5.8 Useful Routines

The following code is an example of useful routines.

```
//-------------------------------------------------------------------
// copy words from the source to the destination
//-------------------------------------------------------------------
void Copy (u32 sa, u32 da, u32 words)
{
	u32 i;
	for (i = 0; i < words; i++)
		*(u32 *)(da + i*4) = *(u32 *)(sa + i*4);
}
```

55 Keypad Interface

55.1 Overview of Keypad Interface

The Keypad Interface block in Exynos 4412 SCP facilitates communication with external keypad devices. The ports multiplexed with GPIO ports provide up to 14 rows and eight columns. You can use keypad interface on port 0 or port 1. Port 0 and port 1 has the same function. You can use any port for the GPIO connection. Port 0 column is using alive power, therefore, it can use wakeup source without any setting. But port 1 column is using normal power, therefore, it can use wakeup source with GPIO setting for retention. Interrupt delivers the events of key press or key release to the CPU.

There are two types of scans in Keypad Interface. They are, Software Scan and Hardware Scan.

In software scan mode, if one of the interrupt occurs from row lines, then the software should scan the column lines using the proper procedure to detect one or multiple key press or release.

In hardware scan mode, if you press any one of the keys, then the hardware reports the row and column number of the pressed key after it scans the column line automatically. Multiple key press support in hardware scan mode is limited to dual key with other row.

It provides interrupt status register bits at the time of key pressed or key released or both cases (when it enables two interrupt conditions). To prevent the switching noises, keypad interface comprise of internal debouncing filter.

Figure 55-1 illustrates the key matrix interface external connection guide.

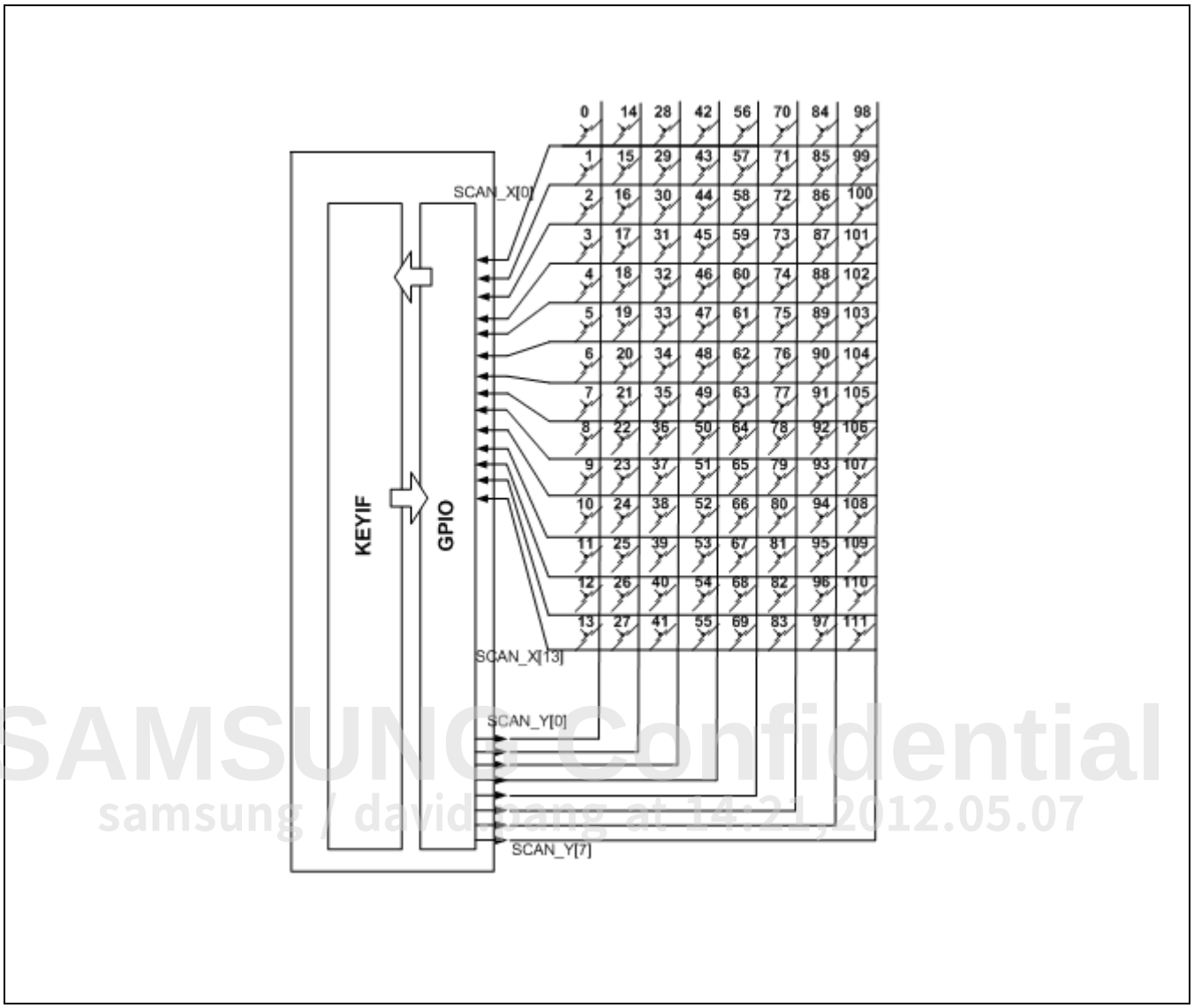


Figure 55-1 Key Matrix Interface External Connection Guide

55.2 Debouncing Filter

Supports debouncing filter for keypad interrupt of any key input. The filtering width is approximately 62.5 usec ("FCLK" two-clock, when the FCLK is 32 kHz). The keypad interrupt (key pressed or key released) to the CPU in software scan mode is an ANDed signal of the all row input lines after filtering.

Figure 55-2 illustrates the internal debouncing filter operation.

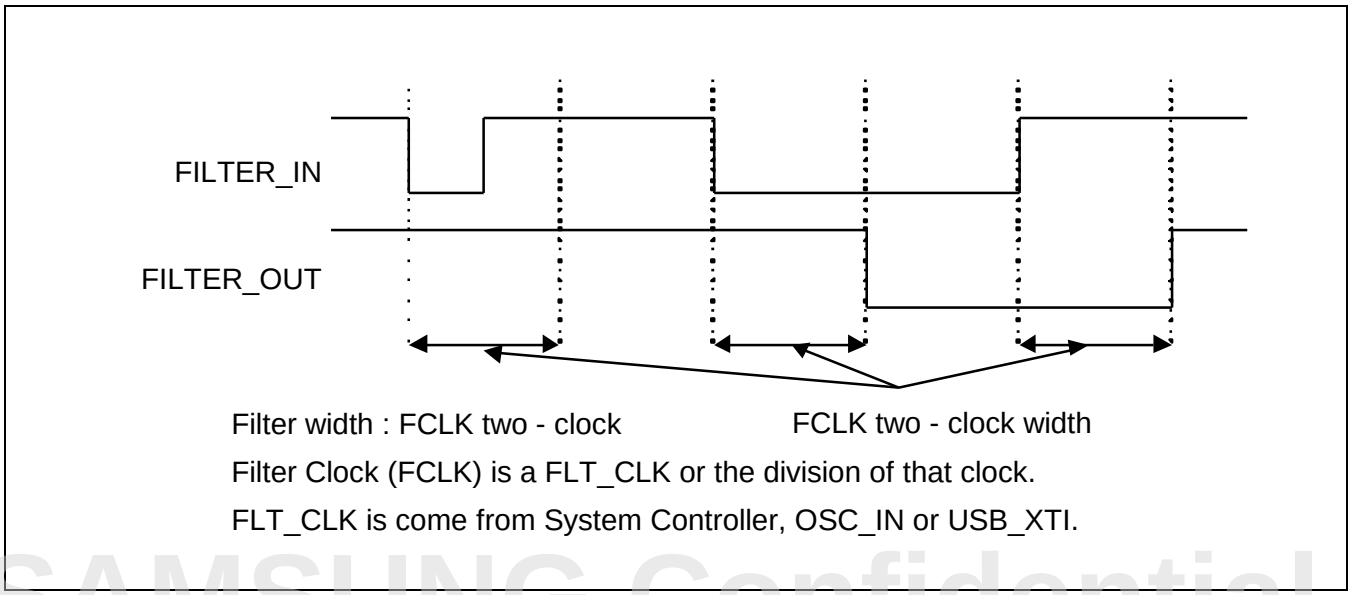


Figure 55-2 Internal Debouncing Filter Operation

55.3 Filter Clock

It divides the KEYPAD interface debouncing filter clock (FCLK) from FLT_CLK that is OSC_IN. You can set compare value for 10-bit up-counter (KEYIFFC). When filter enable bit (FC_EN) is HIGH, filter clock divider is ON. The frequency of FCLK is frequency of FLT_CLK/((KEYIFFC + 1) × 2). On the contrary, if FC_EN is LOW, then the filter clock divider does not divide FLT_CLK.

55.4 Wakeup Source

It uses KEYPAD inputs as a wakeup source. When it uses Key input for wakeup source from Audio playback, STOP, DSTOP, or SLEEP mode, it does not require KEYPAD interface register setting. However, GPIO register (GPX1CON, GPX2CON, GPX3CON, or GPL2CON) should be set for KEYPAD interface and SYSCON register should be set for masking.

55.5 Keypad Scanning Procedure for Software Scan

At initial state, all column lines (outputs) are low level. But column data output tri-state enable bits are all high. Therefore, when it does not use the tri-state enable mode, these bits should be written to zeros. When the status of the key is not pressed, then all row lines (inputs) are high (used pull-up pads). When you press any key, then the corresponding row and column lines are shortened together and a low level is driven on the corresponding row line. This generates a keypad interrupt.

The CPU (software) outputs a LOW on one column line and Hi-Z on the others by setting KEYIFCOLEN and KEYIFCOL fields in KEYIFCOL register. Each time when it writes, the CPU reads the value of the KEYIFROW register and detects if one key of the corresponding column line is pressed. If KEYIF has pull-up PAD, then it reads each KEYIFROW bits as HIGH, except pressed ROW bit. When the scanning procedure ends, it detects the pressed key (one or more).

Figure 55-3 illustrates the keypad scanning procedure.

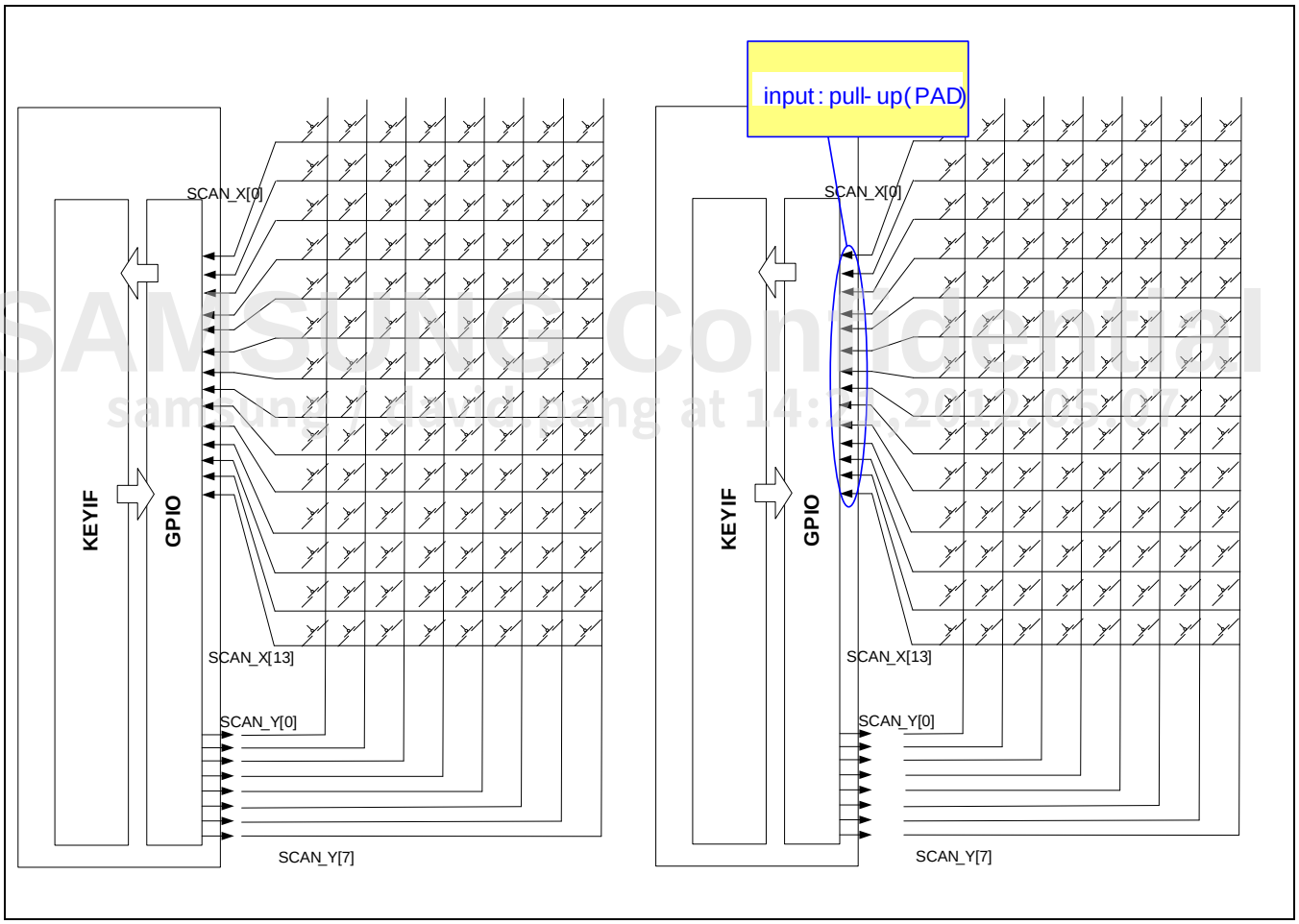


Figure 55-3 Keypad Scanning Procedure

Figure 55-4 illustrates the keypad scanning procedure II.

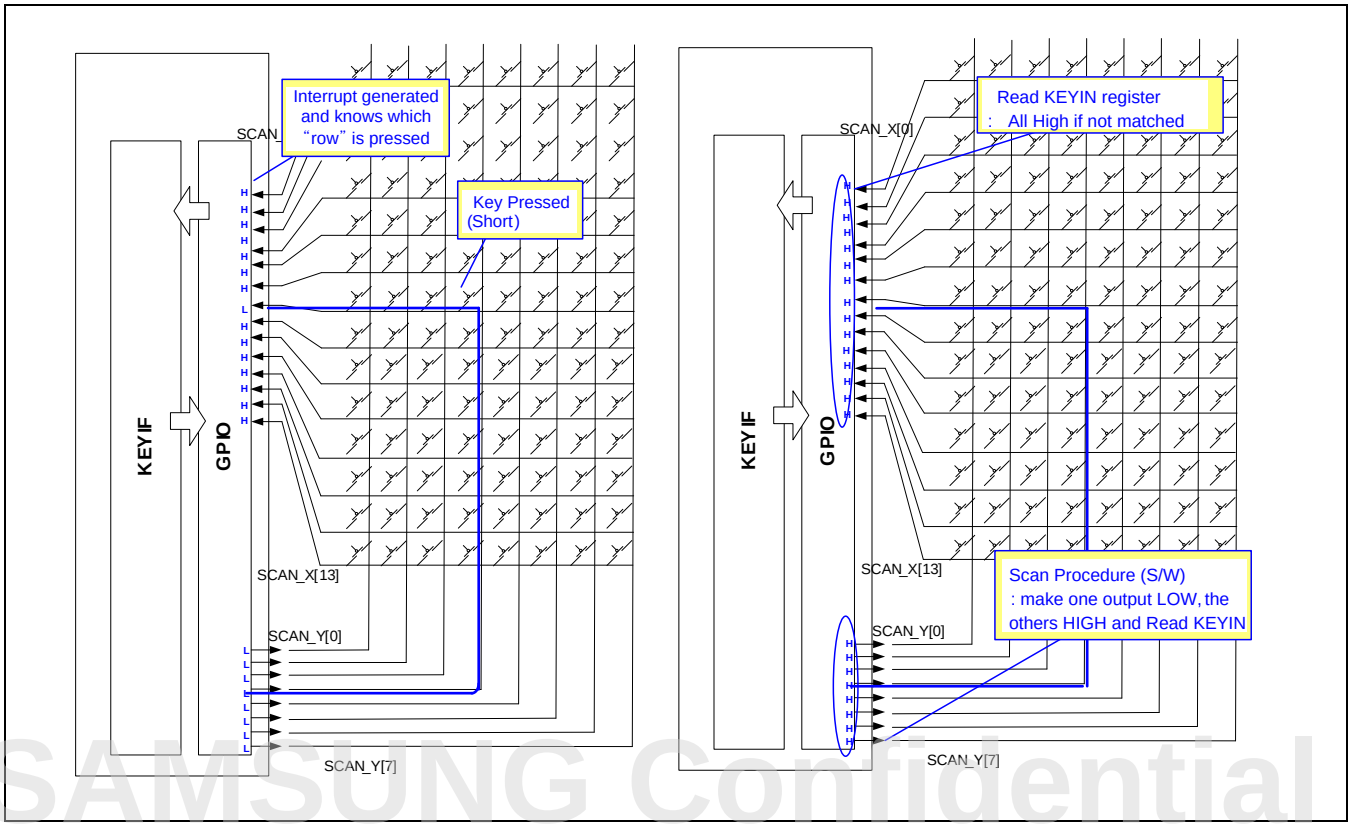


Figure 55-4 Keypad Scanning Procedure II

Figure 55-5 illustrates the keypad scanning procedure III.

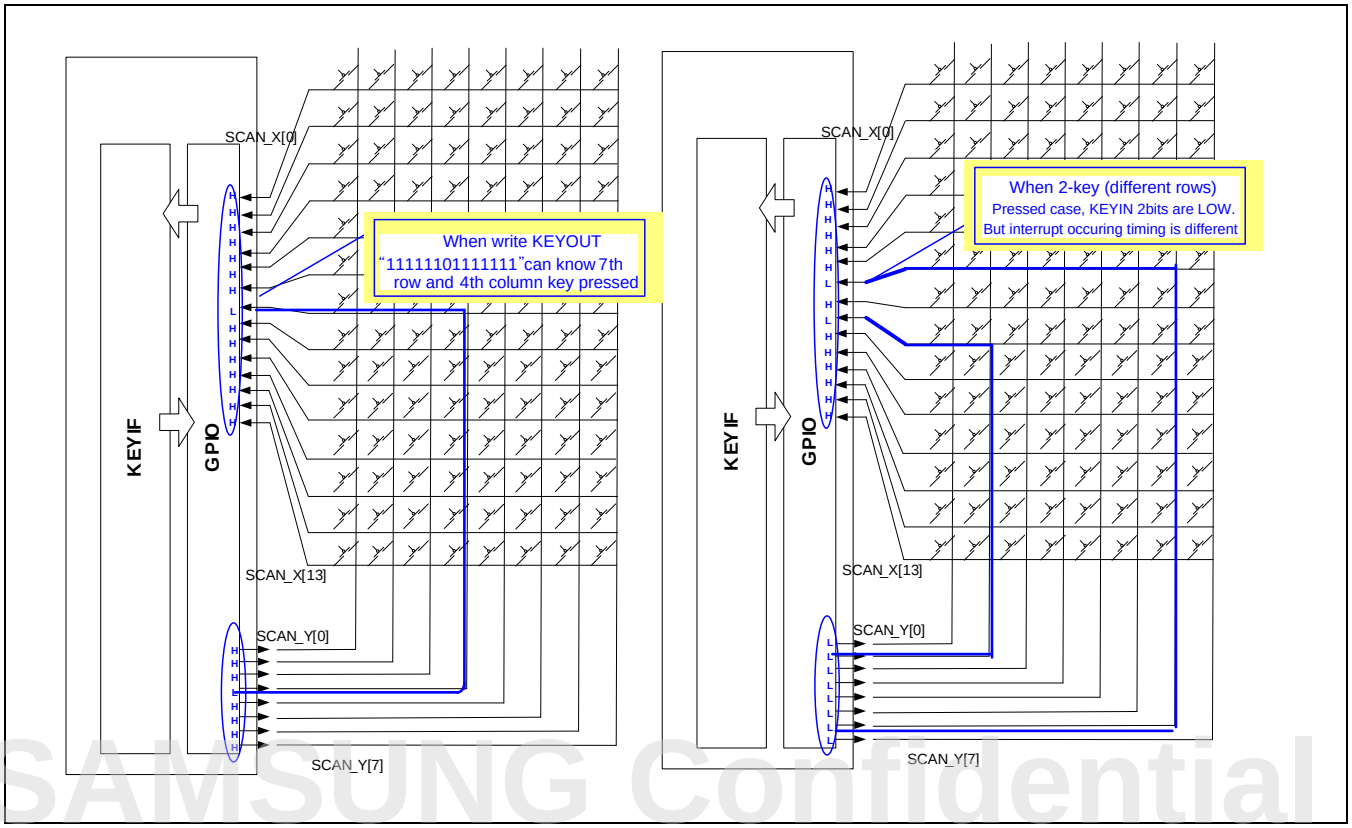


Figure 55-5 Keypad Scanning Procedure III

Figure 55-6 illustrates the keypad scanning procedure when the two-key pressed with different row.

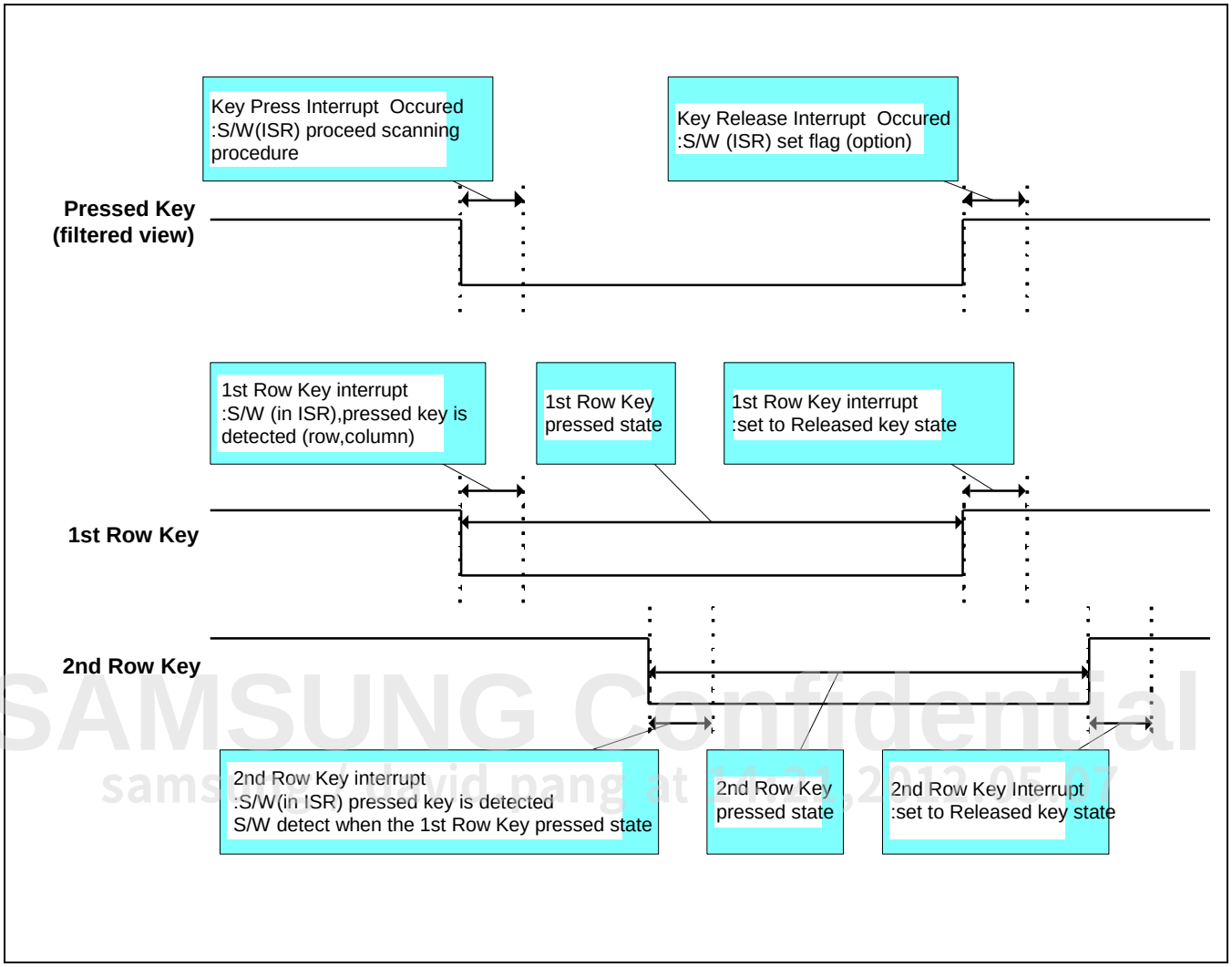


Figure 55-6 Keypad Scanning Procedure when the Two-key Pressed with Different Row

Figure 55-7 illustrates the keypad I/F block diagram.

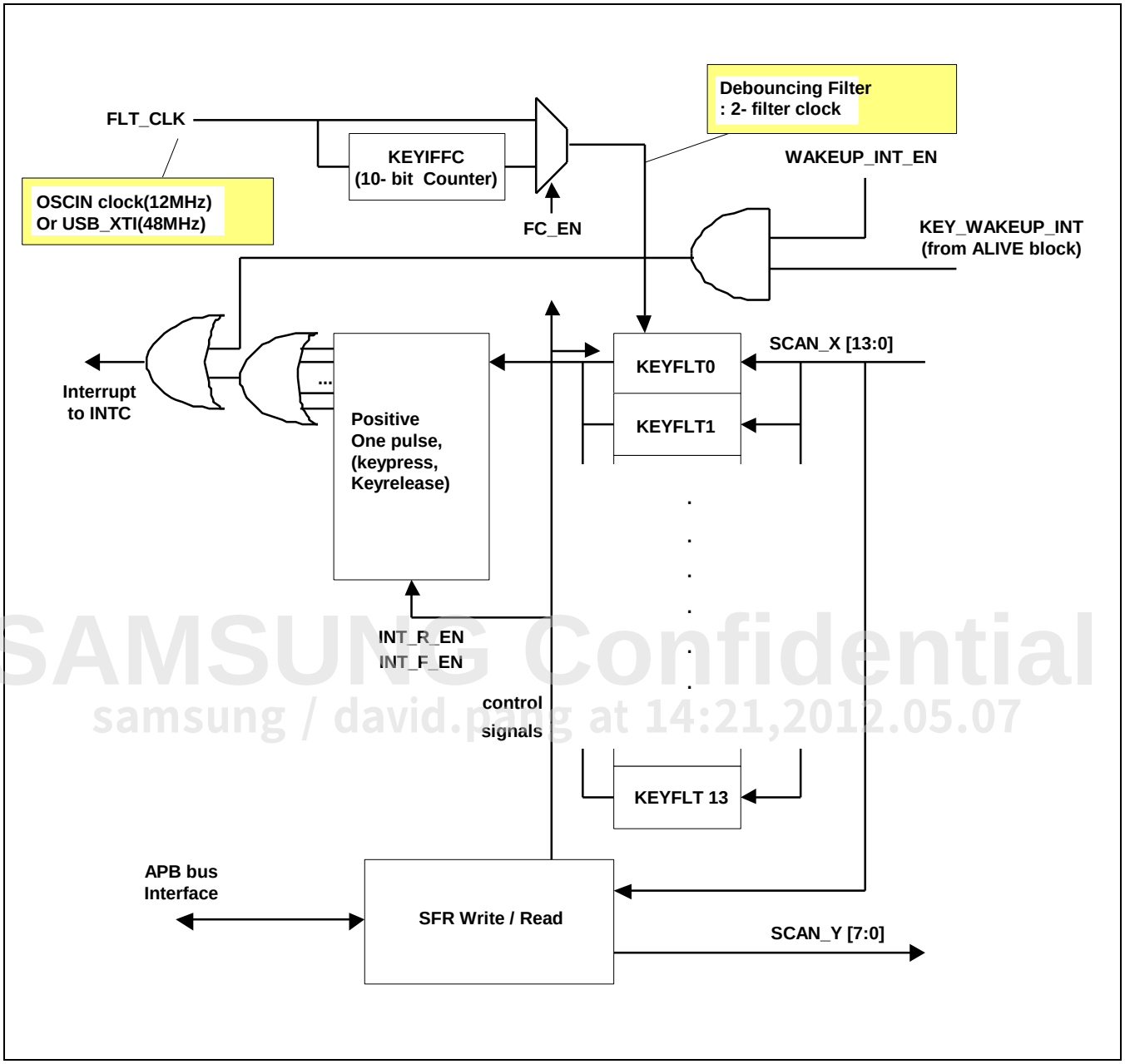


Figure 55-7 Keypad I/F Block Diagram

55.6 Keypad Scanning Procedure for Hardware Scan

At initial stage, the keypad scanning procedures are same as software scan. If any key is pressed, the hardware automatically scans the corresponding row and column lines and writes the information into the register. After scan and write to the register, it generates keypad interrupt.

The CPU (software) can get the row and column number by accessing the KEYIFSCAN1 (first key) or KEYIFSCAN2 (second key) register. The value of KEYIFSCAN1 and KEYIFSCAN2 is valid when the key is pressed. At hardware scan mode, you should set the H_CNT value in KEYIFCON register. The initial value is 0xF. In each scanning step, after driving the column, scanning hardware waits for H_CNT cycle. When the row input signal is stable (after H_CNT cycle), scanning hardware verifies the row input signal.

It limits the multiple key press support in hardware scan mode to dual key with other row.

55.7 I/O Description

Table 55-1 describes the keypad interface I/O.

Table 55-1 Keypad Interface I/O Description

Signal	I/O	Description	Pad		Type
			Port0	Port1	
KP_ROW[13]	I	KEYPAD interface row[13] data	XEINT_29 (GPX3[5])	XEINT_29 (GPX3[5])	muxed
KP_ROW [12]	I	KEYPAD interface row[12] data	XEINT_28 (GPX3[4])	XEINT_28 (GPX3[4])	muxed
KP_ROW [11]	I	KEYPAD interface row[11] data	XEINT_27 (GPX3[3])	XEINT_27 (GPX3[3])	muxed
KP_ROW [10]	I	KEYPAD interface row[10] data	XEINT_26 (GPX3[2])	XEINT_26 (GPX3[2])	muxed
KP_ROW [9]	I	KEYPAD interface row[9] data	XEINT_25 (GPX3[1])	XEINT_25 (GPX3[1])	muxed
KP_ROW [8]	I	KEYPAD interface row[8] data	XEINT_24 (GPX3[0])	XEINT_24 (GPX3[0])	muxed
KP_ROW [7]	I	KEYPAD interface row[7] data	XEINT_23 (GPX2[7])	XEINT_23 (GPX2[7])	muxed
KP_ROW [6]	I	KEYPAD interface row[6] data	XEINT_22 (GPX2[6])	XEINT_22 (GPX2[6])	muxed
KP_ROW [5]	I	KEYPAD interface row[5] data	XEINT_21 (GPX2[5])	XEINT_21 (GPX2[5])	muxed
KP_ROW [4]	I	KEYPAD interface row[4] data	XEINT_20 (GPX2[4])	XEINT_20 (GPX2[4])	muxed
KP_ROW [3]	I	KEYPAD interface row[3] data	XEINT_19 (GPX2[3])	XEINT_19 (GPX2[3])	muxed
KP_ROW [2]	I	KEYPAD interface row[2] data	XEINT_18 (GPX2[2])	XEINT_18 (GPX2[2])	muxed
KP_ROW [1]	I	KEYPAD interface row[1] data	XEINT_17 (GPX2[1])	XEINT_17 (GPX2[1])	muxed
KP_ROW [0]	I	KEYPAD interface row[0] data	XEINT_16 (GPX2[0])	XEINT_16 (GPX2[0])	muxed
KP_COL [7]	O	KEYPAD interface column[7] data	XEINT_15 (GPX1[7])	XGNSS_GPIO_7 (GPL2[7])	muxed
KP_COL [6]	O	KEYPAD interface column[6] data	XEINT_14 (GPX1[6])	XGNSS_GPIO_6 (GPL2[6])	muxed
KP_COL [5]	O	KEYPAD interface column[5] data	XEINT_13 (GPX1[5])	XGNSS_GPIO_5 (GPL2[5])	muxed
KP_COL [4]	O	KEYPAD interface column[4] data	XEINT_12 (GPX1[4])	XGNSS_GPIO_4 (GPL2[4])	muxed
KP_COL [3]	O	KEYPAD interface column[3] data	XEINT_11	XGNSS_GPIO_3	muxed

Signal	I/O	Description	Pad		Type
			Port0	Port1	
			(GPX1[3])	(GPL2[3])	
KP_COL [2]	O	KEYPAD interface column[2] data	XEINT_10 (GPX1[2])	XGNSS_GPIO_2 (GPL2[2])	muxed
KP_COL [1]	O	KEYPAD interface column[1] data	XEINT_9 (GPX1[1])	XGNSS_GPIO_1 (GPL2[1])	muxed
KP_COL [0]	O	KEYPAD INTERFACE COLUMN[0] data	XEINT_8 (GPX1[0])	XGNSS_GPIO_0 (GPL2[0])	muxed

55.8 Register Description

55.8.1 Register Map Summary

- Base Address: 0x100A_0000

Register	Offset	Description	Reset Value
KEYIFCON	0x0000	Specifies KEYPAD interface control register	0x000F_0000
KEYIFSTSCLR	0x0004	Specifies KEYPAD interrupt for software scan status and clear register	0x0000_0000
KEYIFCOL	0x0008	Specifies KEYPAD interface column data output register	0x0000_FF00
KEYIFROW	0x000C	Specifies KEYPAD interface row data input register	Reflects input ports
KEYIFFC	0x0010	Specifies KEYPAD interface debouncing filter clock division register	0x0000_0000
KEYIFSCAN1	0x0014	Specifies KEYPAD interface output result of hardware scan for first key register	0x0000_0000
KEYIFSCAN2	0x0018	Specifies KEYPAD interface output result of hardware scan for second key register	0x0000_0000
KEYIFHSC	0x001C	Specifies KEYPAD interrupt for hardware scan status and clear register	0x0000_0000

55.8.1.1 KEYIFCON

- Base Address : 0x100A_0000
- Address = Base Address + 0x0000, Reset Value = 0x000F_0000

Name	Bit	Type	Description	Reset Value
H_CNT	[31:16]	RW	Counter value for hardware scan column to row interval	16'hF
RSVD	[15:10]	–	Reserved	–
HIZSCAN_EN	[9]	RW	Hi-Z mode scan enable for hardware scan 0 = Normal scan (driving "low and high") 1 = Hi-Z mode scan (driving "low and Hi-Z") In Hi-Z mode, it should disable GPIO internal pull-down.	1'b0
SEL_HSCAN	[8]	RW	Select hardware scan/software scan 0 = Software scan 1 = Hardware scan	1'b0
RSVD	[7:4]	–	Reserved	–
FC_EN	[3]	RW	10-bit counter (for debouncing digital filter clock) enable 0 = Disables. Does not use division counter 1 = Enables. uses division counter	1'b0
DF_EN	[2]	RW	KEYPAD input port debouncing filter enable 0 = Disables 1 = Enables	1'b0
INT_R_EN	[1]	RW	KEYPAD input port rising edge (key-released) interrupt 0 = Disables 1 = Enables	1'b0
INT_F_EN	[0]	RW	KEYPAD input port falling edge (key-pressed) interrupt 0 = Disables 1 = Enables	1'b0

NOTE: Selects both edge interrupt when both INT_F_EN and INT_R_EN are set.

55.8.1.2 KEYIFSTSCLR

- Base Address: 0x100A_0000
- Address = Base Address+ 0x0004, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
R_INT	[29:16]	RW	KEYPAD input "release" interrupt (rising edge) status (read) and clear (write). Read: 0 = Does not occur 1 = Released interrupt occurs Write: Clears released interrupt when write "1" The R_INT[13:0] indicates that each key pressed from 0 to 13 has a dedicated interrupt from R_INT[16] to R_INT[29]	14'b0
P_INT	[13:0]	RW	KEYPAD input "press" interrupt (falling edge) status(read) and clear(write) Read: 0 = Does not occur 1 = Pressed interrupt occurs Write: Clears pressed interrupt when write "1" The P_INT[13:0] indicate that each key released from 0 to 13 has a dedicated interrupt from P_INT[0] to P_INT[13]	14'b0

NOTE: Clears keypad wakeup interrupt when the write access to the KEYIFSTSCLR.

55.8.1.3 KEYIFCOL

- Base Address: 0x100A_0000
- Address = Base Address +0x0008, Reset Value = 0x0000_FF00

Name	Bit	Type	Description	Reset Value
RSVD	[31:16]	–	Reserved	–
KEYIFCOLEN	[15:8]	RW	KEYPAD interface column data output tri-state enable register Each bit is for each KEYIFCOL bit. 0 = Enables output pad tri-state buffer (Normal output, KEY enable) 1 = Disables output pad Tri-state buffer (High-Z output, KEY disable)	8'b1111_1111
KEYIFCOL	[7:0]	RW	KEYPAD interface column data output register	8'b0

55.8.1.4 KEYIFROW

- Base Address: 0x100A_0000
- Address : Base Address +0x000C, Reset Value = Reflects input ports

Name	Bit	Type	Description	Reset Value
RSVD	[31:14]	–	Reserved	–
KEYIFROW	[13:0]	R	KEYPAD interface row data input register (read only) This register values from input ports are not filtered data.	Reflects input ports

55.8.1.5 KEYIFFC

- Base Address: 0x100A_0000
- Address = Base Address+ 0x0010, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:10]	–	Reserved	–
KEYIFFC	[9:0]	RW	KEYPAD interface debouncing filter clock division register. You can set compare value for 10-bit up-counter. This register value means when FC_EN bit is HIGH. FCLK = FLT_CLK/(KEYIFFC[9:0] + 1) (FLT_CLK is from OSC_IN)	10'b0

55.8.1.6 KEYIFSCAN1

- Base Address : 0x100A_0000
- Address = Base Address+ 0x0014, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:22]	–	Reserved	–
ROWSCAN1	[21:8]	R	KEYPAD interface scan result of row (only pressed row has "1") Clears value when first key is released	14'b0
COLSCAN1	[7:0]	R	KEYPAD interface scan result of column (only pressed column has "1") Clears value when first key is released	8'b0

55.8.1.7 KEYIFSCAN2

- Base Address: 0x100A_0000
- Address = Base Address+ 0x0018, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:22]	–	Reserved	–
ROWSCAN2	[21:8]	R	KEYPAD interface scan result of row (only pressed row has "1") Clears value when first key is released	14'b0
COLSCAN2	[7:0]	R	KEYPAD interface scan result of column (only pressed column has "1") Clears value when first key is released	8'b0

55.8.1.8 KEYIFHSC

- Base Address: 0x100A_0000
- Address = Base Address+ 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:4]	–	Reserved	–
HSCAN_R2	[3]	R	KEYPAD input "release" interrupt (rising edge) status(read) and clear(write) for hardware scan of second Key Read: 0 = Does not occur 1 = Released interrupt occurs Write: Clears released interrupt when write "1"	1'b0
HSCAN_P2	[2]	R	KEYPAD input "press" interrupt (falling edge) status(read) and clear(write) for HW scan of second Key Read: 0 = Does not occur 1 = Pressed interrupt occurs Write: Clear pressed interrupt when write "1"	1'b0
HSCAN_R1	[1]	R	KEYPAD input "release" interrupt (rising edge) status(read) and clear(write) for HW scan of first Key Read: 0 = Does not occur 1 = Released interrupt occurs Write: Clears released interrupt when write "1"	1'b0
HSCAN_P1	[0]	R	KEYPAD input "press" interrupt (falling edge) status(read) and clear(write) for hardware scan of first Key Read: 0 = Does not occur 1 = Pressed interrupt occurs Write: Clears pressed interrupt when write "1"	1'b0

56 ADC

This chapter describes the functions and usage of general ADC.

56.1 Overview of ADC

The 10-bit or 12-bit CMOS Analog to Digital Converter (ADC) comprises of 4-channel analog inputs. It converts the analog input signal into 10-bit or 12-bit binary digital codes at a maximum conversion rate of 1MSPS with 5MHz A/D converter clock. A/D converter operates with on-chip sample-and-hold function. ADC supports low power mode.

56.2 KEY Features of ADC for motor control

The ADC includes the following features:

- Resolution: 10-bit / 12-bit (optional)
- Differential Nonlinearity Error: ± 2.0 LSB (Max.)
- Integral Nonlinearity Error: ± 4.0 LSB (Max.)
- Top Offset Error : 0 ~ + 55 LSB
- Bottom Offset Error : 0 ~ - 55 LSB
- Maximum Conversion Rate: 1 MSPS
- Low Power Consumption
- Power Supply Voltage: 1.8V (Typ.), 1.0V (Typ., Digital I/O Interface)
- Analog Input Range: 0 ~ 1.8V

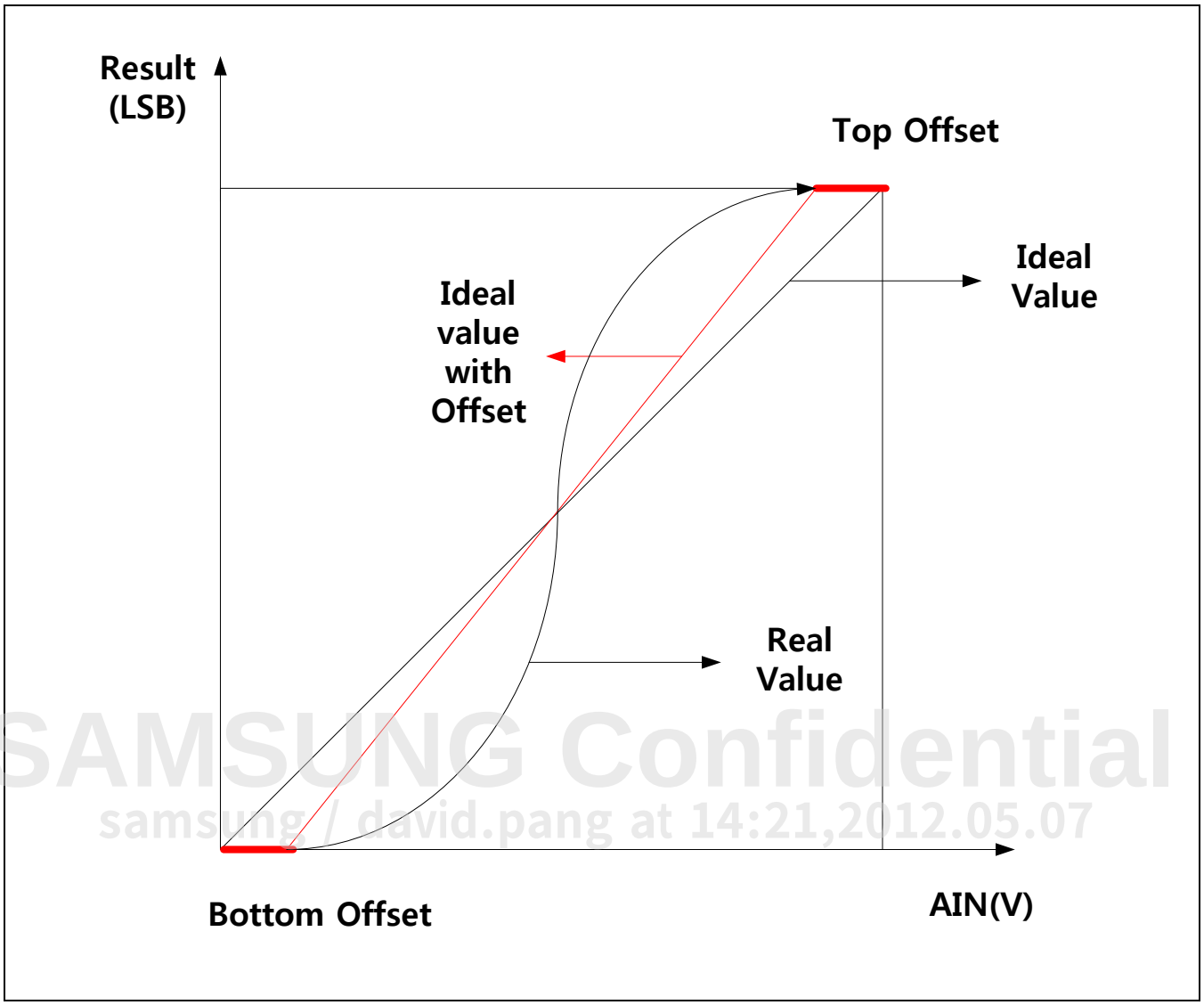


Figure 56-1 ADC Top/Bottom Offset Error Diagram

56.3 ADC Operation

56.3.1 Block Diagram ADC

Figure 56-2 is the functional block diagram of general A/D converter.

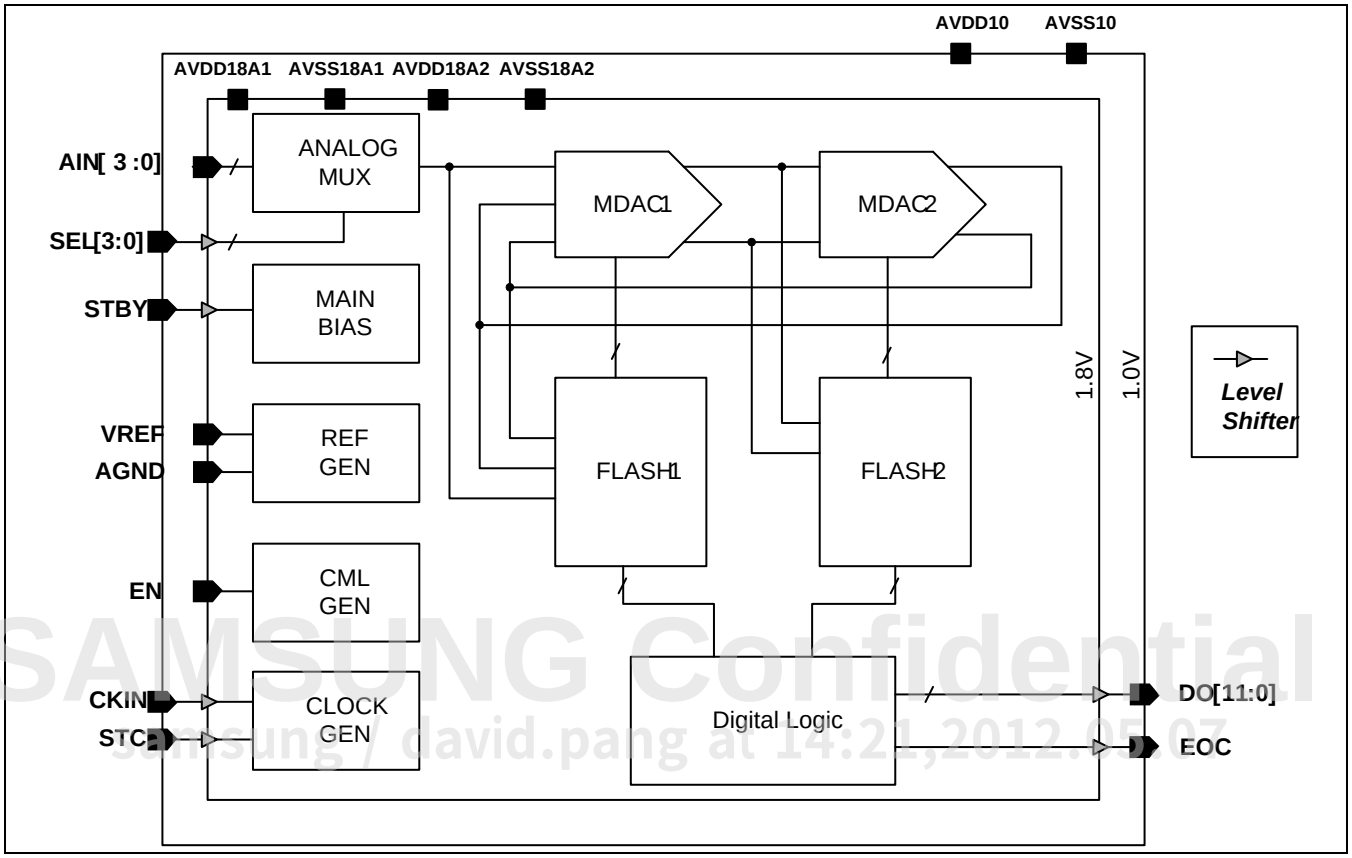


Figure 56-2 ADC Functional Block Diagram

56.4 Function Descriptions

56.4.1 ADC selection

Exynos 4412 SCP has two ADC blocks, General ADC and MTCADC_ISP. User can select one of ADC blocks by setting ADC_CFG[16] bit in System Register SFR.

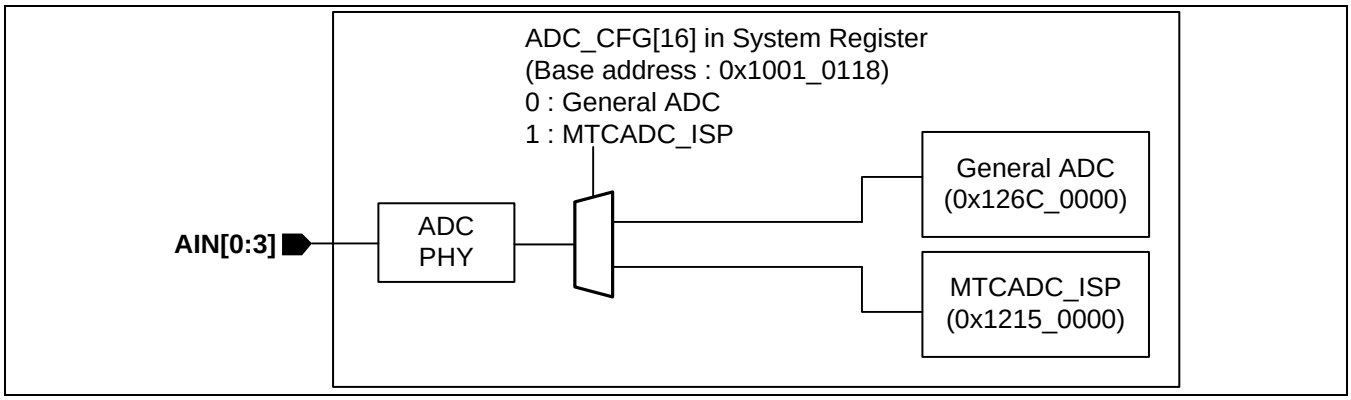


Figure 56-3 ADC selection

56.4.2 A/D Conversion Time

When the APB bus clock(PCLK) frequency is 66MHz and the prescaler value is 65, total 12-bit conversion time is as follows.

- A/D converter freq. = 66MHz/(65+1) = 1MHz
- Conversion time = 1/(1MHz / 5cycles) = 1/200kHz = 5us

NOTE: This A/D converter was designed to operate at maximum 5MHz clock, so the conversion rate can go up to 1MSPS.

56.4.3 ADC conversion Mode

The operation of this mode is same as AIN0~AIN3's. To initialize this mode, set the ADCCON (ADC control register). The converted data can be read out from ADCDAT (ADC conversion data register).

56.4.4 Standby Mode

Standby mode is activated when TSSEL bit is ‘0’ and STANDBY bit is '1' in TSADCCON0 register. In this mode, A/D conversion operation is halted and TSDATXn registers hold their values.

56.4.4.1 Programming Notes

1. The A/D converted data can be accessed by means of interrupt or polling method. With interrupt method, the overall conversion time - from A/D converter start to converted data read - may be delayed because of the return time of interrupt service routine and data access time. With polling method, to determine the read

time for ADCDATXn register, check the ADCCONn[15] – end of conversion flag – bit.

2. A/D conversion can be activated in different way. After ADCCONn[1] - A/D conversion start-by-read mode- is set to 1. A/D conversion starts simultaneously when converted data is read.

56.5 ADC Input Clock Diagram

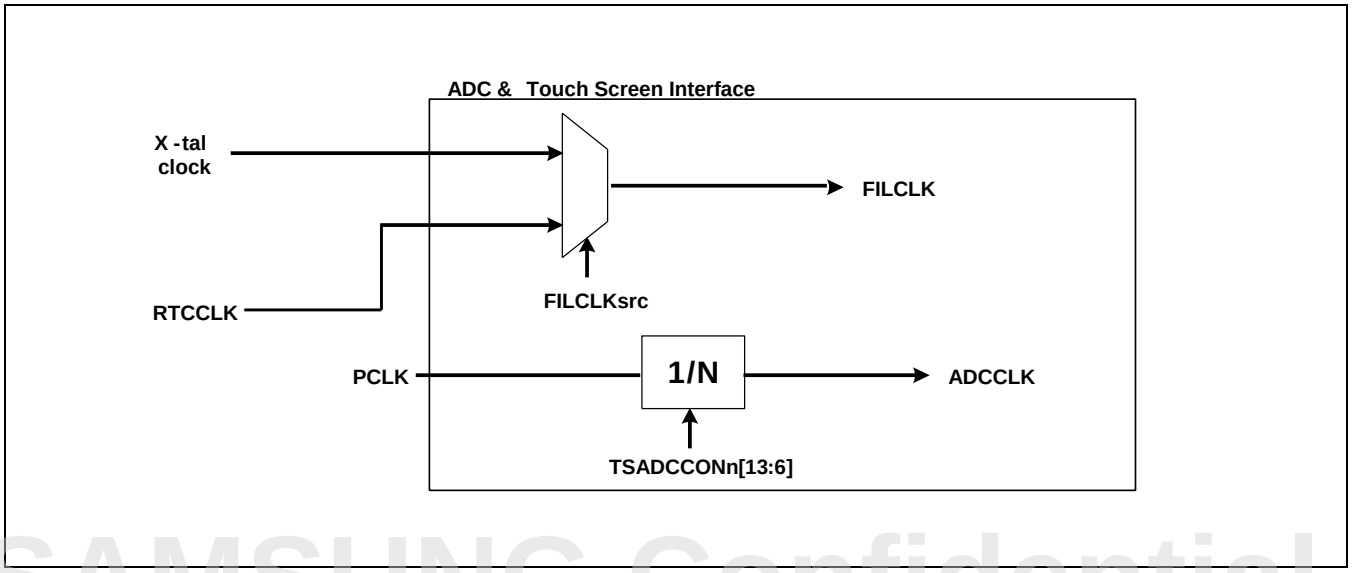


Figure 56-4 Input Clock Diagram for ADC & Touch Screen Interface

56.6 I/O Descriptions

Signal	I/O	Description	Pad	Type
AIN[3]	Input	ADC Channel[3] Analog input	Xadc1AIN_3	Analog
AIN[2]	Input	ADC Channel[2] Analog input	Xadc1AIN_2	Analog
AIN[1]	Input	ADC Channel[1] Analog input	Xadc1AIN_1	Analog
AIN[0]	Input	ADC Channel[0] Analog input	Xadc1AIN_0	Analog

56.7 Register Description

56.7.1 Register Map Summary

- Base Address: 0x126C_0000

Register	Offset	Description	Reset Value
ADCCON	0x0000	ADC Control Register	0x0000_3FC4
ADCDLY	0x0008	ADC Start or Interval Delay Register	0x0000_00FF
ADCDAT	0x000C	ADC Conversion Data Register	Undefined
CLRINTADC	0x0018	Clear ADC Interrupt	Undefined
ADCMUX	0x001C	Specifies the Analog input channel selection	0x0000_0000

56.7.1.1 ADCCON (ADC Control Register)

- Base Address: 0x126C_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_3FC4

Name	Bit	Type	Description	Reset Value
RES	[16]	RW	ADC output resolution selection 0 = 10bit A/D conversion 1 = 12bit A/D conversion	0
ECFLG	[15]	RW	End of conversion flag(Read only) 0 = A/D conversion in process 1 = End of A/D conversion	0
PRSCEN	[14]	RW	A/D converter prescaler enable 0 = Disable 1 = Enable	0
PRSCVL	[13:6]	RW	A/D converter prescaler value Data value: 19 ~ 255 The division factor is (N+1) when the prescaler value is N. For example, ADC frequency is 5MHz if APB bus clock is 100MHz and the prescaler value is 19. Note: This A/D converter is designed to operate at maximum 5MHz clock, so the prescaler value should be set such that the resulting clock does not exceed 5MHz.	0xFF
RSVD	[5:3]	–	Reserved	0
STANDBY	[2]	RW	Standby mode select 0 = Normal operation mode 1 = Standby mode Note: In standby mode, prescaler should be disabled to reduce more leakage power consumption.	1
READ_ START	[1]	RW	A/D conversion start by read 0 = Disables start by read operation 1 = Enables start by read operation	0
ENABLE_ST ART	[0]	RW	A/D conversion starts by enable. If READ_START is enabled, this value is not valid. 0 = No operation 1 = A/D conversion starts and this bit is automatically cleared after the start-up.	0

56.7.1.2 ADCDLY (ADC Delay Register)

- Base Address: 0x126C_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_00FF

Name	Bit	Type	Description	Reset Value
FILCLKsrc	[16]	RW	Reference clock source for delay. 0 = X-tal clock. 1 = RTC clock.	0
DELAY	[15:0]	RW	In case of ADC conversion mode (Normal, Separate, Auto conversion); ADC conversion is delayed by counting this value. Counting clock is PCLK. → ADC conversion delay value. In case of waiting for Interrupt mode: When stylus down occurs in waiting for interrupt mode, it generates interrupt signal (INT_PENn) at interval of several ms for Auto X/Y position conversion. If this interrupt occurs in STOP mode, it generates Wake-Up signal, having interval (several ms), for Exiting STOP MODE. Note: Do not use zero value(0x0000)	00ff

Before ADC conversion, Touch screen uses X-tal clock.

During ADC conversion PCLK (Max. 66MHz) is used.

56.7.1.3 ADCDAT (ADC Conversion Data Register)

- Base Address: 0x126C_0000
- Address = Base Address + 0x000C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
DATA	[11:0]	R	ADC conversion data value Data value: 0x0 ~ 0xFFF	–

56.7.1.4 CLRINTADC (ADC Interrupt Clear Register)

- Base Address: 0x126C_0000
- Address = Base Address + 0x0018, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
INTADCCLR	[0]	W	INT_ADCn interrupt clear. Cleared if any value is written.	–

These registers are used to clear the interrupts. Interrupt service routine is responsible to clear interrupts after the interrupt service is completed. Writing any values on this register will clear up the relevant interrupts asserted. When it is read, undefined value will be returned

56.7.1.5 ADCMUX (ADC Channel Mux Register)

- Base Address: 0x126C_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
SEL_MUX	[3:0]	RW	Analog input channel select 0000 = AIN 0 0001 = AIN 1 0010 = AIN 2 0011 = AIN 3	0

57 Thermal Management Unit

57.1 Overview

In the age of deep sub-micron, dynamic dissipation can result in high temperature. This high temperature may cause a chip to malfunction, and this may lead to expensive cost such as package, cooling device, and so on. To alleviate the phenomenon, it is necessary to manage on-chip temperature so that a chip maintains the proper temperature to continue operation while reducing performance or suspending operation in a specific IP.

The Thermal Management Unit (TMU) in Exynos 4412 SCP provides software controlled (denoted thermal throttling) and hardware controlled (denoted thermal tripping) management scheme. TMU monitors temperature variation in a chip by measuring on-chip temperature, and generates interrupt to CPU when temperature exceeds or goes below pre-defined threshold. Instead of using interrupt generation scheme, CPU can obtain on-chip temperature information by reading the related register field, that is, by using polling scheme. The TMU operating clock is the crystal input clock (XXTI).

Figure 57-1 illustrates the overview of operation of TMU.

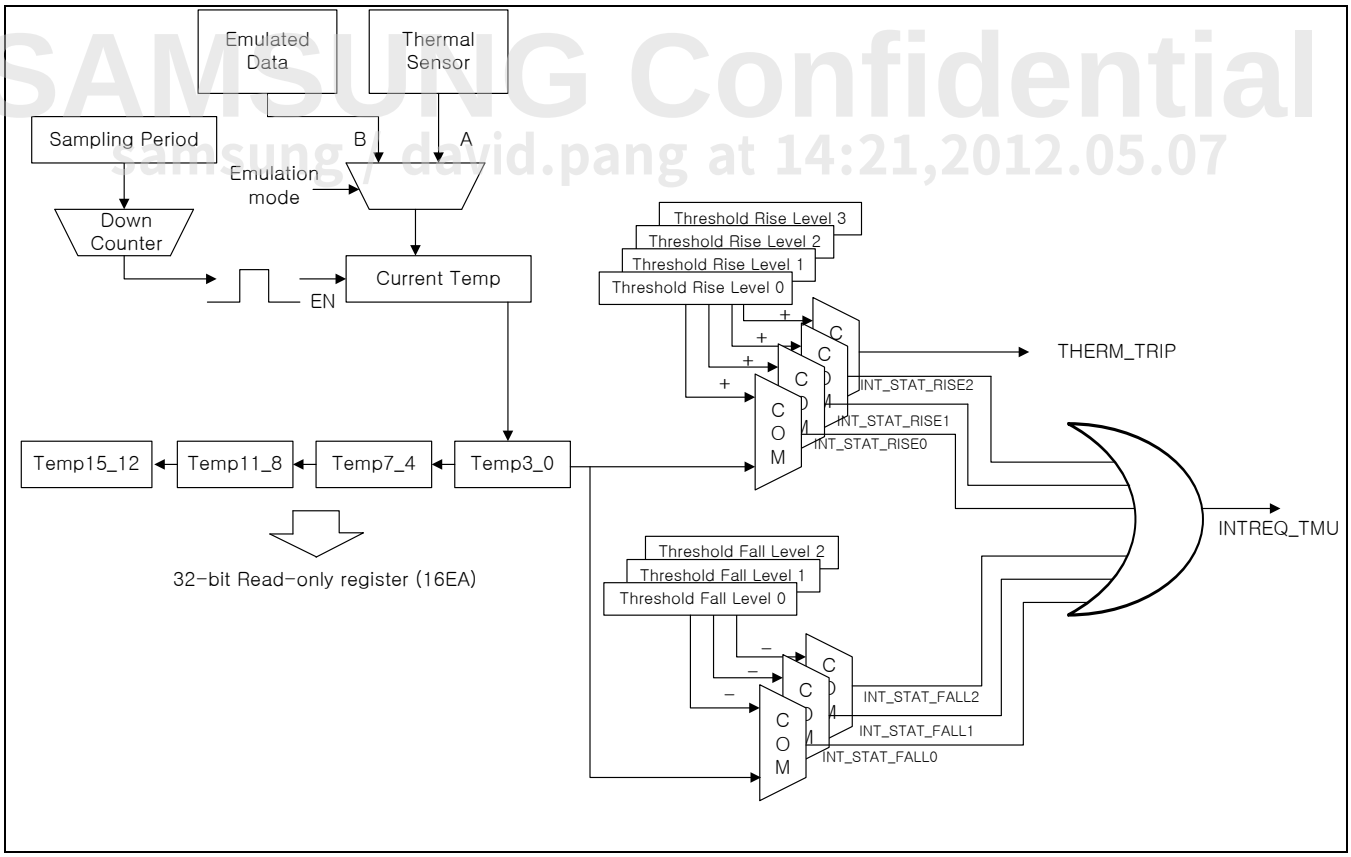


Figure 57-1 Overview of Operation of TMU

The temperature data obtained by temperature sensor contains the measuring error. Therefore, for a normal operation, it should calibrate the temperature data. The software performs the calibration of temperature data. The calibration consists of reading the measured data from e-fuse and writing the calibrated threshold temperature to generate interrupts into THRESHOLD_TEMP_RISE or THRESHOLD_TEMP_FALL register by using calibration equation.

When current temperature exceeds or goes down a threshold temperature, then it generates corresponding interrupt. After reading interrupt status register, it should perform the proper task for that interrupt.

Especially when temperature exceeds an extremely high threshold temperature and if it is denoted as THRES_LEVEL_RISE3 field of THRESHOLD_TEMP_RISE register, then the hot temperature should not damage the Exynos 4412 SCP. In this case, TMU urgently sends active-high signal (THERM_TRIP) to Power Management Unit (PMU), and it performs thermal tripping by hardware logic that is, PMU. Thermal tripping means that PMU cuts off the entire power of Exynos 4412 SCP by controlling external voltage regulator.

57.2 Temperature Sensing Auto Mode with External Clocks

Figure 57-2 illustrates the temperature sensor timing diagram.

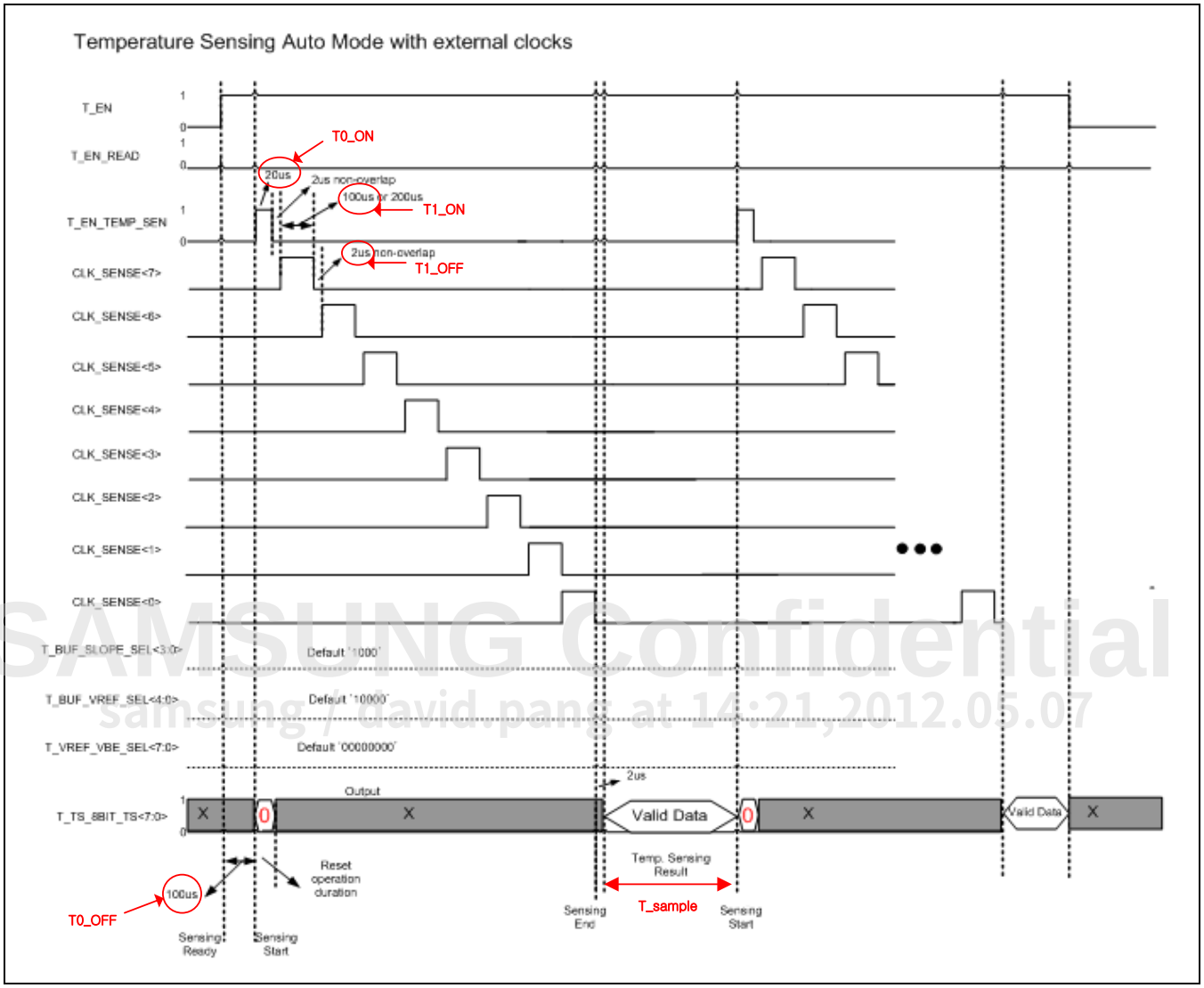


Figure 57-2 Temperature Sensor Timing Diagram

57.3 Temperature Code Table

Table 57-1 lists the complete 8-bit code map to temperature.

Table 57-1 8-bit Code Map to Temperature

8-bit Code	Temp. (°C)	8-bit Code	Temp. (°C)	8-bit Code	Temp. (°C)
100011	10	1001010	49	1110000	87
100100	11	1001011	50	1110001	88
100101	12	1001100	51	1110010	89
100110	13	1001101	52	1110011	90
100111	14	1001110	53	1110100	91
101000	15	1001111	54	1110101	92
101001	16	1010000	55	1110110	93
101010	17	1010001	56	1110111	94
101011	18	1010010	57	1111000	95
101100	19	1010011	58	1111001	96
101101	20	1010100	59	1111010	97
101110	21	1010101	60	1111011	98
101111	22	1010110	61	1111100	99
110000	23	1010110	61	1111101	100
110001	24	1010111	62	1111110	101
110010	25	1011000	63	1111111	102
110011	26	1011001	64	10000000	103
110100	27	1011010	65	10000001	104
110101	28	1011011	66	10000010	105
110110	29	1011100	67	10000011	106
110111	30	1011101	68	10000100	107
111000	31	1011110	69	10000101	108
111001	32	1011111	70	10000110	109
111010	33	1100000	71	10000111	110
111011	34	1100001	72	10001000	111
111100	35	1100010	73	10001001	112
111101	36	1100011	74	10001010	113
111110	37	1100100	75	10001011	114
111111	38	1100101	76	10001100	115
1000000	39	1100110	77	10001101	116
1000001	40	1100111	78	10001110	117
1000010	41	1101000	79	10001111	118
1000011	42	1101001	80	10010000	119

8-bit Code	Temp. (°C)	8-bit Code	Temp. (°C)	8-bit Code	Temp. (°C)
1000100	43	1101010	81	10010001	120
1000101	44	1101011	82	10010010	121
1000110	45	1101100	83	10010011	122
1000111	46	1101101	84	10010100	123
1001000	47	1101110	85	10010101	124
1001001	48	1101111	86	10010110	125

NOTE: For temperature lower than 10 °C, temperature sensor outputs 8-bit code is decreased by one, and for temperature higher than 125 °C, temperature sensor outputs 8-bit code is increased by one. However, this value may be inaccurate.

57.4 I/O Description

Signal	I/O	Description	Pad	Type
T_RES_EXT	Input	External Resistor (100 kΩ, 1 %) Connection Pin.	XtsEXT_RES	Dedicated

57.5 Register Description

57.5.1 Register Map Summary

- Base Address: 0x100C_0000

Register	Offset	Description	Reset Value
TRIMINFO	0x0000	E-fuse information for trimming sensor data	Undefined
RSVD	0x0004 to 0x0010	Reserved	0x0000_0000
TRIMINFO_CONTROL	0x0014	TrimInfo control register	0x0000_0010
RSVD	0x0018 to 0x001C	Reserved	0x0000_0000
TMU_CONTROL	0x0020	TMU control register	0x1000_8802
RSVD	0x0024	Reserved	0x0000_0000
TMU_STATUS	0x0028	TMU Status register	0x0000_0001
SAMPLING _INTERVAL	0x002C	TMU sampling interval control between adjacent sampling points	0x0001_D71A
COUNTER_VALUE0	0x0030	Set time to control EN_TEMP_SEN	0x01E0_0960
COUNTER_VALUE1	0x0034	Set time to control CLK_SENSE	0x0960_0030
RSVD	0x0038 to 0x003C	Reserved	0x0000_0000
CURRENT_TEMP	0x0040	Current temperature information	0x0000_0000
RSVD	0x0044 to 0x004C	Reserved	0x0000_0000
THRESHOLD_TEMP _RISE	0x0050	Threshold for temperature rising	0x645A_5046
THRESHOLD_TEMP _FALL	0x0054	Threshold for temperature falling	0x645A_5046
RSVD	0x0058 to 0x005C	Reserved	0x0000_0000
PAST_TEMP3_0	0x0060	Past temperature 3-0 for tracing temperature	0x0000_0000
PAST_TEMP7_4	0x0064	Past temperature 7-4 for tracing temperature	0x0000_0000
PAST_TEMP11_8	0x0068	Past temperature 11-8 for tracing temperature	0x0000_0000
PAST_TEMP15_12	0x006C	Past temperature 15-12 for tracing temperature	0x0000_0000
INTEN	0x0070	Interrupt enable register	0x0000_0000
INTSTAT	0x0074	Interrupt status register	0x0000_0000
INTCLEAR	0x0078	Interrupt clear register	0x0000_0000
RSVD	0x007C	Reserved	0x0000_0000
EMUL_CON	0x0080	Emulation control register	0x0001_0000
RSVD	0x0084 to 0xFFFF	Reserved	0x0000_0000

57.5.1.1 TRIMINFO

- Base Address: 0x100C_0000
- Address = Base Address + 0x000C, Reset Value = Undefined

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	X
TRIMINFO_25	[7:0]	R	E-fuse information for trimming temperature sensor error for 25 °C.	–

NOTE: Section *57.6 Programming Guide* describes the programming guide for the usage of this register.

57.5.1.2 TRIMINFO_CONTROL

- Base Address: 0x100C_0000
- Address = Base Address + 0x0014, Reset Value = 0x00000010

Name	Bit	Type	Description	Reset Value
RSVD	[31:6]	–	Reserved	0x000
ACTIME	[5:4]	RW	AC time	01
RSVD	[3:1]	–	Reserved	000
RELOAD	[0]	RW	RELOAD Trim information. Before read TRIMINFO, you shall set RELOAD to 1. 1 = Reload	0

57.5.1.3 TMU_CONTROL

- Base Address: 0x100C_0000
- Address = Base Address + 0x0020, Reset Value = 0x1000_8802

Name	Bit	Type	Description	Reset Value
RSVD	[31:29]	–	Reserved	0
BUF_VREF_SEL	[28:24]	RW	Setting the reference voltage of amplifier in the positive-TC generator block. It may change the default value after analyzing the property of mass data.	0x10
Reserved	[23]	-	Reserved	0x0
MUX_ADDR	[22:20]	RW	Test MUX address setting bits. This bit should be set to 3'b110.	0x0
Reserved	[19:16]	-	Reserved	0x0
THERM_TRIP_MODE	[15:13]	RW	Select thermal tripping mode Value of 000 means it does not consider the noise by power supply and so on. Values of 100, 101, 110, and 111 means, it enables noise cancellation mode, and defines the range of data which is considered. When it enables noise cancellation mode, it triggers thermal tripping event when all data considered exceeds THRES_LEVEL_RISE3 field of THRESHOLD_TEMP_RISE. 000 = Considers only CURRENT_TEMP 100 = Considers CURRENT_TEMP and PAST_TEMP3_0 101 = Considers CURRENT_TEMP, PAST_TEMP7_4, and PAST_TEMP3_0. 110 = Considers CURRENT_TEMP, PAST_TEMP11_8, PAST_TEMP7_4, and PAST_TEMP3_0. 111 = Considers CURRENT_TEMP,PAST_TEMP15_12, PAST_TEMP11_8, PAST_TEMP7_4, and PAST_TEMP0-3	0x4
THERM_TRIP_EN	[12]	RW	Thermal Tripping Enable 0 = Disables 1 = Enables	0
BUF_SLOPE_SEL	[11:8]	RW	Setting the gain of amplifier in the positive-TC generator block. It may change the default value after analyzing the property of mass data.	0x8
RSVD	[7:6]	–	Reserved	0
RSVD	[5:1]	–	Reserved	0x01
CORE_EN	[0]	RW	TMU core (state machine) enable/disable 0 = Disables 1 = Enables	0

57.5.1.4 TMU_STATUS

- Base Address: 0x100C_0000
- Address = Base Address + 0x002, Reset Value = 0x0000_0001

Name	Bit	Type	Description	Reset Value
RSVD	[31:1]	–	Reserved	0x00
TMU_IDLE	[0]	R	Indicates that sensing operation is idle. 0 = BUSY 1 = IDLE	1

57.5.1.5 SAMPLING_INTERVAL

- Base Address: 0x100C_0000
- Address = Base Address + 0x002C, Reset Value = 0x1000_D71A

Name	Bit	Type	Description	Reset Value
SAMPLE_ INTERVAL	[31:0]	RW	Sampling interval control for sensing temperature sensor. T_sample = SAMPLE_INTERVAL x Input Clock Period (Refer to *Figure 57-2* for more information.) SAMPLE_INTERVAL should be equal to or greater than 0x1.	0x0001_D71A

57.5.1.6 COUNTER_VALUE0

- Base Address: 0x100C_0000
- Address = Base Address + 0x0030, Reset Value = 0x01E0_0960

Name	Bit	Type	Description	Reset Value
EN_TEMP_SEN_ ON	[31:16]	RW	Timing control between T_EN_TEMP_SEN_ON (rising edge) and T_EN_TEMP_SEN_ON (falling edge) T0_ON = EN_TEMP_SEN_ON x Input Clock Period (Refer to *Figure 57-2* for more information.) T0_ON should be equal or greater than 20 µs.	0x0000_01E0
EN_TEMP_SEN_ OFF	[15:0]	RW	Timing control between T_EN (rising edge) and T_EN_TEMP_SEN_ON (rising edge) T0_OFF = EN_TEMP_SEN_OFF x Input Clock Period (Refer to *Figure 57-2* for more information.) T0_OFF should be equal or greater than 100 µs.	0x0000_0960

57.5.1.7 COUNTER_VALUE1

- Base Address: 0x100C_0000
- Address = Base Address + 0x0034, Reset Value = 0x0960_0030

Name	Bit	Type	Description	Reset Value
CLK_SENSE_ON	[31:16]	RW	Timing control between CLK_SENSE_ON[7:0] (rising edge) and CLK_SENSE_ON[7:0] (falling edge) T1_ON = CLK_SENSE_ON x Input Clock Period (Refer to *Figure 57-2* for more information.) T1_ON should be equal or greater than 100 µs.	0x0000_0960
CLK_SENSE_OFF	[15:0]	RW	Timing control between CLK_SENSE_ON[7:0] (falling edge) and CLK_SENSE_ON[7:0] (falling edge) T1_OFF = CLK_SENSE_OFF x Input Clock Period (Refer to *Figure 57-2* for more information.) T1_OFF should be equal or greater than 2 µs.	0x0000_0030

57.5.1.8 CURRENT_TEMP

- Base Address: 0x100C_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:8]	–	Reserved	0x0
CURRENT_TEMP	[7:0]	R	Current Temperature Refer to *Table 57-1* for more information.	0x0

57.5.1.9 THRESHOLD_TEMP_RISE

- Base Address: 0x100C_0000
- Address = Base Address + 0x0050, Reset Value = 0x645A_5046

Name	Bit	Type	Description	Reset Value
THRES_LEVEL_RISE3	[31:24]	RW	Threshold for temperature rising to generate interrupt 3	0x64
THRES_LEVEL_RISE2	[23:16]	RW	Threshold for temperature rising to generate interrupt 2	0x5A
THRES_LEVEL_RISE1	[15:8]	RW	Threshold for temperature rising to generate interrupt 1	0x50
THRES_LEVEL_RISE0	[7:0]	RW	Threshold for temperature rising to generate interrupt 0	0x46

57.5.1.10 THRESHOLD_TEMP_FALL

- Base Address: 0x100C_0000
- Address = Base Address + 0x0054, Reset Value = 0x645A_5046

Name	Bit	Type	Description	Reset Value
RSVD	[31:24]	–	Reserved	0x64
THRES_LEVEL_FALL2	[23:16]	RW	Threshold for temperature falling to generate interrupt 2	0x5A
THRES_LEVEL_FALL1	[15:8]	RW	Threshold for temperature falling to generate interrupt 1	0x50
THRES_LEVEL_FALL0	[7:0]	RW	Threshold for temperature falling to generate interrupt 0	0x46

57.5.1.11 PAST_TEMP0_3

- Base Address: 0x100C_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PAST_TEMP3	[31:24]	R	Past Temperature 3	0x0
PAST_TEMP2	[23:16]	R	Past Temperature 2	0x0
PAST_TEMP1	[15:8]	R	Past Temperature 1	0x0
PAST_TEMP0	[7:0]	R	Past Temperature 0	0x0

57.5.1.12 PAST_TEMP7_4

- Base Address: 0x100C_0000
- Address = Base Address + 0x0064, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PAST_TEMP7	[31:24]	R	Past Temperature 7	0x0
PAST_TEMP6	[23:16]	R	Past Temperature 6	0x0
PAST_TEMP5	[15:8]	R	Past Temperature 5	0x0
PAST_TEMP4	[7:0]	R	Past Temperature 4	0x0

57.5.1.13 PAST_TEMP11_8

- Base Address: 0x100C_0000
- Address = Base Address + 0x0068, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PAST_TEMP11	[31:24]	R	Past Temperature 11	0x0
PAST_TEMP10	[23:16]	R	Past Temperature 10	0x0
PAST_TEMP9	[15:8]	R	Past Temperature 9	0x0
PAST_TEMP8	[7:0]	R	Past Temperature 8	0x0

57.5.1.14 PAST_TEMP15_12

- Base Address: 0x100C_0000
- Address = Base Address + 0x006C, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
PAST_TEMP15	[31:24]	R	Past Temperature 15	0x0
PAST_TEMP14	[23:16]	R	Past Temperature 14	0x0
PAST_TEMP13	[15:8]	R	Past Temperature 13	0x0
PAST_TEMP12	[7:0]	R	Past Temperature 12	0x0

57.5.1.15 INTEN

- Base Address: 0x100C_0000
- Address = Base Address + 0x0070, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
INTEN_FALL2	[24]	RW	Enables INTREQ_FALL[2]	0
RSVD	[23:21]	–	Reserved	0x0
INTEN_FALL1	[20]	RW	Enables INTREQ_FALL[1]	0
RSVD	[19:17]	–	Reserved	0x0
INTEN_FALL0	[16]	RW	Enables INTREQ_FALL[0]	0
RSVD	[15:9]	–	Reserved	0x0
INTEN_RISE2	[8]	RW	Enables INTREQ_RISE[2]	0
Reserved	[7:5]	–	Reserved	0x0
INTEN_RISE1	[4]	RW	Enables INTREQ_RISE[1]	0
RSVD	[3:1]	–	Reserved	0x0
INTEN_RISE0	[0]	RW	Enables INTREQ_RISE[0]	0

57.5.1.16 INTSTAT

- Base Address: 0x100C_0000
- Address = Base Address + 0x0074, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:25]	–	Reserved	0x0
INTSTAT_FALL2	[24]	R	INTREQ_FALL[2] status 0 = Not pending 1 = Pending	0
RSVD	[23:21]	–	Reserved	0x0
INTSTAT_FALL1	[20]	R	INTREQ_FALL[1] status 0 = Not pending 1 = Pending	0
RSVD	[19:17]	–	Reserved	0x0
INTSTAT_FALL0	[16]	R	INTREQ_FALL[0] status 0 = Not pending 1 = Pending	0
RSVD	[15:9]	–	Reserved	0x0
INTSTAT_RISE2	[8]	R	INTREQ_RISE[2] status 0 = Not pending 1 = Pending	0
RSVD	[7:5]	–	Reserved	0x0
INTSTAT_RISE1	[4]	R	INTREQ_RISE[1] status 0 = Not pending 1 = Pending	0
RSVD	[3:1]	–	Reserved	0x0
INTSTAT_RISE0	[0]	R	INTREQ_RISE[0] status 0 = Not pending 1 = Pending	0

When current temperature exceeds a threshold rise temperature, then it generates corresponding interrupt (INTREQ_RISE[2:0]).

When current temperature goes below a threshold fall temperature, then it generates corresponding interrupt (INTREQ_FALL[2:0].

Figure 57-3 illustrates the example of temperature profile and interrupt generation.

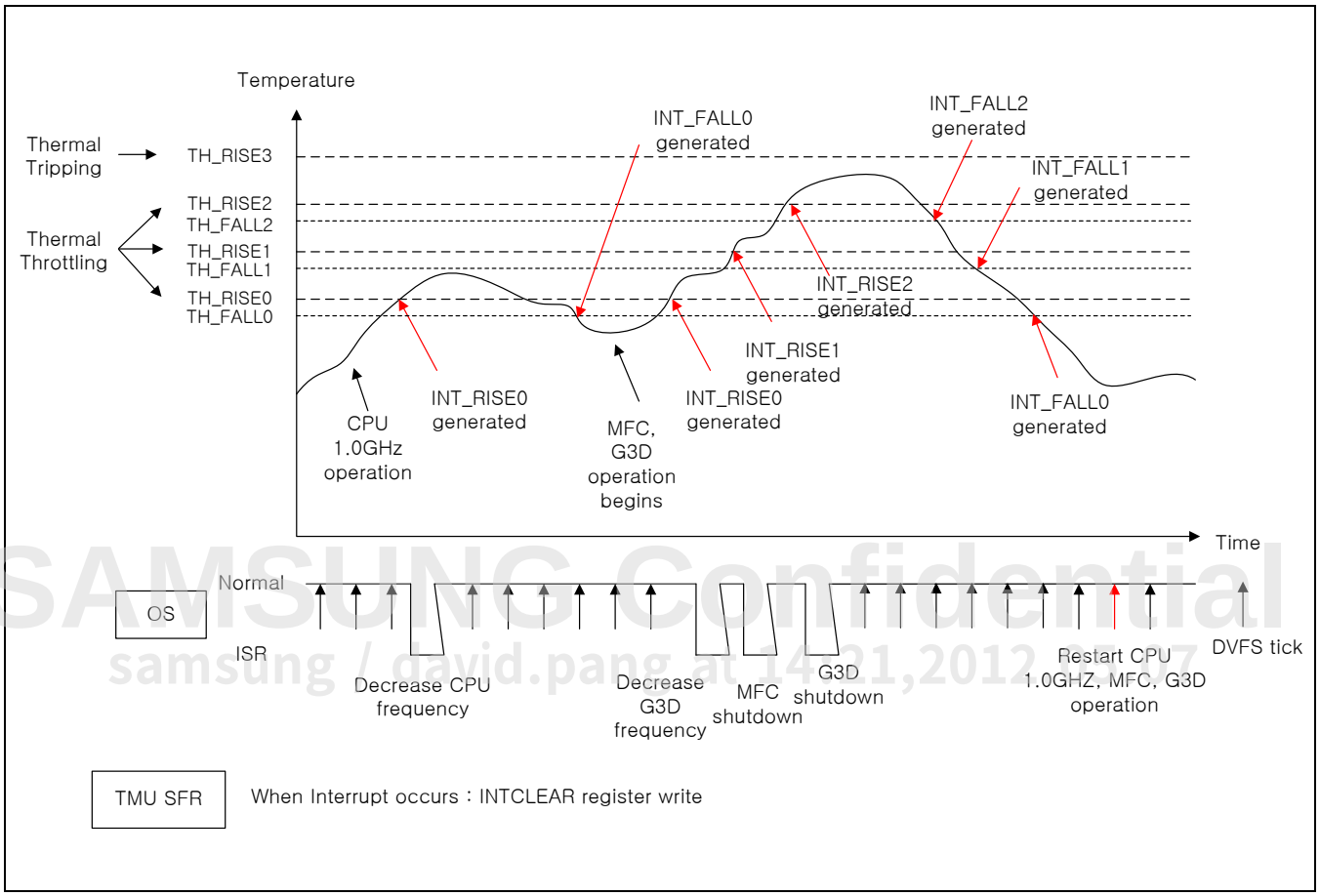


Figure 57-3 Temperature Profile and Interrupt Generation

57.5.1.17 INTCLEAR

- Base Address: 0x100C_0000
- Address = Base Address + 0x0078, Reset Value = 0x0000_0000

Name	Bit	Type	Description	Reset Value
RSVD	[31:21]	–	Reserved	0x0
INTCLEAR_FALL2	[20]	RW	Clear interrupt pending bit of INTREQ_FALL[2] Clears this bit automatically after writing 1'b1. 1 = Clear pending interrupt status	0
RSVD	[19:17]	–	Reserved	0x0
INTCLEAR_FALL1	[16]	RW	Clear interrupt pending bit of INTREQ_FALL[1] Clears this bit automatically after writing 1'b1. 1 = Clear pending interrupt status	0
RSVD	[15:13]	–	Reserved	0x0
INTCLEAR_FALL0	[12]	RW	Clear interrupt pending bit of INTREQ_FALL[0] Clears this bit automatically after writing 1'b1. 1 = Clear pending interrupt status	0
RSVD	[11:9]	–	Reserved	0x0
INTCLEAR_RISE2	[8]	RW	Clear interrupt pending bit of INTREQ_RISE[2] Clears this bit automatically after writing 1'b1. 1 = Clear pending interrupt status	0
RSVD	[7:5]	–	Reserved	0x0
INTCLEAR_RISE1	[4]	RW	Clear interrupt pending bit of INTREQ_RISE[1] Clears this bit automatically after writing 1'b1. 1 = Clear pending interrupt status	0
RSVD	[3:1]	–	Reserved	0x0
INTCLEAR_RISE0	[0]	RW	Clear interrupt pending bit of INTREQ_RISE[0] Clears this bit automatically after writing 1'b1. 1 = Clear pending interrupt status	0

57.5.1.18 EMUL_CON

- Base Address: 0x100C_0000
- Address = Base Address + 0x0080, Reset Value = 0x0001_0000

Name	Bit	Type	Description	Reset Value
NEXT_TIME	[31:16]	RW	Emulation data occurs after Tnext. Tnext = NEXT_TIME x Input Clock Period NOTE: You should not set this field to 0x0.	0x1
NEXT_DATA	[15:8]	RW	Next emulation data value	0x0
RSVD	[7:1]	–	Reserved	0x0
EMUL_EN	[0]	RW	Emulation mode enable 0 = Disables 1 = Enables	0

You can use emulation mode to develop software code and debug the operation of TMU. By using emulation mode, software developer can generate the temperature profile as one requires.

When you want to apply the same time delay as sensing time, that is, 938 μs, then you should set NEXT_TIME field to 0x57F0.

Figure 57-4 illustrates the example of emulation mode operation.

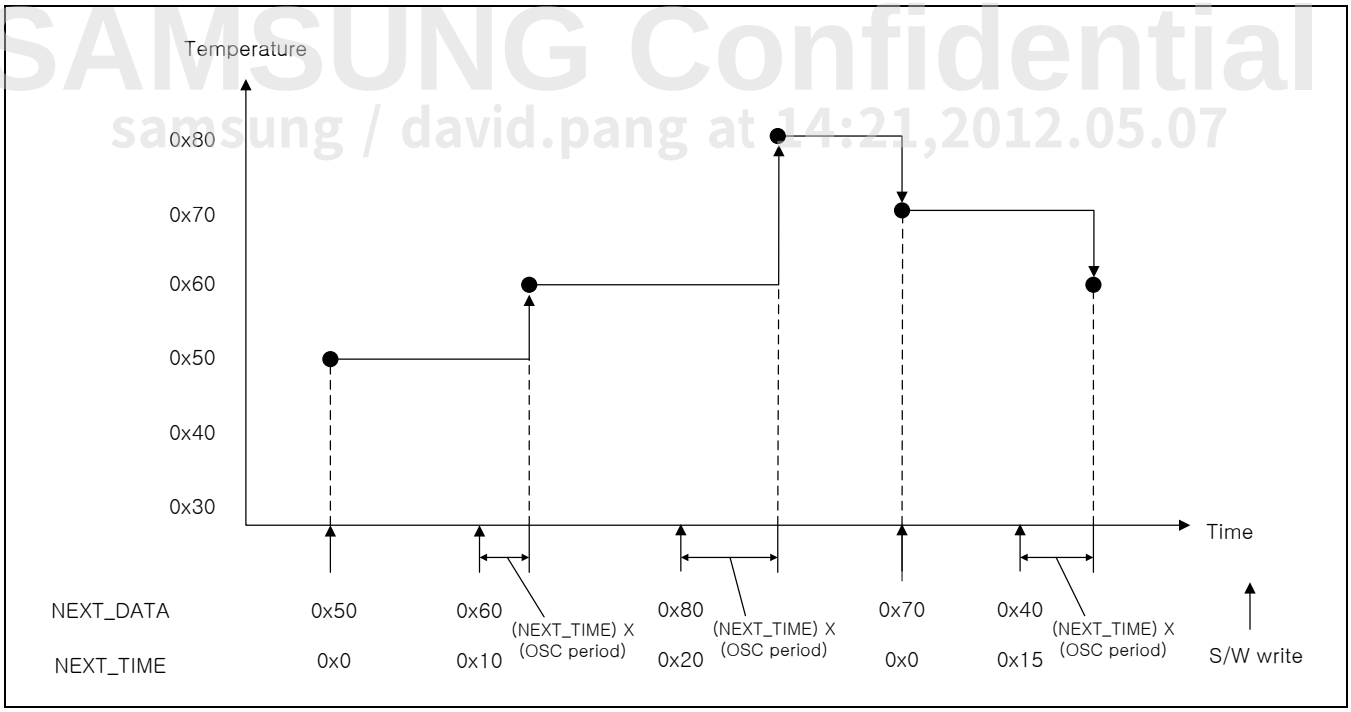


Figure 57-4 Temperature Profile by Using Emulation Mode

When software writes new emulated data (NEXT_DATA[15:8]), it updates this data after (NEXT_TIME) X (OSC period) time delay. It keeps current temperature until it writes new emulated data.

57.6 Programming Guide

This section includes:

- Calibration Equation
- Software Sequence
- Interrupt Service Routine
- Tracing Past Temperature

57.6.1 Calibration Equation

For calibration of temperature sensor, it uses 1-point trimming. 1-point trimming uses the measured data at 25 °C, and calibrates the offset between the measured temperature and 25 °C.

The calibration consists of reading the measured data from e-fuse and writing the calibrated threshold temperature to generate interrupts into trigger level 0-3 register by using calibration equation. The calibration equation for 1-point trimming is as follows:

Tnew = Torg + (TE – 25)

where,
Tnew: calibrated threshold temperature rewritten to TRIG_LEVEL0-3 register.
Torg: original threshold temperature
TE: TRIMINFO_25 of TRIMINFO @0x100C_0000

For example, we want to set threshold temperature to 50 (℃) (= Torg), and TE = 35 (℃), then Tnew is:

Tnew = 50 + (35 – 25) = 60 (℃)
This value is rewritten to TRIG_LEVEL0 (@0x100C_0050) as follows:
THRES_TEMP = 0x55 (60℃)

57.6.2 Software Sequence

The parameter may be changed according to an application.

The example of software sequence is as follows:

```
/* Read the measured data from e-fuse */
Triminfo_25 = TRIMINFO[7:0]

/* Calibrated threshold temperature is written into THRES_TEMP_RISE and THRES_TEMP_FALL */
/* Refer to 1.6.1 */
THRES_TEMP_RISE0 = 0x40;
THRES_TEMP_RISE1 = 0x50;
THRES_TEMP_RISE2 = 0x60;
THRES_TEMP_RISE3 = 0x70;

THRES_TEMP_FALL0 = 0x3A;
THRES_TEMP_FALL1 = 0x4A;
THRES_TEMP_FALL2 = 0x5A;

/* Parameter for sampling interval is set */
SAMPLING_INTERVAL = 0x1;

/* Interrupt enable */
INTEN[24] =0x1;    // for INTEN_FALL2
INTEN[20] =0x1;    // for INTEN_FALL1
INTEN[16] =0x1;    // for INTEN_FALL0
INTEN[8] =0x1;     // for INTEN_RISE2
INTEN[4] =0x1;     // for INTEN_RISE1
INTEN[0] =0x1;     // for INTEN_RISE0

/* Thermal tripping mode selection */
THERM_TRIP_MODE = 0x4;

/* Thermal tripping enable */
THERM_TRIP_EN = 0x1;

/* Check sensing operation is idle */
tmu_idle = 0;
while(tmu_idle&1) {
      tmu_idle = TMU_STATUS[0];
}

/* Start sensing operation */
TMU_CONTROL |= 1;
```

57.6.3 Interrupt Service Routine

```
ISR_INTREQ_TMU () {
/* Read interrupt status register */
int_status = INTSTAT;

if(int_status[24]) {
      ISR_INT_FALL2();
}
else if(int_status[20]) {
      ISR_INT_FALL1();
}
else if(int_status[16]) {
      ISR_INT_FALL0();
}
Else if(int_status[8]) {
      ISR_INT_RISE2();
}
else if(int_status[4]) {
      ISR_INT_RISE1();
}
else if(int_status[0]) {
      ISR_INT_RISE0();
}
else {
      $display("Some error occurred..!");
}

ISR_INT0 () {
      /* Perform proper task for decrease temperature */
      INTCLEAR[0] = 0x1;
}
```

57.6.4 Tracing Past Temperature

To trace the past temperature, read the following registers:

- PAST_TEMP3_0 (@ 0x100C_0060)
- PAST_TEMP7_4 (@ 0x100C_0064)
- PAST_TEMP11_8 (@ 0x100C_0068)
- PAST_TEMP15_12 (@ 0x100C_006C)

58 FIMC-IS

58.1 Exynos 4412 SCP

58.1.1 Overview

Exynos 4412 SCP is an Imaging Sub-system to process image signal from an image sensor. FIMC-IS V1.5 has many sub-blocks for image processing as well as an ARM core. Those sub-blocks for image processing include FIMC-ISP, FIMC-DRC, and FIMC-FD. In addition, it has many peripheral IPs such as I2C, SPI, PWM, UART, and ADC for sensor control.

58.1.2 Features

FIMC-IS V1.5 has the following features:

- A dedicated processor, Cortex™-A5 with Neon, 16 K instruction cache and 16 K data cache.
- GIC, an interrupt controller of cortex-A5.
- PWM timer & Watchdog timer.
- MCUCTL to communicate with main host processor.
- Sensor module control: Multi-PWM, 2 I2Cs, 2 SPIs, ADC, UART, 18 GPIs & 18 GPOs.
- Supported image resolution: 3840 × 2452@15 fps, Full-HD@60 fps.
- Sensor defect compensation, Statistics for 3A.
- Demosaicing
- Denoising
- Dynamic range compression.
- Face detection.

58.1.3 FIMC-IS V1.5 Brief Interface

FIMC-IS V1.5 has an AHB port for main host interface, peripheral ports for sensor control, a pixel input interface port and output interface port, two AXI ports for DMA accesses, a Coresight debugger port and one interrupt port for the main host processor.

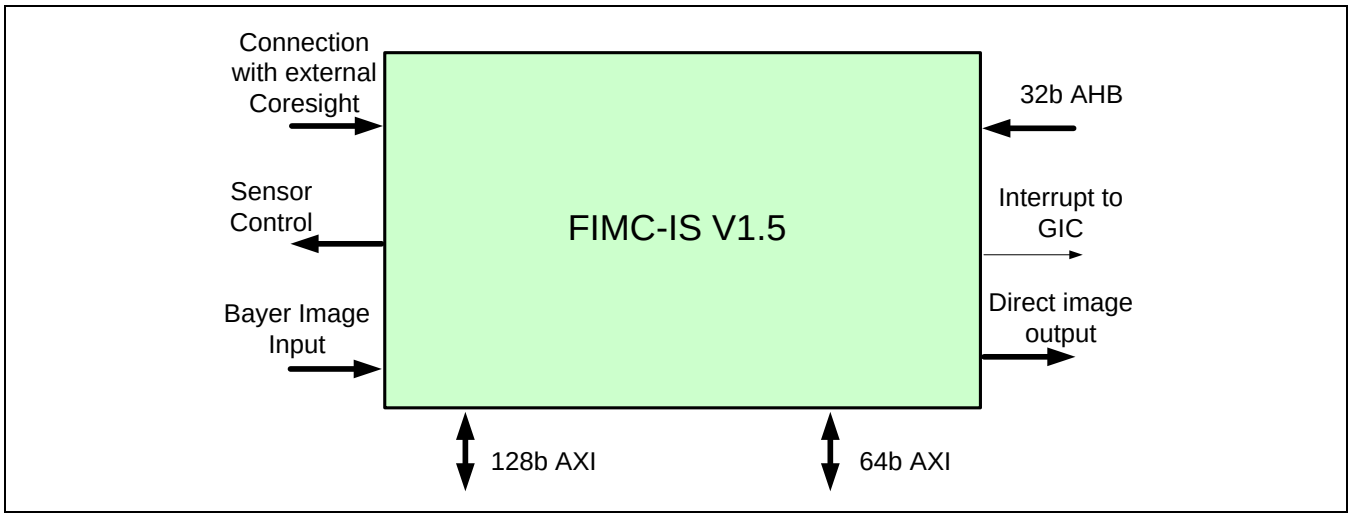


Figure 58-1 Basic Interface of FIMC-IS V1.5

58.1.4 General Description

Detailed architecture of Exynos 4412 SCP is shown in *Figure 58-2* Exynos 4412 SCP has its own processor to control the peripheral devices and ISP chain, and to run image control algorithms such as 3A and face detection. A GIC is an interrupt controller that delivers interrupts from devices to the processor. FIMC-IS V1.5 has a MCUCTL block, which is used for Cortex-A5 to communicate with the main host processor.

The ISP chain consists of three main blocks: FIMC-ISP, FIMC-DRC, and FIMC-FD. FIMC-ISP is main signal processing block for various types of sensor defect compensation, 3A statistics gathering, demosaicing, denoising, and other image enhancements. FIMC-DRC and FIMC-FD perform dynamic range compression and face detection, respectively.

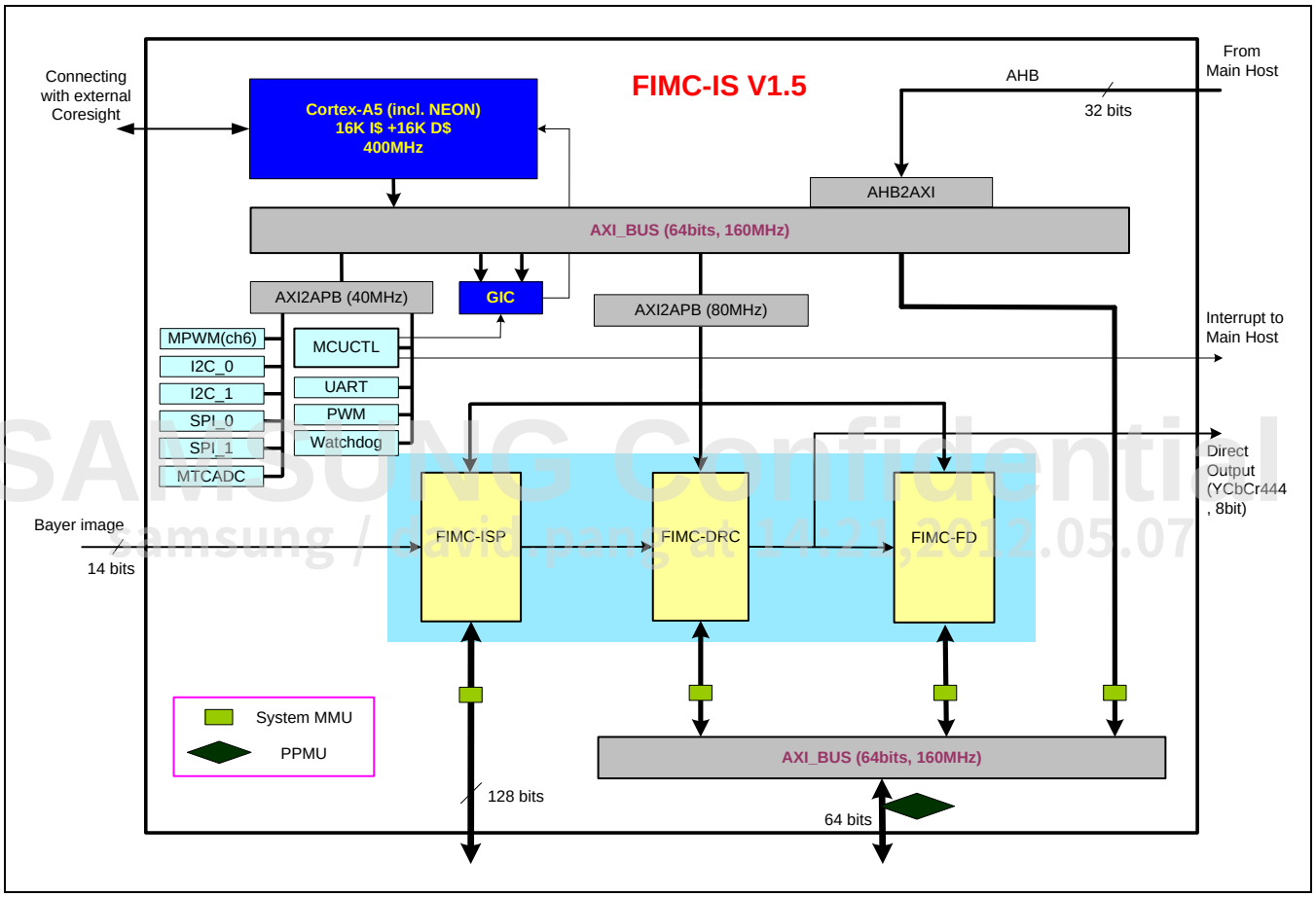


Figure 58-2 FIMC-IS V1.5 Architecture

58.1.4.1 Input and Output Interfaces

Input image can be driven either by on-the-fly interface from the sensor or by DMA input interface from memory. The input image should have one of 8, 10, 12, or 14-bit Bayer format. The Bayer image is transformed into YCbCr 444 or 422 format internally in FIMC-ISP.

Output images can be stored to memory by DMA output interface from FIMC-ISP or (simultaneously) driven to the following FIMC-DRC block by on-the-fly interface. When two outputs are driven simultaneously in FIMC-ISP, the format of the two outputs should be the same. FIMC-DRC's on-the-fly output is the final output image of FIMC-IS V1.5. FIMC-FD's output through DMA is not an image but an internal map data for SW to detect faces from.

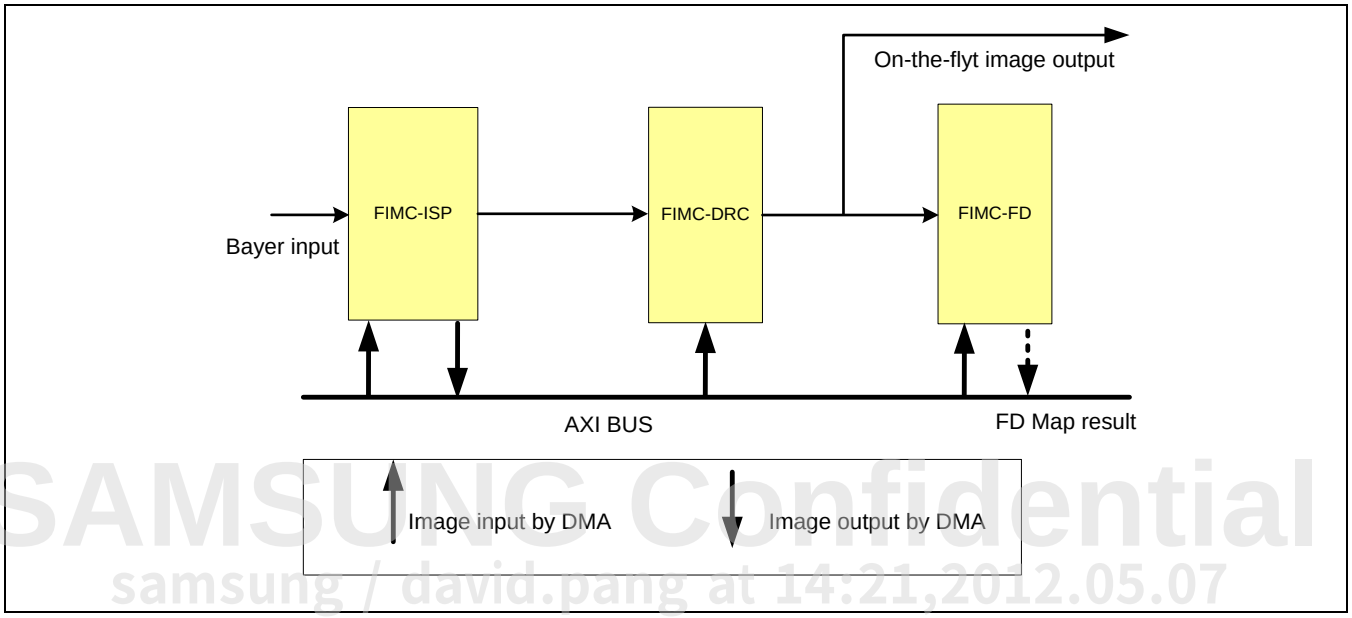


Figure 58-3 ISP Chain

Selectable data-path configurations and I/O data formats of each block are shown in the following tables. Base address of DMA interfaces should be aligned with 16 bytes for FIMC-ISP and 8 bytes for FIMC-DRC and FIMC-FD.

Table 58-1 Selectable Data-path Configurations

	Input		Output	
	On-the-fly	DMA	On-the-fly	DMA
FIMC-ISP	O	O	O	O
FIMC-DRC	O	O	O	X
FIMC-FD	O	O	X	O

Table 58-2 Selectable Data Formats

	Input		Output	
	On-the-fly	DMA	On-the-fly	DMA
FIMC-ISP	Bayer 8/10/12/14 bits	Bayer 8/10/12/14 bits	YCbCr 444/422	YCbCr422, 1/2 plane YCbCr444, 1/3 plane RGB888, 1 plane
FIMC-DRC	YCbCr 444	YCbCr 422, 1 plane YCbCr 444, 1 plane	YCbCr 444	–
FIMC-FD	YCbCr 444	YCbCr 420, 2/3 plane YCbCr 422, 1/2/3 plane YCbCr 444, 2/3 plane	–	Map results

Caution: FIMC-ISP's output format should be same for both On-the-fly and DMA.

Base address of each block included inside FIMC-IS V1.5 is described in the following table.

Table 58-3 Base Address of Blocks in FIMC-IS V1.5

IP	Cortex A-5's View	Main Host's View
FIMC-ISP	0xE000_0000	0x1200_0000
FIMC-DRC	0xE001_0000	0x1201_0000
FIMC-FD	0xE004_0000	0x1204_0000
MPWM	0xE011_0000	0x1211_0000
I2C0	0xE013_0000	0x1213_0000
I2C1	0xE014_0000	0x1214_0000
TCADC	0xE015_0000	0x1215_0000
PWM	0xE016_0000	0x1216_0000
WDT	0xE017_0000	0x1217_0000
MCUCTL	0xE018_0000	0x1218_0000
UART	0xE019_0000	0x1219_0000
SPI0	0xE01A_0000	0x121A_0000
SPI1	0xE01B_0000	0x121B_0000
GIC_C	0xE01E_0000	0x121E_0000
GIC_D	0xE01F_0000	0x121F_0000

There are many BUS components such as SysMMU, PPMU inside FIMC-IS. These components can be accessed only by main host processor. Cortex-A5 cannot access these blocks. The base address of each BUS component is listed in the following.

Table 58-4 Base Address of BUS Components (SysMMU/PPMU)

BUS Components	Main Host's View
SysMMU_FIMC-ISP	0x1226_0000
SysMMU_FIMC-DRC	0x1227_0000
SysMMU_FIMC-FD	0x122A_0000
SysMMU_ISPCPU	0x122B_0000
PPMU_ISPX	0x1236_0000

58.1.4.2 Sync Signal Constraints

For on-the-fly input and/or output, there are sync signal constraints to be met. Each sub-block has its own sync signal constraints. Therefore, all sync signal constraints of all sub-blocks included in the selected sub-block chain should be met together for correct operation. Refer to the following sync signal constraints.

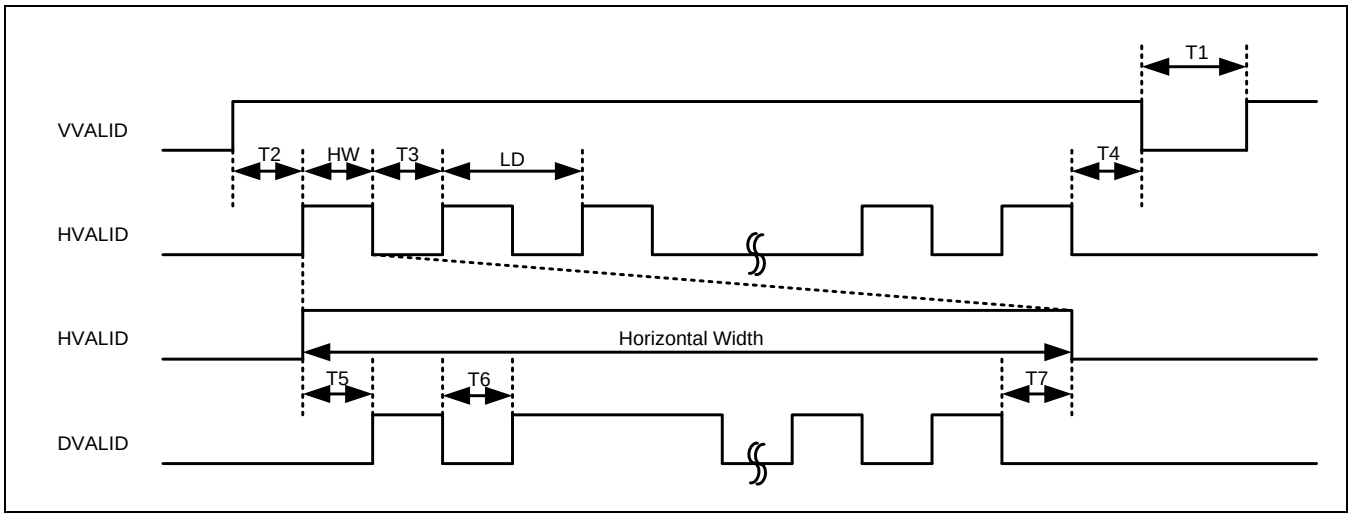


Figure 58-4 Sync Signal Constraints

Table 58-5 Sync Signal Constraints for Each Sub-IP

	Input	Output
(a) FIMC-ISP		
T1	2 clk (** T1 + T4 >= 42 lines + 1500 clk)	Variable depending on configuration
T2	2 clk (** T1 + T4 >= 42 lines + 1500 clk)	1 clk (it will have around tens of line)
T3	1 clk (** T3 + T5 + T7 >= around 40 clk)	1 clk
T4	0 clk (** T1 + T4 >= 42 lines + 1500 clk)	0 clk
T5	0 clk (** T3 + T5 + T7 >= around 40 clk)	1 clk
T6	0 clk	0 clk
T7	0 clk (** T3 + T5 + T7 >= around 40 clk)	0 clk
(b) FIMC-DRC		
T1	T1 ≥ 32,800 clock	T1 ≥ 1 clock
T2	T2 ≥ 0 clock	T2 ≥ 0 clock
T3	T3 ≥ 2 clock	T3 ≥ 60 clock
T4	T4 ≥ 0 clock	T4 ≥ 0 clock
T5	T5 ≥ 0 clock	T5 ≥ 0 clock
T6	T6 ≥ 0 clock	T6 ≥ 0 clock
T7	T7 ≥ 0 clock	T7 ≥ 0 clock

	Input	Output
(c) FIMC-FD		
T1	T1 ≥ 1 clock	–
T2	T2 ≥ 0 clock	–
T3	T3 ≥ 1 clock	–
T4	T4 ≥ 0 clock	–
T5	T5 ≥ 0 clock	–
T6	T6 ≥ 0 clock	–
T7	T7 ≥ 0 clock	–

58.1.5 Programmer's View

The main host processor should first invoke a start command by programming the MCUCTL (0x1218_0000) to turn on the FIMC-IS. MCUCTL has a register to designate the address of the boot code. MCUCTL has interrupt generation registers from main host processor to Cortex-A5, and dedicated interrupt registers to show the status of FIMC-IS.

MCUCTL has 32 bits × 64 shared programmable registers to communicate with the main host processor. That is, the shared programmable registers (0x1218_0080 to 0x1218_017C) can be used as Mail-Box. Additionally, MCUCTL has registers to control 18 GPI and 18 GPO signals for the camera module.

The main host processor should give capture commands to MCUCTL to capture images using FIMC-IS, and Cortex-A5 inside FIMC-IS will program all related registers of each block of the ISP chain according to the given commands. Please refer to the following Figure to understand the program sequences for image capture using FIMC-IS.

The first step for image capture is for the main host processor to give start command to MCUCTL, and then MCUCTL will generate an interrupt for Cortex- A5. The second step is for the Cortex-A5 to interpret commands given by the main host processor. The third step is for Cortex-A5 to set programmable registers of all sub-blocks according to the given scenario. After the completion of the given scenario, Cortex-A5 will notify the completion of the scenario to the main host processor by raising an interrupt. These four steps are depicted in the following Figure.

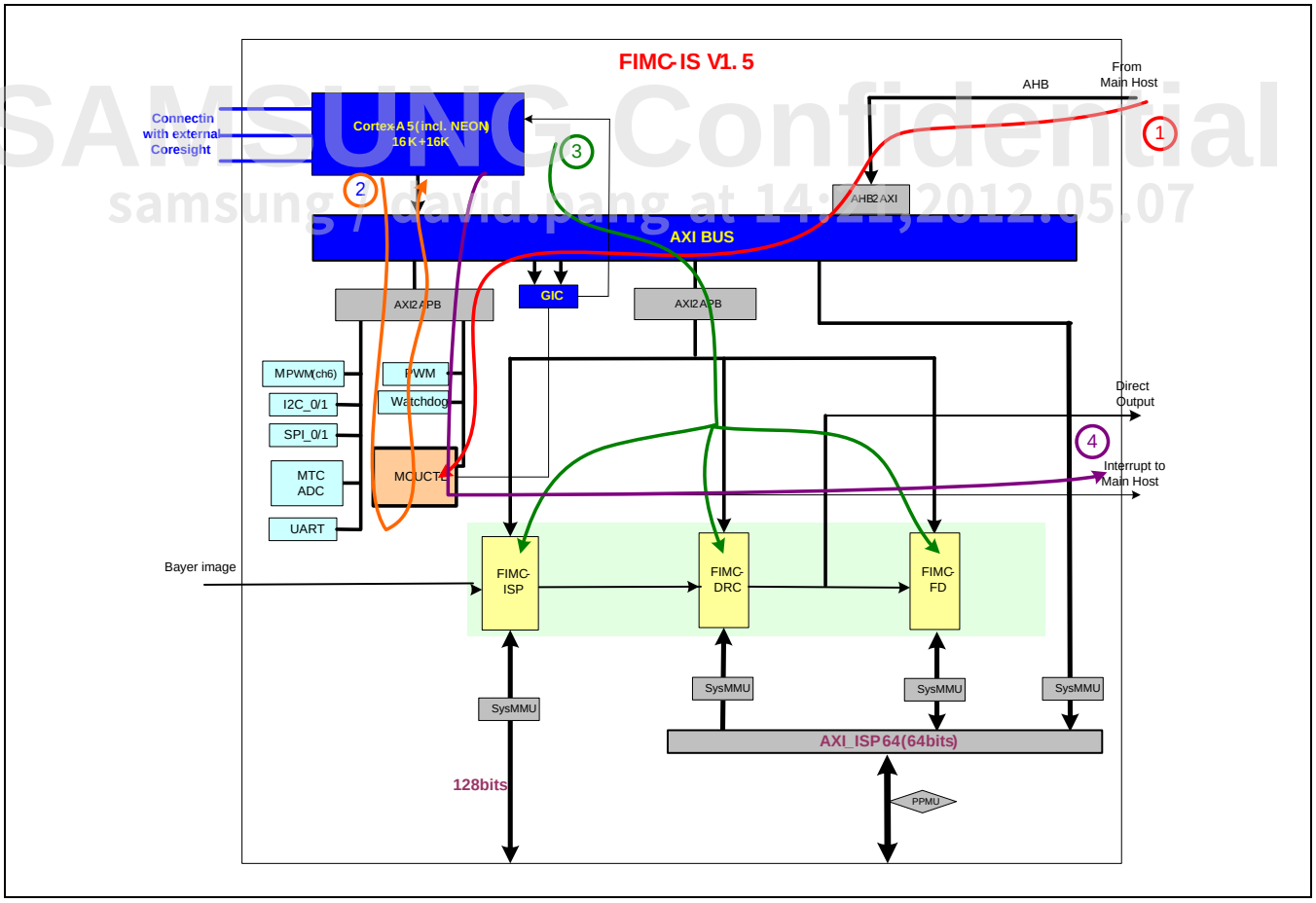


Figure 58-5 Access Sequence for Image Capturing

58.1.5.1 Interrupts

There are two groups of interrupts: One for Cortex-A5, and the other for the main host processor.

58.1.5.1.1 Interrupts for Cortex-A5

The first group of interrupts is for Cortex- A5. These interrupts are divided into two categories depending on the interrupt sources. The first category is interrupts generated by MCUCTL to notify commands from the main host processor. It consists of 6 dedicated and 10 general purpose interrupts. The 10 general purpose interrupts are combined into one interrupt pin, INT_COMB_GEN_VIC. It is recommended to use the six dedicated interrupts for notification of commands such as Wake-up, Power-off, SWreset, CaptureEnable, SingleFrameCapture. The second category is 23 interrupts generated by sub-blocks for Cortex- A5's program control. Therefore, total 30 interrupt pins for the first group are connected to the internal GIC. It is tabularized in the *Table 58-6*.

58.1.5.1.2 Interrupts for the Main Host Processor

The second group of interrupts is for the main host processor. It can also be divided into two categories depending on the interrupt sources. The first category is interrupts generated by MCUCTL to notify command-response to the main host processor. It consists of 10 general purpose interrupts. It is recommended to use the 10 general purpose interrupts for notification of FrameStart, FrameEnd, Overflow, IS_ready, and FD_finished and etc. The second category is interrupts generated by sub-blocks for the main host processor's program control. It consists of 16 interrupts. It is available only for the case in which the main host processor wants to control the sub-blocks directly. For the second group of interrupts, there are only two interrupt pins for the external GIC of the main host: one (INT_COMB_ARMISP_GIC) for 10 general purpose interrupts, the other (INT_COMB_ISP_GIC) for 16 interrupts from the sub-blocks in FIMC-IS.

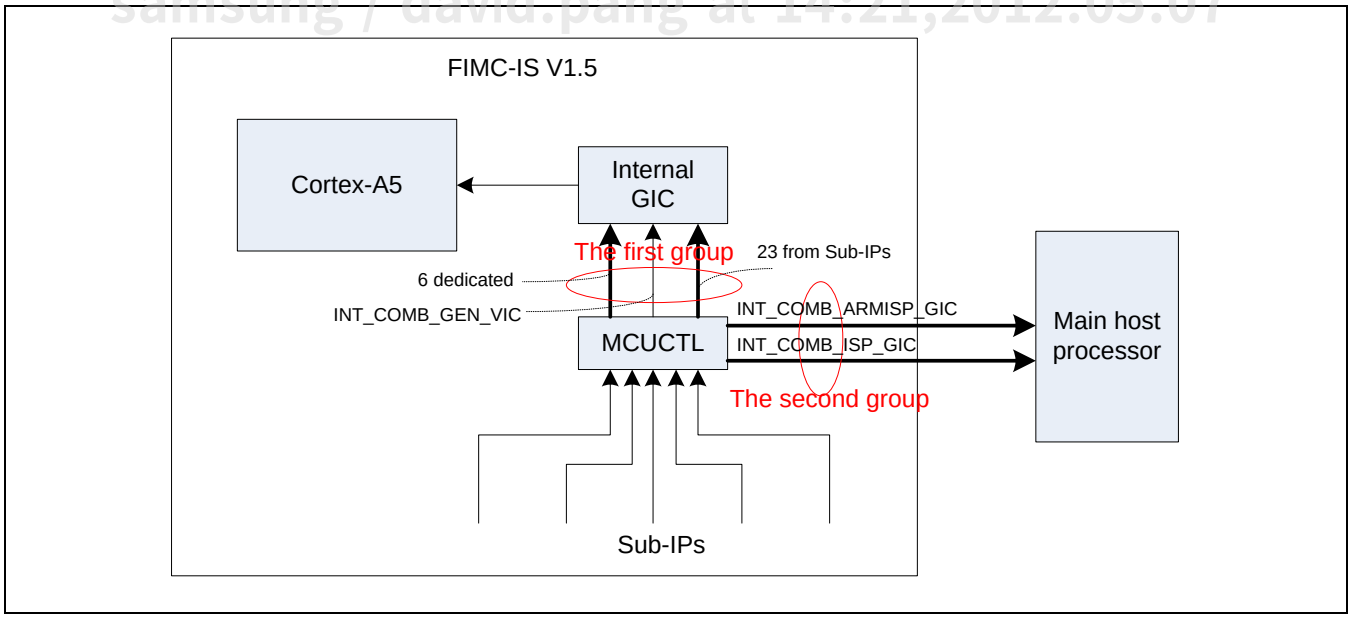


Figure 58-6 Interrupts Group of FIMC-IS

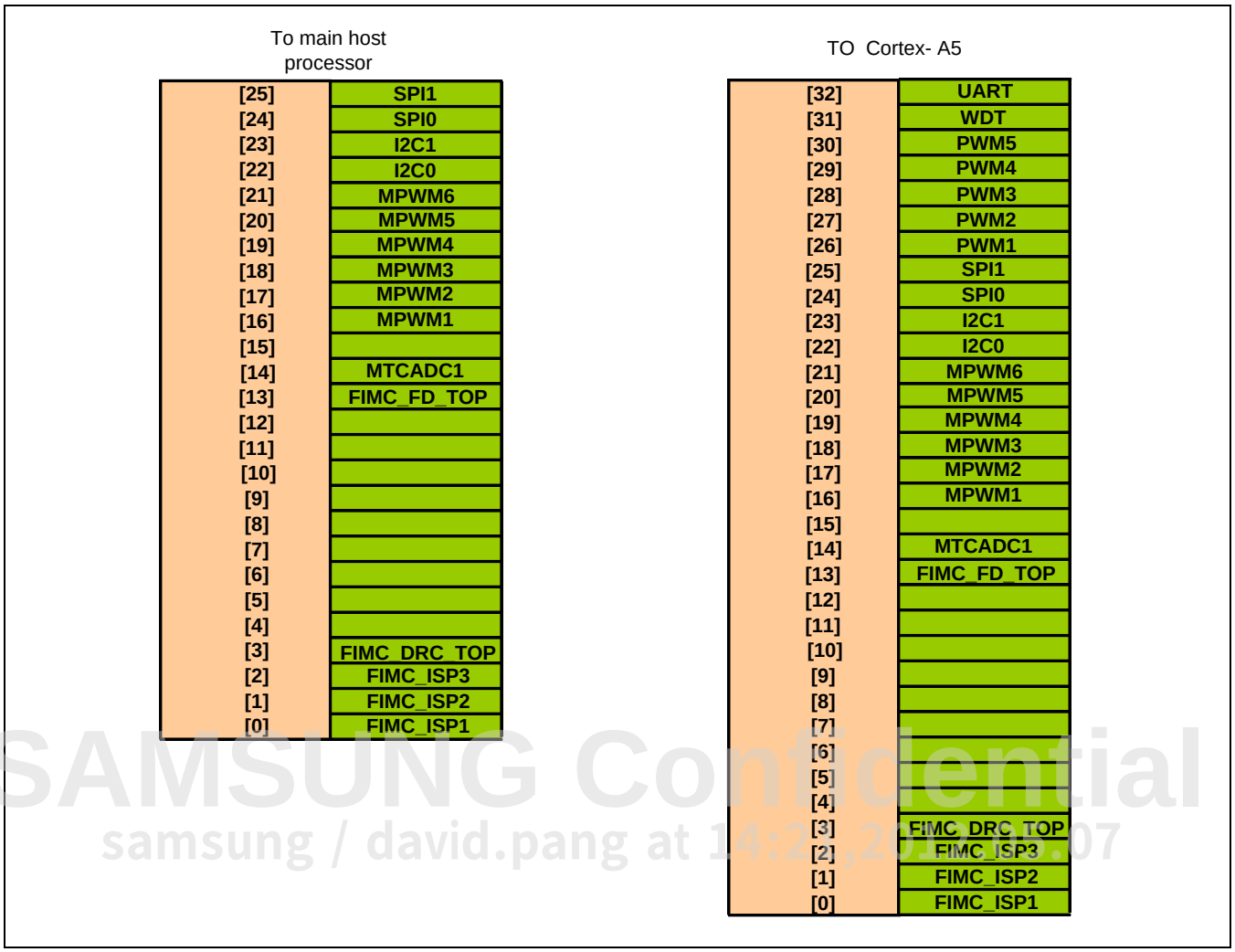


Figure 58-7 Interrupts for Main Host Processor and Cortex-A5

58.1.5.1.3 Interrupt Pin Assignment in the Internal GIC

Interrupts from MCUCTL and Sub-IPs are connected into the internal GIC for Cortex-A5.The number of these interrupts is 30 pins. These 30 interrupt pins are connected into the internal GIC as shown in the following Table.

Table 58-6 Interrupt Pin Assignment in the Internal GIC

GIC's spi[n]	Interrupt Source	GIC's spi[n]	Interrupt Source
0	MCUCTL dedicated1	32	MTCADC1
1	MCUCTL dedicated2	33	0
2	MCUCTL dedicated3	34	MPWM1
3	MCUCTL dedicated4	35	MPWM2
4	MCUCTL dedicated5	36	MPWM3
5	MCUCTL dedicated6	37	MPWM4
6	FIMC_ISP1	38	MPWM5
7	FIMC_ISP2	39	MPWM6
8	FIMC_ISP3	40	I2C0
9	FIMC_DRC_TOP	41	I2C1
10	0	42	SPI0
11	0	43	SPI1
12	0	44	PWM1
13	0	45	PWM2
14	0	46	PWM3
15	0	47	PWM4
16	0	48	PWM5
17	0	49	WDT
18	0	50	UART
19	FIMC_FD_TOP	51	0
20	INT_COMB_GEN_VIC	52	0
21	0	53	0
22	0	54	0
23	0	55	0
24	0	56	0
25	0	57	0
26	0	58	0
27	0	59	0
28	0	60	0
29	0	61	0
30	0	62	0
31	0	63	0

58.2 FIMC-ISP

58.2.1 Overview

FIMC-ISP receives pixel data from sensor or memory, and provides direct pixel output to next IP on-the-fly and DMA pixel output to external memory (DMA). FIMC-ISP is capable of handling image preview and image capture such as video-image or still-image.

58.2.2 Features

Input Format	8, 10, 12, 14-bit Bayer
Output Format	YCbCr 4:4:4, YCbCr 4:2:2, RGB888
Supported Functions	**Bayer domain:** Digital test pattern generation, Bayer Gamma, Bayer crop, Generic offset surface, Periodic mismatch correction, Anti-shading, Thumbnail statistics (for 3A), RGBY histogram statistics, Bayer gains and offsets, Despeckle correction, Disparity correction, Chromatic aberration correction, Bayer downscaler, Denoising, AF statistics, Demosaicing **RGB domain:** Sharpening, Stain killer, Color correction matrix, Skin color detection, RGB Gamma, RGB to YUV conversion **YUV domain:** YUV444 to RGB888 conversion, YUV444 to YUV422 conversion

58.2.3 General Description

58.2.3.1 Input

FIMC-ISP can select input of pixel data between direct input and DMA input data in 8, 10, 12, or 14-bit Bayer. Direct input data is real-time stream data from sensor. DMA input data is stored in memory as 2D with definition of width and height. Data read from memory can be swapped and transformed into 14-bit Bayer format.

58.2.3.2 Output

FIMC-ISP can generate pixel output to the next block on- the-fly and/or DMA pixel output to external memory. Output format can be 12-bit RGB/YUV 4:4:4/4:2:2. DMA output data can be written in 1/2/3 planes depending on scenarios as in *Table 58-2*.

58.2.4 Functional Description

58.2.4.1 Digital Pattern Generation

This block generates various test patterns for testing and debugging of the ISP. This block can generate patterns such as programmable solid color, eight color bars, 16 fade to grey's color bars, Macbeth chart, and pseudo-random PN9/PN12 sequence.

58.2.4.2 Bayer Crop

Bayer Crop block changes the image size by means of arbitrary cropping of pixels in the periphery of an image.

58.2.4.3 Bayer Gamma

The purpose of the Bayer Gamma block is to apply some kind of a function (typically nonlinear function) on the input Bayer pixel stream. Gamma block implements piecewise-linear interpolation of LUT with 32 points distributed on non-equal intervals. Position of points is also programmable but the distances between positions are constrained to be powers of 2.

58.2.4.4 Grid Offset Surface

The purpose of this block is to compensate for non-uniform offset present in the input image signal. The 4x5 grid is used for offset computation through bilinear interpolation.

58.2.4.5 Periodic Bayer Mismatch Correction

This block implements linear correction of Bayer sensor signal over 4x2 periodic matrix produced by pixel design as shown in the following Figure. Each cell in the Figure contains offset and gain.

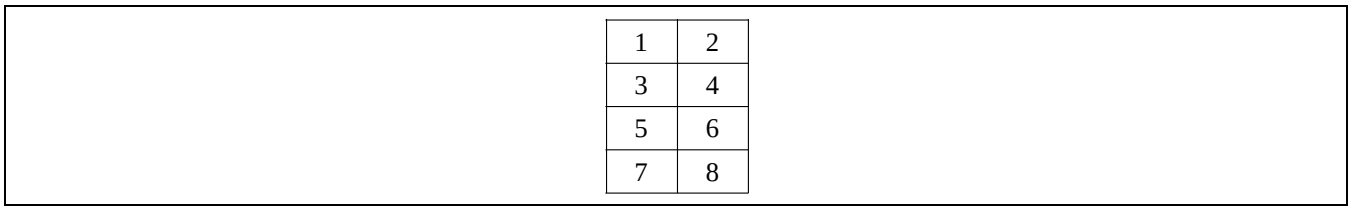


Figure 58-8 MSM Pattern

58.2.4.6 Anti-shading

This block compensates for lens shading (vignetting). A combination of programmable 13x11 grid points and bi-quadratic surface polynomial for each color compoment allows for generation of smooth shading profiles.

58.2.4.7 Thumbnail Statistics

This block produces various types of statistics for the input image. The output statistics is used by AWB, AE, Auto Flicekr cancellation, and Auto-Flash algorithms.

58.2.4.8 RGBY Histogram

The purpose of RGBYHIST block is to collect R, G and B Bayer signal histograms as well as brightness histogram. A user can select the number of bins among 32, 64, 128, and 256 bins.

58.2.4.9 White Balance Gains

The purpose of WBG block is to apply white balance gains, digital gain and, if necessary, signal offsets and programmable limiting.

58.2.4.10 Despeckle Correction

The purpose of Despeckle block is to replace speckle pixels dynamically and statically (memory based).

The block can correct one-pixel speckle as well as clustered speckles. 2x2 clustered speckles are treated specially. The larger the cluster the less robust correction will be. Also the more speckle fall on the same color in the region, the less robust correction can be achieved.

58.2.4.11 Disparity Correction

The purpose of Disparity block is to correct disparity between two green channels.

Disparity appears when the two types of greens (those in even rows and those in odd rows) respond differently to light. This is caused by crosstalk, and the effect is that "checkers" are visible on the demosaiced image. The compensating algorithm is applied to ensure that the average value of the "odd" and "even" greens is the same.

58.2.4.12 Color Aberration Correction

The purpose of the block is to eliminate the chromatic aberration artifacts by scaling R/G/B color channels so that they align with each other.

Due to different refraction index for different wavelengths red, blue and green components of a light ray are mapped to different places on the focal plane. When a Bayer sensor is placed in the focal plane, this translates to the scaling of color channels relatively to each other. This results in color artifacts, especially in places with abrupt color transitions (e.g. building outlines, trees branches on a bright sky background).

58.2.4.13 Bayer Downscaler

The purpose of the block is to resize Bayer images. This block supports downscale ratio of up to 1/16 and minimal output image size is 512 × 512 pixels. The accuracy of scale factor is 5-bit integer and 8-bit fraction.

58.2.4.14 AF Statistics

AF block calculates statistics for auto-focus algorithm. The statistics can be calculated from pixels before Bayer downscaler, denoising or demosaicing block. Connection to demosaicing block can be useful for the auto-focus algorithm to work on denoised image in low-light conditions, while connection to denoising block can be useful to save power when working on a downscaled image.

58.2.4.15 Demosaicing

Desmosaicing block transforms Bayer input format to output RGB using a window of neighbor pixels.

Interpolated pixels are generated for homogeneous, vertical and horizontal directions separately and they pass weighted mixing in accordance with detected directions. After interpolation, various algorithms are applied to reduce possible artifacts.

58.2.4.16 Denoising and Sharpening

ISP performs noise reduction while enhancing sharpness and keeping fine details over pixel data from Bayer to RGB domain.

Denoising block genenerates a denoised, main image stream by applying a kind of average filter to pixels with similar patterns, effectively increasing SNR of the pixel.

In addition to main image stream outputs, the block outputs noise signal that is a difference between original input signal and denoised signal. The noise signal is processed in a noise filter block. The noise filter block removes checker/pattern artifacts caused by Bayer structure. The filtered noise signal is later added to main image stream to preserve fine details.

Denoising block generates a local contrast enhanced (LCE) stream to improve local contrast (visual depth) of an image, while suppressing halo artifacts. The strength of LCE stream is controlled by contrast. The strength of the signal is reduced in areas where local contrast is already high because there is no need to increase local contrast.

It also has Stain Killer block to reduce large-sized color and luminance stains, which is low-contrast and large-scale noise remnants.

A sharpening block collects sharpening information of an image to produce a sharpening signal that will be added to main image stream.

There is also a skin detection block to give a per-pixel estimate on how close pixel's color to skin tone is. At the end of denoising and sharpening block, sharpening power and noise can be reduced at person′s face and other open skin areas using this information.

All generated pixel streams in the denoising and sharpening block are mixed at the end of the block. The strength of each signal can be adjusted according to luminance and skin color values and signals are merged with the main image stream.

58.2.4.17 Color Correction Matrix (CCM)

The purpose of this block is to balance and correct colors by means of minimizing color error on the given target color. The block divides the gamut space and applies different transformation to each divided space. It can reproduce more accurate color compared to conventional CCM. This block supports also gamut compression to preserve the original hue color when CCM outputs are out of gamut. In addition, the block keeps input unchanged not to emphasize distortion of chromaticity when the input is saturated.

58.2.4.18 RGB Gamma

It has Gamma tables for each of R, G, B channels. Gamma block implements piecewise-linear interpolation of LUT with 32 points distributed on non-equal intervals. Position of points is also programmable but the distances between positions are constrained to be powers of 2.

58.2.4.19 Output Formatter

The block converts an image from RGB color space to YUV color space after Gamma correction. The block can later convert an image from YUV color space to RGB color space, or YUV444 to YUV422 format if programmed by user. On-the-fly output and memory output can be enabled simultaneously, in which case their format should be identical.

58.3 FIMC-DRC

58.3.1 Overview

FIMC-DRC is a general-purpose dynamic range compression (DRC) engine resembling the human visual system (HVS). It applies an adaptive local tone-mapping algorithm to an input image or video frame. Based on a statistical analysis of the entire image, the algorithm calculates a gain value for each color component of each pixel. Statistical analysis is based on the luminance component of each pixel, which is separated from chrominance via a color space transform derived from a human visual system model. Algorithm parameters govern the strength of dynamic range compression, weighting between shadows and highlights and other image quality attributes. Additional modules are incorporated in the core to provide for non-linear color correction of enhanced regions and preservation of local contrast.

58.3.2 Features

- General purpose dynamic range compression
- Support an adaptive, local tone-mapping
- Support non-linear color correction
- Support 12 bit YCbCr 444 on-the fly input (10 and 8 bit also supported by using internal shifter)
- Support 12 bit YCbCr 444 output (10 and 8 bit also supported by using internal shifter, rounding and dither)
- Support 12 bit YCbCr 444/422 DMA input (interleaved) (from 8 bit to 12 bit input supported by using internal shifter)

58.3.3 General Description

58.3.3.1 Input

FIMC-DRC can select input either from on-the-fly interface or memory by DMA interface. Minimum size of input data resolution is 320x240. When FIMC-DRC is passed through, minimum can be down to 128x64.

FIMC-DRC has inter-frame dependency; statistics are collected in a frame and used to process the next frame. Correct statistics can be generated only in images larger than or equal to 320x240. Hence, if user wants to switch from pass-through mode with image smaller than 320x240 to operational mode, the first frame after switching will not be processed correctly.

On-the-fly input format to FIMC-DRC is 12-bit YCbCr 4:4:4 sent from FIMC-ISP or the previous IP.

Iput data from memory via DMA interface can be processed and it supports single plane in YCbCr 4:4:4/4:2:2 interlace data format. Depending on scenarios, 4:2:2 format can be either YYCbCr or YCbYCr format. 8-bit input data as 1 byte format or 9-, 10-, 11-, or 12-bit input data as 2 bytes format are possible. DMA input data is converted into a 12-bit data format and sent to the DRC core.

58.3.3.2 Output

Only on-the-fly output interface is available. No DMA output is available. Users can apply dithering, rounding and right-shift operations to the MSB-aligned 12-bit data from the DRC core. MSB 8 bits are connected to the on-the-fly output interface.

58.3.4 Functional Description

58.3.4.1 DRC Strength

This parameter allows to set processing strength of DRC. Minimum value is 0, maximum is 255.

58.3.4.2 Amplification Limits and Slope Restriction

The parameters such as "slope restriction", "dark amplification limit", and "bright amplification limit" are used to restrict the possible range of tone-curves which FIMC-DRC can adaptively generate for each pixel.

When these parameters are turned off, the FIMC-DRC can generate curves with very strong positive gain in very dark areas, and strong negative gain in very bright areas. This response is similar to the human visual system. However, such enhancement can boost noise in the source image and make some images look unnatural.

58.3.4.3 Variance Control

Variance control value affects the sensitivity of the transformation to different areas of the image and can be increased in order to emphasize small area (e.g., faces). If this parameter is set to zero, then the sensitivity to the small areas is maximal and the transform becomes more local. If the parameter is set to 0xFF, then the transformation de-emphasizes small local details and transformation becomes more global (for example, sky at the top of the frame can affect flowers at the bottom of the frame).

58.3.4.4 Black Level and White Level

Black_Level is used as zero level for FIMC-DRC processing in all unsigned data channels. Data below Black_Level will not be processed and stay unchanged.

Similarly, White_Level is used as white level for DRC processing in all unsigned data channels. Data above White_Level will not be processed and stay unchanged.

58.3.4.5 Asymmetry Function

The Asymmetry Function provides with a Lookup Table with 33 words, each of which is 16 bits. The Asymmetry function is used to balance the DRC effect between the dark and bright regions of the image.

58.3.4.6 Color Correction

The Color Correction function provides with a Lookup Table with 33 words, each of which is 16 bits. The Color Correction Lookup table is used to compensate for color shifts when the data is either non- linear or when non-standard gamma is applied after iridix.

58.3.4.7 Dithering Mode

There can be a situation when FIMC-DRC output signal is 12-bit but the signal needs to be compressed into 8-bit. The best way to fulfill this is to use dithering of least significant bits and then truncate these bits.

58.3.4.8 Rounding Schemes

The rounding unit will perform rounding only when processed output bit-shift value is non-zero. It implements four different variants of rounding scheme.

- **floor:** truncation, round towards zero
- **ceil:** round towards maximum
- **round:** round to nearest integer
- **round2mid:** round towards mid-value (e.g., 0x800 for 12-bit case)

58.4 FIMC-FD

58.4.1 Overview

FIMC-FD is to detect and track faces in images and videos. It consists of hardware and software. Hardware generates map results and software uses them to detect and track faces.

58.4.2 Features

- Very good general-purpose detection in extreme illumination conditions and face poses.
- No limitations on the number of faces - can detect and track any number of faces in a wide range of poses. Scene dependent parameters are collected and used to improve accuracy.
- Implements advanced tracking algorithms, thus not reliant only on frontal face detection. Once the face is "locked", momentary pose changes do not lose the face, being able to track the pose changes to very extreme.
- When an orientation sensor is not available in the camera, the face tracking can provide face orientation information (by default: 0, 30, 45, 60, 90, – 30, – 45, – 60 and – 90 degrees).
- Self calibrating to difficult lighting conditions, providing a real help to improve the quality of the image and movies (back-light illumination conditions, under/over exposure, low lighting conditions, etc.).
- Uses full color information which allows for a very low false positive ratio, but it can also work on luminance only information.
- Smile and blink detection.

58.4.3 General Description

58.4.3.1 Input

FIMC-FD can select input source either from on-the-fly interface or memory by DMA interface. For on-the-fly input, MSB 8-bit data of FIMC-DRC on-the-fly output is used. For DMA input, user can select a format among 4:4:4, 4:2:2, and 4:2:0.

58.4.3.2 Output

Only DMA output is available, which is map data for software to detect faces.

58.4.4 Functional Description

58.4.4.1 Detection and Tracking Angles

The initial locking of the face can occur only under some orientation restrictions:

- ROP (Rotation Out of Plane)
  - left-right (yaw): up to 90 degrees
  - up-down (pitch): up to 40 degrees
- RIP (Rotation In Plane) left-right (roll): FD can support the following sets of basic RIP angles. Each particular basic angle can be enabled or disabled to achieve a desired performance trade-off. FD can detect the faces at the specified below deviation from a basic angle:
  - Base angles: 0°, ± 90°, 180°. Up to ± 20° deviation.
  - Base angles: ± 45°, ± 135°. Up to 10° deviation.
  - Base angles: ± 30°, ± 60°, ± 120°, ± 150°. Up to 10° deviation.

After a face is locked, it can be followed by the tracking beyond the above angles.

58.5 MCUCTL

58.5.1 Overview

The MCUCTL unit in FIMC-IS V1.5 is a slave on the ARM APB bus. Its function is to hold the programmable registers that control the ARM sub-system configurations as well as the initial condition of the ARM core. It also has shared registers to communicate with the main host processor and help booting the Cortex-A5 core. Furthermore, it supports registers to control 18 GPIO signals for camera module.

58.5.2 Features

- Supports shared registers with 64x32 words.
- Supports H/W interrupt generation and status to VIC from the main host processor.
- Supports H/W interrupt generation and status to GIC from the Cortex-A5 core.
- Supports 18 GPIO signals to control camera module.
- Controls remap signal and booting initial address for test debugging.

58.5.3 Functional Description

The MCUCTL supports main control signal and interrupts to ISP ARM sub-systems. It shall help start booting sequences for Cortex-A5. FIMC-IS does not have an internal memory. Therefore, the core should be booted from an external memory. This block has APB interface to set its registers.

The MCU control unit provides shared registers with 64x32 bits to share information between the main host processor and the Cortex-A5 in FIMC-IS. Both CPUs can access them. In addition, it supports S/W interrupts to inform each CPU of their current status through its registers. For that, this block provides two kinds of interrupts as direct interrupts and combined interrupt. The direct interrupt registers support 6 interrupts that can be generated by setting 6 bits of directed interrupt register. These interrupts is directly connected to 6 interrupts of VIC within ISP ARM sub-systems respectively. There are status registers and masking registers for corresponding interrupt direct registers. The combined interrupt registers support 10 interrupts that can be generated by setting 10 bits of combined interrupt register.

To control the external camera module, it also provides with 18 GPI (General Purpose Input) ports and 18 GPO (General Purpose Output) ports. 18 GPOEN (General Purpose Enable) registers should be set to enable output port of pads of corresponding 18 GPO ports.

58.5.3.1 Block Diagram

Figure 58-9 shows the block diagram MCU Control unit and connected modules.

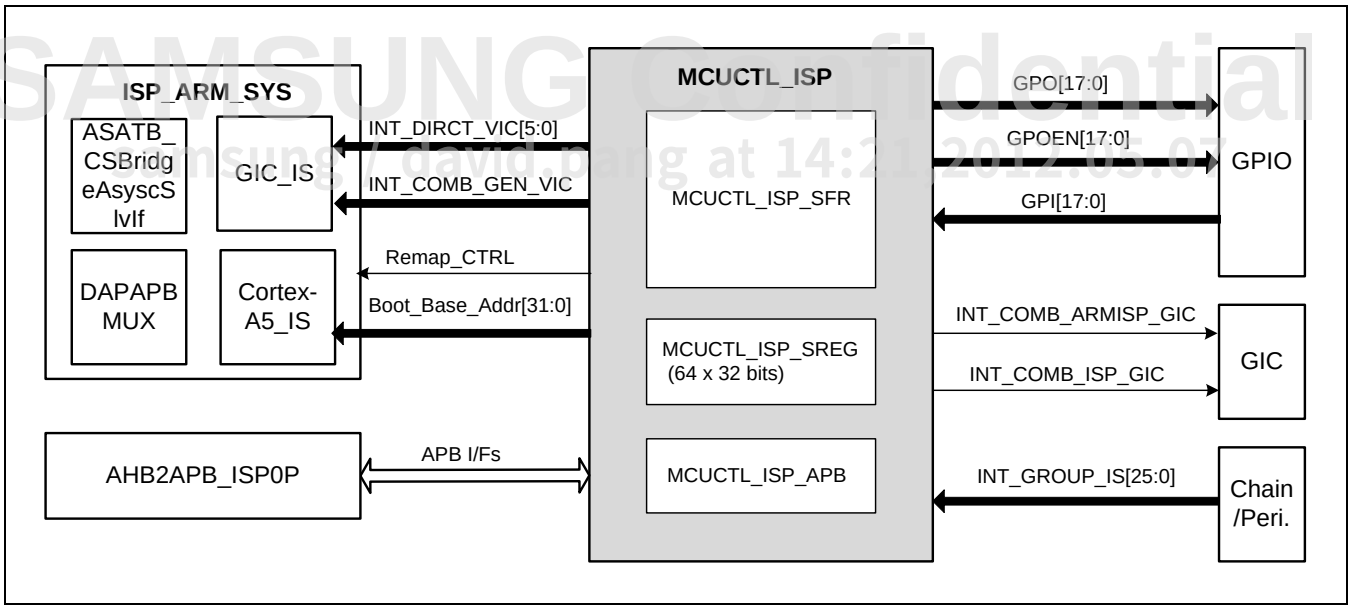


Figure 58-9 Block Diagram of MCU Controller and Interfaces

58.6 MPWM

58.6.1 Overview

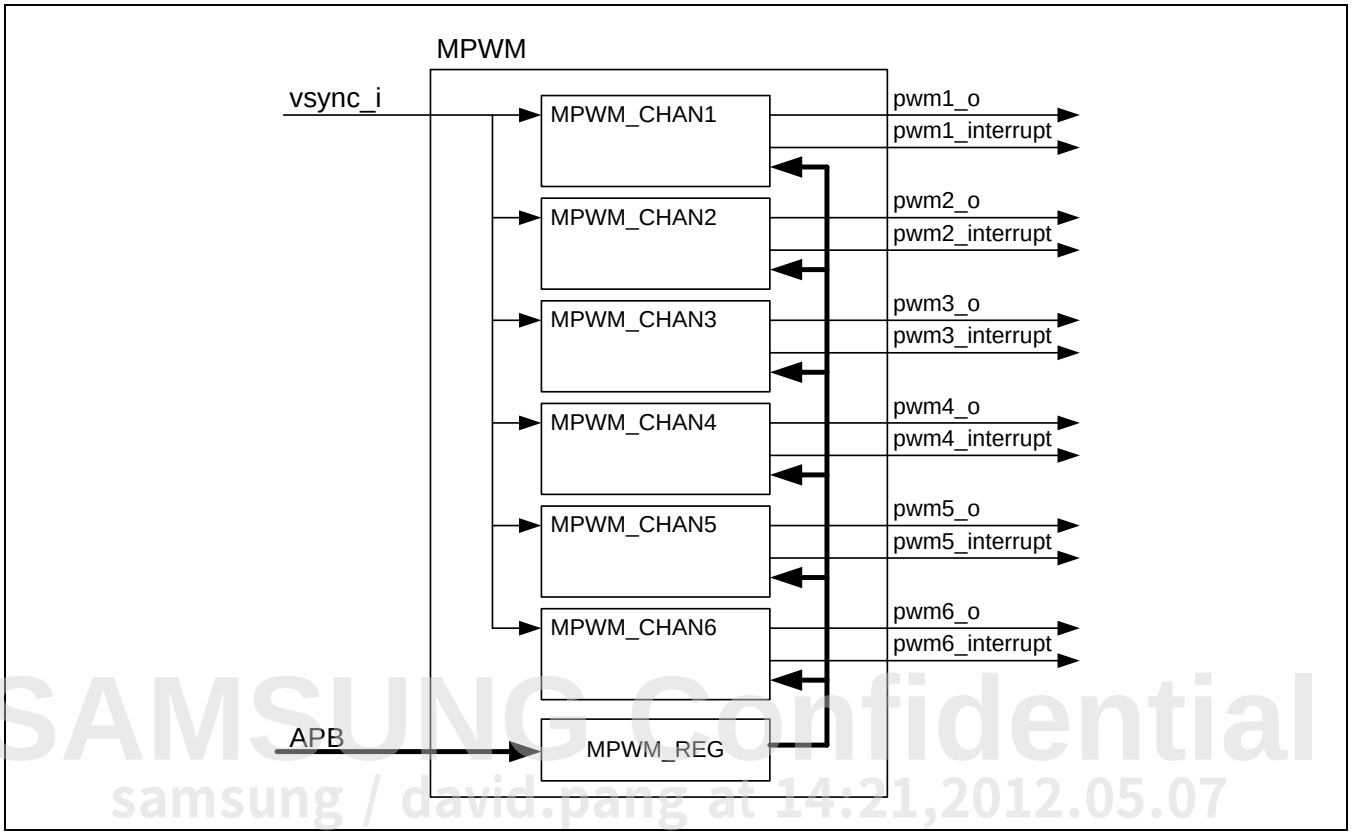


Figure 58-10 Block Diagram of MPWM

Figure 58-10 shows a block diagram of MPWM. The MPWM consists of 6 MWPM_CHAN modules and 1 MPWM_REG module. Each MPWM_CHAN makes multi-PWM pulse according to register setting. Each module can make a pulse which has 5 different widths and this pulse can be generated multiple times by setting of register.

MPWM_REG module has registers which can be set by APB bus. These register is used to set each MPWM_CHAN module's behavior.

58.6.2 Features

Features of MPWM are as follows.

- Five period parameters and counter setting (PWMx_P, PWMx_T1, PWMx_T2, PWMx_T3, PWMx_T4)
- Each modulation period should be (PWMx_T1 < PWMx_T2 < PWMx_T3 < PWMx_T4)
- 32 MHz, 31.25 ns accuracy 16 bit data
- Enable control by PWM_ENABLE
- Only if PWM_ENALBE is set to HIGH, PWM pulse can be started
- If PWM_ENABLE register set to LOW during operation, all counters will become reset and PWM output will be LOW and interrupt
- Number of PWM pulse can be set by PWMx_P_NUM
- Total cycle of PWM pulse become PWMx_P (NOTE) PWMx_P_NUM
- Initial output level can be specified HIGH or LOW by PWM_POLARITY
- Output level is reversed at every PWMx_T (NOTE)
- If PWMx_T (NOTE) is larger than PWMx_P, PWM output after PWMx_P will be ignored. Only PWM output within PWMx_P duration will happen.

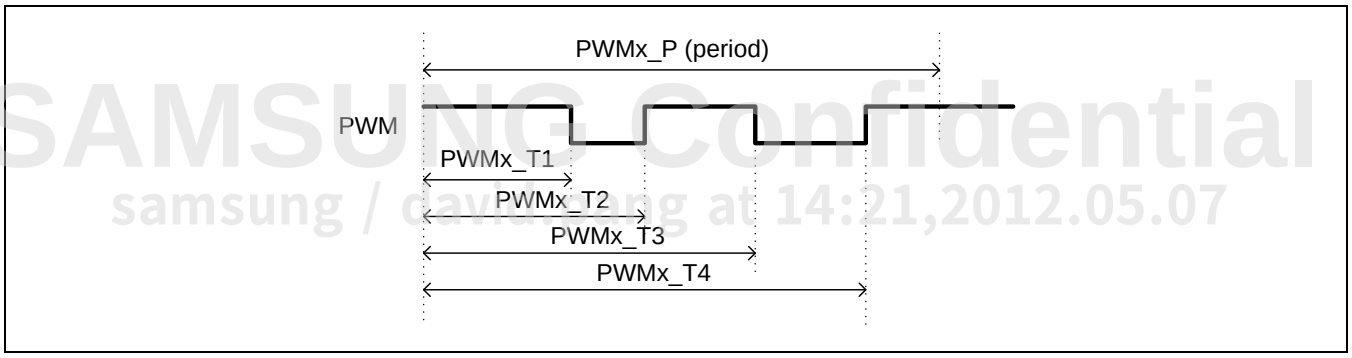


Figure 58-11 PWM Cycle Timing

NOTE: If only one pulse is needed, make sure that (PWMx_T1 < PWMx_T2) and (PWMx_T2 = PWMx_T3 = PWMx_T4 = PWMx_P) as *Figure 58-12*.

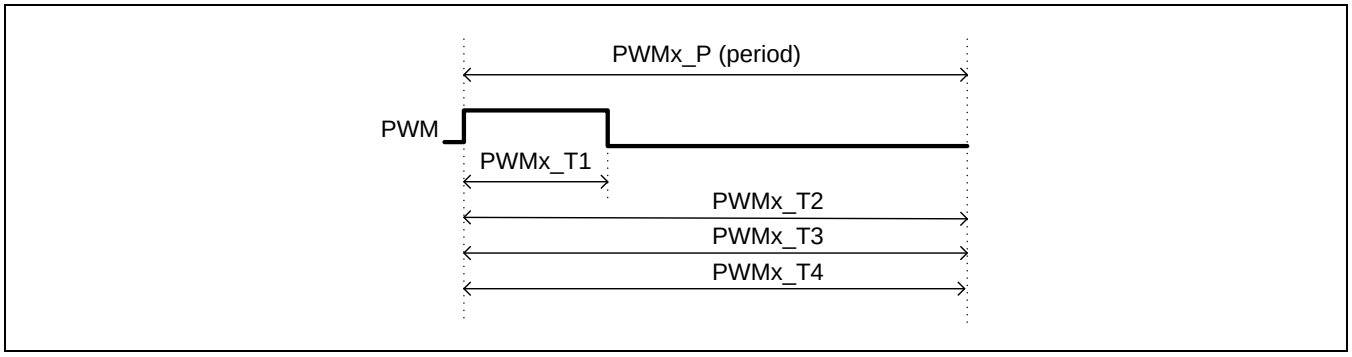


Figure 58-12 Generation of One PWM Pulse

58.6.3 Operation of MPWM

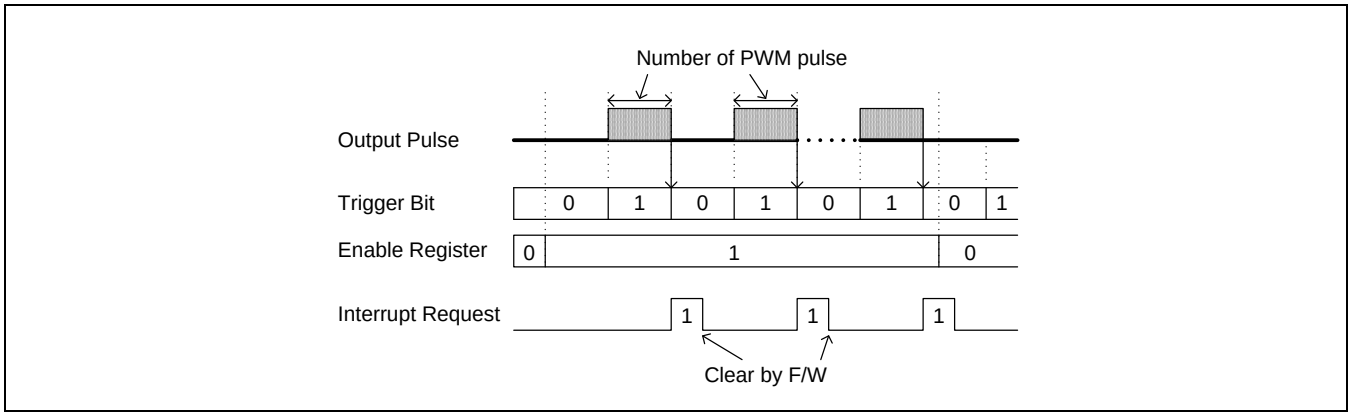


Figure 58-13 Free Trigger Mode

1. PWM pulse will start when trigger bit is set.
2. Generate interrupt request at the end of PWM output.
3. Trigger bit will be cleared at the end of PWM output.
4. Trigger bit is valid while enable register is set as "1".

NOTE: To prevent overlapped interrupt, (PWMx_P × PWMx_P_num) + (APB setting time for trigger) should be bigger than 5 cycle as *Figure 58-14*.

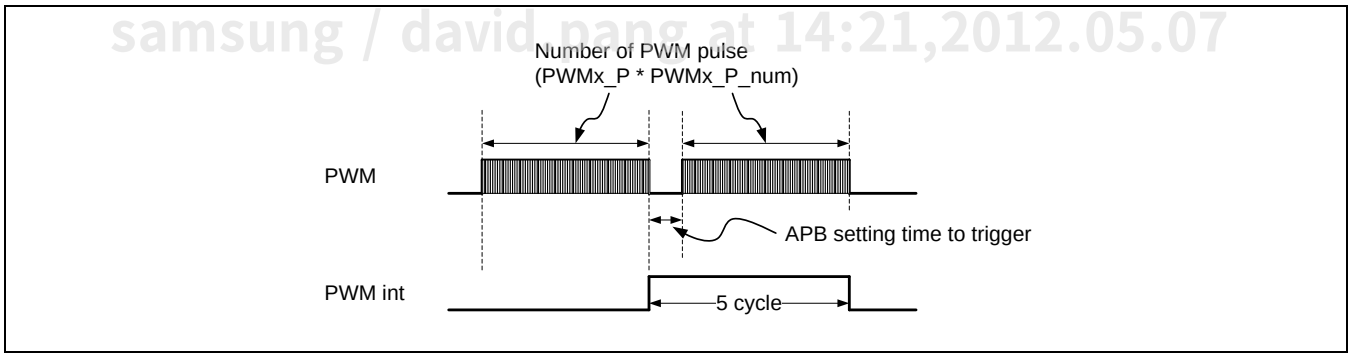


Figure 58-14 In case of Overlapped Interrupt

58.6.4 Interrupt

Each MPWM's interrupt output has a form of pulse. When the PWM reaches to number of PWM pulse, Interrupt will be asserted for 5 cycles. Then it will be de-asserted automatically.

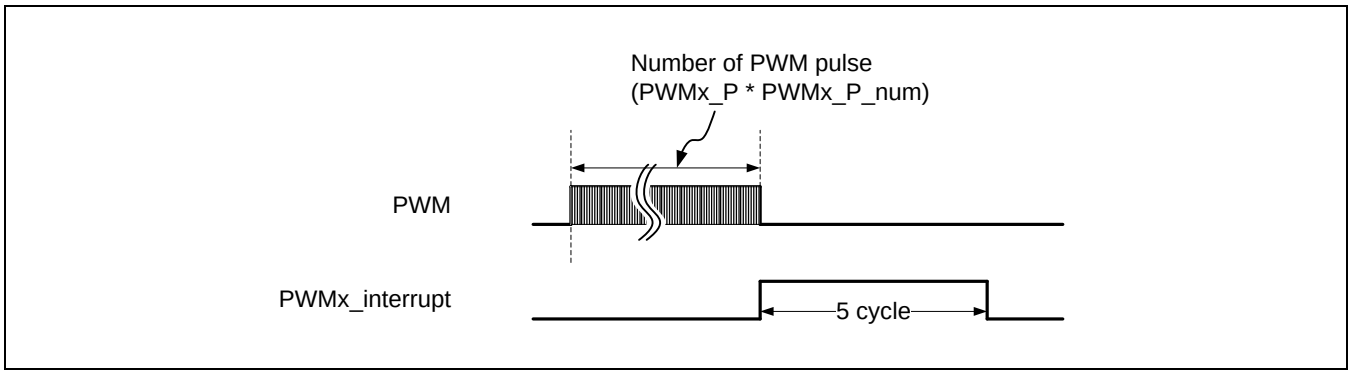


Figure 58-15 The Shapes of PWM Interrupt

58.7 ADC for Motor Control

This chapter describes the functions and usage of ADC for motor control.

58.7.1 Overview of ADC

The 10-bit or 12-bit CMOS Analog to Digital Converter (ADC) comprises of 10-channel analog inputs. It converts the analog input signal into 10-bit or 12-bit binary digital codes at a maximum conversion rate of 1MSPS with 5 MHz A/D converter clock. A/D converter operates with on-chip sample-and-hold function. ADC supports low power mode.

58.7.2 Key Features of ADC for Motor Control

The ADC includes the following features:

- Resolution: 10-bit/12-bit (optional)
- Differential Nonlinearity Error: ± 2.0 LSB (Max.)
- Integral Nonlinearity Error: ± 4.0 LSB (Max.)
- Maximum Conversion Rate: 1 MSPS
- Low Power Consumption
- Power Supply Voltage: 1.8 V (Typ.), 1.0 V (Typ., Digital I/O Interface)
- Analog Input Range: 0 to 1.8 V
- On-chip sample-and-hold function
- Signal to noise & distortion ratio: 54 dB (Min.)
- Operating temperature range: – 40 °C to 85 °C

58.7.3 Block Diagram of ADC

Figure 58-16 is the functional block diagram of A/D converter for motor control.

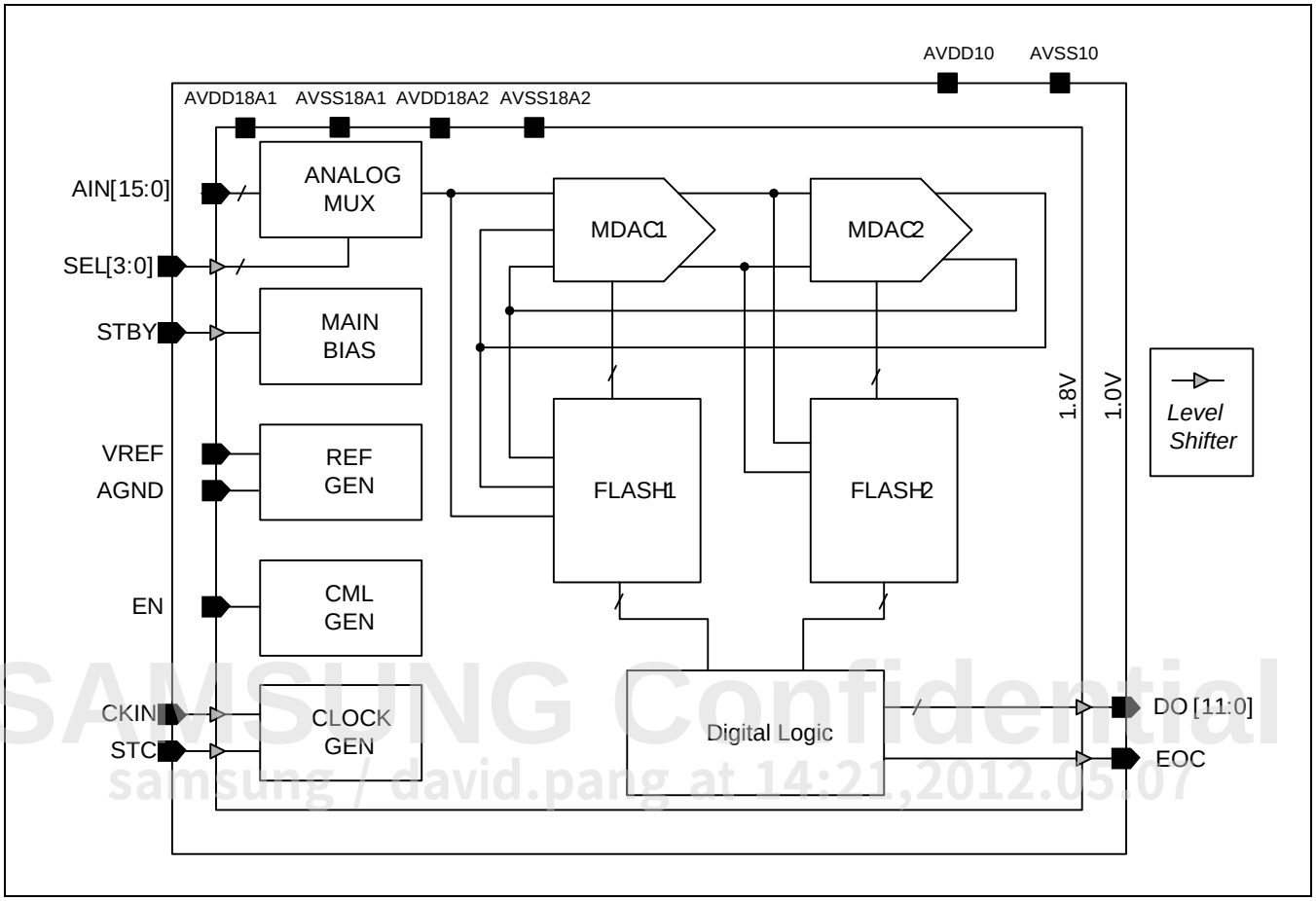


Figure 58-16 ADC Functional Block Diagram

58.7.4 Function Descriptions

58.7.4.1 A/D Conversion Time

When the APB bus clock (PCLK) frequency is 66 MHz and the prescaler value is 65, total 12-bit conversion time is as follows.

- A/D converter freq. = 66 MHz/(65 + 1) = 1 MHz
- Conversion time = 1/(1 MHz/5 cycles) = 1/200 kHz = 5 μs

NOTE: This A/D converter was designed to operate at maximum 5 MHz clock, so the conversion rate can go up to 1 MSPS.

58.7.4.2 ADC Conversion Mode

The operation of this mode is same as AIN0tofAIN3's. To initialize this mode, set the TSADCCON0 (ADC control register). The converted data can be read out from TSDATX0 (ADC conversion data X register).

58.7.4.3 Standby Mode

Standby mode is activated when TSSEL bit is "0" and STANDBY bit is "1" in TSADCCON0 register. In this mode, A/D conversion operation is halted and TSDATXn registers hold their values.

58.7.4.3.1 Programming Notes

1. The A/D converted data can be accessed by means of interrupt or polling method. With interrupt method, the overall conversion time-from A/D converter start to converted data read-may be delayed because of the return time of interrupt service routine and data access time. With polling method, to determine the read time for TSDATXn register, check the TSADCCONn[15]-end of conversion flag-bit.

2. A/D conversion can be activated in different way. After TSADCCONn[1]-A/D conversion start-by-read mode-is set to 1. A/D conversion starts simultaneously when converted data is read.

59 Electrical Data

This chapter describes the Electrical data of Exynos 4412 SCP.

59.1 Absolute Maximum Ratings

Any Stress beyond "Absolute Maximum Ratings" listed in *Table 59-1* can cause permanent damage to the device. These are only stress ratings. You should keep the "Recommended Operation Conditions" for functional operation of the device. Exposure to absolute-maximum rated conditions for extended periods can affect the reliability of the device.

Table 59-1 lists absolute maximum ratings.

Table 59-1 Absolute Maximum Ratings

Parameter	Symbol	Rating			Unit
			Min.	Max.	
DC supply voltage	VDD	1.0 V VDD	– 0.5	1.5	V
		1.8 V VDD	– 0.5	2.5	
		2.5 V VDD	– 0.5	3.6	
		3.3 V VDD	– 0.5	3.8	
DC input voltage	VIN	1.8 V input buffer	– 0.5	2.5	
		2.5 V input buffer	– 0.5	3.6	
		3.3 V input buffer	– 0.5	3.8	
DC output voltage	VOUT	1.8 V output buffer	– 0.5	2.5	
		2.5 V output buffer	– 0.5	3.6	
		3.3 V output buffer	– 0.5	3.8	
In/Out current	I/O	–	± 20		mA
Storage temperature	T_A	–	– 65 to 150		°C

NOTE: Absolute maximum ratings are those values beyond which damage to the device may occur. You should keep the "Recommended Operation Conditions" for functional operation of the device.

59.2 Recommended Operating Conditions

Table 59-2 lists operate Exynos 4412 SCP based on the recommended operating conditions.

Table 59-2 Recommended Operating Conditions (Ta=-25 ~ 85°C and Tj=-25 ~ 125°C, unit = V)

Pin Name	Parameter	Typical	Range	Note
VDD_APLL VDD_MPLL VDD_EPLL VDD_VPLL	DC Supply voltage for core block	1.1	±5%	–
VDD_ALIVE		1.0	±5%	–
VDD_ARM 59-4		1.3 59-4	±5%	For ARM 1.4GHz
VDD_INT 59-4		1.0 59-4	±5%	For DRAM 400MHz
VDD_MIF 59-4		1.0 59-4	±5%	For DRAM 400MHz
VDD_G3D 59-4		1.0 59-4	±5%	For G3D 440MHz
VDDQ_M0	DC supply voltage for memory interface 0 (NOR/NAND/OneNAND)	1.8	±5%	–
VDDQ_M1	DC supply voltage for memory interface 1 (DRAM) (2)	1.2	±5%	–
VDDQ_M2	DC supply voltage for memory interface 2 (DRAM) (2)	1.2	±5%	–
VDDQ_C2C	DC supply voltage for C2C	1.2/1.8	±5%	–
VDDQ_C2C_W	DC supply voltage for C2C wakeup	1.2/1.8	±5%	–
VDDQ_PRE	DC supply voltage for IO predriver	1.8	±5%	–
VDDQ_SYS0	DC supply voltage for SYS0 block (XCLKOUT, XGNSS_MCLK/CLK_REQ/RTO_OUT, XXTI, XOM, XPWRRGTON, XEINT8 to 31 …)	1.8	±5%	–
VDDQ_SYS1	DC supply voltage for SYS1 block (XPKG_MODE, XuotgDRVVBUS, XuhostPWREN, XuhostOVERCUR, JTAG)	1.8	±5%	–
VDDQ_SYS2	DC supply voltage for SYS2 block (XusbXTI, XusbXTO)	1.8	±5%	–
VDDQ_SYS33	DC supply voltage for SYS33 block (XEINT0 to 7)	1.8	±5%	–
VDDQm1_CKE	DC supply voltage for CKE 1	1.2	±5%	–
VDDQm2_CKE	DC supply voltage for CKE 2	1.2	±5%	–
VDDQ_SBUS	DC supply voltage for slim bus	1.2/1.5/1.8	±5%	–
VDDQ_MIPIHSI	DC supply voltage for MIPIHSI	1.2/1.8	±5%	–
VDDQ_GPS0	DC supply voltage for GPS0	1.8	±5%	–

Pin Name	Parameter	Typical	Range	Note
VDDQ_EXT0	DC supply voltage for EXT0	1.8	±5%	–
VDDQ_CKO	DC supply voltage for CKO	1.8	±5%	–
VDD_RTC	DC supply voltage for RTC	1.8	±5%	–
VDDQ_LCD	DC supply voltage for LCD	1.8	±5%	–
VDDQ_ISP	DC supply voltage for ISP	1.8	±5%	–
VDDQ_CAM	DC supply voltage for CAM	1.8	±5%	–
VDDQ_AUD0	DC supply voltage for AUD0	1.8	±5%	–
VDD10_HDMI	DC supply voltage for HDMI TX	1.0	±5%	–
VDD10_HDMI_PLL	DC supply voltage for HDMI PLL	1.0	±5%	–
VDD18_HDMI_OSC	DC supply voltage for HDMI OSC	1.8	±5%	–
VDD18_MIPI	DC supply voltage for MIPI I/O	1.8	±5%	–
VDD18_MIPI	DC supply voltage for MIPI 4L 1.8	1.8	±5%	–
VDD10_MIPI	DC supply voltage for MIPI 4L 1.0	1.0	±5%	–
VDD10_MIPI_PLL	DC supply voltage for MIPI 4L PLL	1.0	±5%	–
VDD18_MIPI	DC supply voltage for MIPI 2L I/O	1.8	±5%	–
VDD18_MIPI2L	DC supply voltage for MIPI 2L 1.8	1.8	±5%	–
VDD10_MIPI2L	DC supply voltage for MIPI 2L 1.0	1.0	±5%	–
VDD33_UOTG	DC supply voltage for USB OTG I/O	3.3/3.0	±3%	–
VDD10_UOTG	DC supply voltage for USB OTG core	1.0	±5%	–
VDD18_HSIC	DC supply voltage for USB HSIC 1.8 V	1.8	±5%	–
VDD12_HSIC0	DC supply voltage for USB HSIC0 IO	1.2	±5%	–
VDD12_HSIC1	DC supply voltage for USB HSIC1 IO	1.2	±5%	–
VDD10_HSIC	DC supply voltage for USB HSIC CORE	1.0	±5%	–
VDDQ_MMC01	DC supply voltage for MMC 0/1	1.8/2.5/3.3	±5%	–
VDDQ_MMC2	DC supply voltage for MMC 2	1.8/2.5/3.3	±5%	–
VDDQ_MMC3	DC supply voltage for MMC 3	1.8	±5%	–
VDD18_ADC	DC supply voltage for ADC	1.8	±5%	–
VDDQ_E1	DC supply voltage for LPDDR2_1_DQ of MCP	1.2	±5%	–
VDD18_TS	DC supply voltage for temp sensor	1.8	±5%	–
VDD18_ABB0	DC supply voltage for ABB0 (Internal)	1.8/1.95	±3%	1.95V for RBB

Pin Name	Parameter	Typical	Range	Note
VDD18_ABB1	DC supply voltage for ABB1 (Core block)	1.8/1.95	±3%	1.95V for RBB
VDD18_ABB2	DC supply voltage for ABB2, 3 (G3D, ARM)	1.8/1.95	±3%	1.95V for RBB
VDDCA_E1	DC supply voltage for LPDDR2_1 input of MCP	1.2	±5%	–
VDDQ_E2	DC supply voltage for LPDDR2_2_DQ of MCP	1.2	±5%	–
VDDCA_E2	DC supply voltage for LPDDR2_2 input of MCP	1.2	±5%	–
VDD1_E	DC supply voltage for LPDDR2 core-1 of MCP	1.8	±5%	–
VDD2_E	DC supply voltage for LPDDR2 core-2 of MCP	1.2	±5%	–
Ta	Operating ambient temperature	- 25 to 85		°C
Tj	Operating junction temperature	- 25 to 125		°C

NOTE: The typical voltage of ARM/INT/MIF/G3D could be changed by DVFS(Dynamic Voltage & Frequency Scaling) / ASV(Adapted Supply Voltage).

(1) The recommended voltage for VDD_ARM DVFS (Typical ASV group)

ARM Frequency	1.4GHz	1.2GHz	1.0GHz	800MHz	500MHz	200MHz
VDD_ARM	1.2875V±5%	1.1875V±5%	1.0875V±5%	0.9875V±5%	0.9375V±5%	0.90V±5%

(2) The recommended voltage for VDD_INT DVFS (Typical ASV group)

BUS Frequency	200MHz	160MHz	133MHz	100MHz
VDD_INT	1.0V±5%	0.9250V±5%	0.8875±5%	0.85V±5%

(3) The recommended voltage for VDD_MIF DVFS (Typical ASV group)

DRAM Frequency	400MHz	266MHz	200MHz	160MHz	133MHz	100MHz
VDD_MIF	1.0V±5%	0.95V±5%	0.95±5%	0.90V±5%	0.90V±5%	0.90V±5%

(4) The recommended voltage for VDD_G3D DVFS (Typical ASV group)

G3D Frequency	440MHz	350MHz	266MHz	160MHz	100MHz
VDD_G3D	1.025V±5%	0.95V±5%	0.90±5%	0.875V±5%	0.875V±5%

(5) If junction temperature is lower than 10°C, typical voltage shall be higher than 0.9V

59.3 D.C. Electrical Characteristics

The entire DC characteristics listed in *Table 59-3* for each pin include input sense levels, output drive levels, and currents.
Use these parameters to determine maximum DC loading and to determine maximum transition times for a given load.
Table 59-3 lists the DC operating conditions for the high- and low-strength input, output, and I/O pins.

Table 59-3 I/O DC Electrical Characteristics

(VDD = 1.65 V to 3.60 V, Vext = 3.0 to 3.6 V, Ta=-25 ~ 85°C and Tj=-25 ~ 125°C)

Symbol	Parameter	Condition		Min.	Typ.	Max.	Unit
Vtol	Tolerant external voltage	VDD power off and on		–	–	3.6	V
Vih	High level input voltage						
	CMOS interface	–		0.7×VDD	–	VDD	V
Vil	Low level input voltage						
	CMOS interface	VDD=2.5 V±10 %, 3.3V±10 %		- 0.3	–	0.7	V
		VDD = 1.8 V ± 10 %		- 0.3	–	0.3×VDD	
ΔV	Hysteresis volt-age	–		0.15	–	–	V
Iih	High level input current						
	Input buffer	Vin = VDD	VDD power on	- 3	–	3	µA
			VDD power off and SNS = 0	- 5	–	5	
	Input buffer with pull-down	Vin = VDD	VDD = 3.3V±10 %	20	45	80	
			VDD = 2.5V±10 %	20	40	80	
			VDD = 1.8V±10 %	20	40	80	
Iil	Low level input current						
	Input buffer	Vin = VSS	VDD power on and off	- 3	–	3	µA
	Input buffer with pull-up	Vin = VSS	VDD = 3.3 V ± 10 %	- 15	- 40	- 80	
			VDD = 2.5 V ± 10 %	- 15	- 40	- 80	
			VDD = 1.8 V ± 10 %	- 15	- 40	- 80	
Voh	Output high voltage	Ioh = – 1.8 mA, – 3.8 mA, – 7.8 mA, – 11 mA		0.8×VDD	–	VDD	V
Vol	Output low voltage	Iol = 1.8 mA, 3.8 mA, 7.8 mA, 11 mA		0	–	0.2×VDD	
Ioz	Output Hi-Z current	–		- 5	–	5	µA
CIN	Input capacitance	Any input and bidirectional buffers		–	–	5	pF

Table 59-4 lists TC OSC electrical characteristics.

Table 59-4 TC OSC Electrical Characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VIH	DC input logic high	–	0.7×VDDrtc	–	–	V
VIL	DC input logic low	–	–	–	0.3×VDDrtc	
IIH	High level input current	–	- 10	–	10	µA
IIL	Low level input current	–	- 10	–	10	

Table 59-5 lists USB DC electrical characteristics.

Table 59-5 USB DC Electrical Characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VIH	High level input voltage	–	2.0	–	–	V
VIL	Low level input voltage	–	–	–	0.8	
IIH	High level input current	Vin = 3.3 V	- 10	–	10	µA
IIL	Low level input current	Vin = 0.0 V	- 10	–	10	
VOH	Static output high	14.25 kΩ to GND	2.8	–	3.6	V
VOL	Static output low	1.425 kΩ to 3.6 V	–	–	0.3	
VBUS	Valid level voltage	–	4.4	–	5.25	

59.4 IO Driver Strength

This section includes:

- Input/Output Types
- Type A IO Driver Strength
- Type B IO Driver Strength

59.4.1 Input/ Output Types

Configurable Input/Output (I/O) is subdivided into Type A and Type B.

Table 59-6 lists the input/output types.

Table 59-6 I/O types

I/O Type	I/O Group	Description
A	GPA0, GPA1, GPB, GPC0, GPC1, GPD0, GPD1, GPF0, GPF1, GPF2, GPF3, GPJ0, GPJ1, GPK3, GPL0, GPL1,GPL2, GPM0, GPM1, GPM2, GPM3, GPM4, GPV0, GPV1, GPV2, GPV3, GPX0, GPX1, GPX2, GPX3, GPZ, MP0	Normal I/O (1.8 V I/O)
B	GPK0, GPK1, GPK2	Normal I/O (3.3 V tolerant I/O)

59.4.2 Type A IO Driver Strength

Table 59-7 lists the Driver Strength of Type A IO

Table 59-7 Driver Strength of Type A IO

1.8V IO	Drive Current @ slow PVT conditions					
	VDD=1.2V		VDD=1.5V		VDD=1.8V	
	IOH (mA)	IOL (mA)	IOH (mA)	IOL (mA)	IOH (mA)	IOL (mA)
DS0=0, DS1=0, SR=0/1	0.99	0.91	1.70	1.70	2.36	2.46
DS0=0, DS1=1, SR=0/1	1.99	1.82	3.42	3.43	4.72	4.94
DS0=1, DS1=0, SR=0/1	3.98	3.63	6.85	6.86	9.45	9.86
DS0=1, DS1=1, SR=0/1	5.97	5.45	10.27	10.30	14.17	14.80

59.4.3 Type B IO Driver Strength

Table 59-8 lists the Driver Strength of Type B IO.

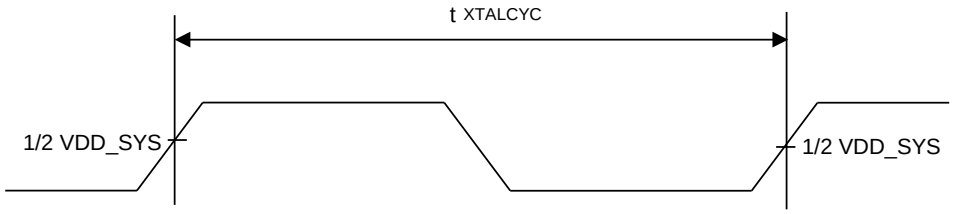
Table 59-8 Driver Strength of Type B IO

3.3V IO	Drive Current @ slow PVT conditions		
	VDD=1.8V	VDD=2.8V	VDD=3.3V
DS0=0, DS1=0, SR=0/1	1.9 mA	2.5 mA	3.0 mA
DS0=0, DS1=1, SR=0/1	3.8 mA	5.0 mA	6.0 mA
DS0=1, DS1=0, SR=0/1	7.6 mA	10.0 mA	12.0 mA
DS0=1, DS1=1, SR=0/1	11.4 mA	15.0 mA	18.0 mA

59.5 CLK Alternating Current Electrical Characteristics

Alternating Current (AC) characteristics of pin include input and output capacitance. These factors determine the loading for external drivers and other load analyses.

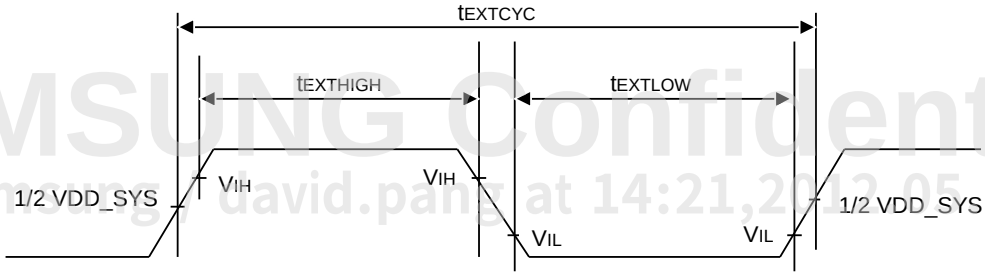
Figure 59-1 illustrates the X_{TIpll} Clock Timing.



NOTE: The clock input from the X_{TIpll} pin.

Figure 59-1 X_{TIpll} Clock Timing

Table 59-9 illustrates the EXTCLK clock input timing.



NOTE: The clock input from the EXTCLK pin.

Figure 59-2 EXTCLK Clock Input Timing

Table 59-9 lists clock timing constants.

Table 59-9 Clock Timing Constants

(Ta=-25 ~ 85°C and Tj=-25 ~ 125°C, VDDSYS = 1.8 V ± 5 %)

Parameter	Symbol	Min.	Typ.	Max.	Unit
Duration for crystal oscillator clock input	$t_{XTALCYC}$	40	–	–	ns
High width for external clock input	$t_{EXTHIGH}$	20	–	–	
Low width for external clock input	t_{EXTLOW}	20	–	–	
APLL lock time	t_{APLL}	–	–	100	usec
MPLL lock time	t_{MPLL_LT}	–	–	400	X_{TIpll} or EXTCLK
EPLL lock time	t_{EPLL_LT}	–	–	3000	
VPLL lock time	t_{VPLL_LT}	–	–	400	

Figure 59-3 illustrates the manual reset input timing.

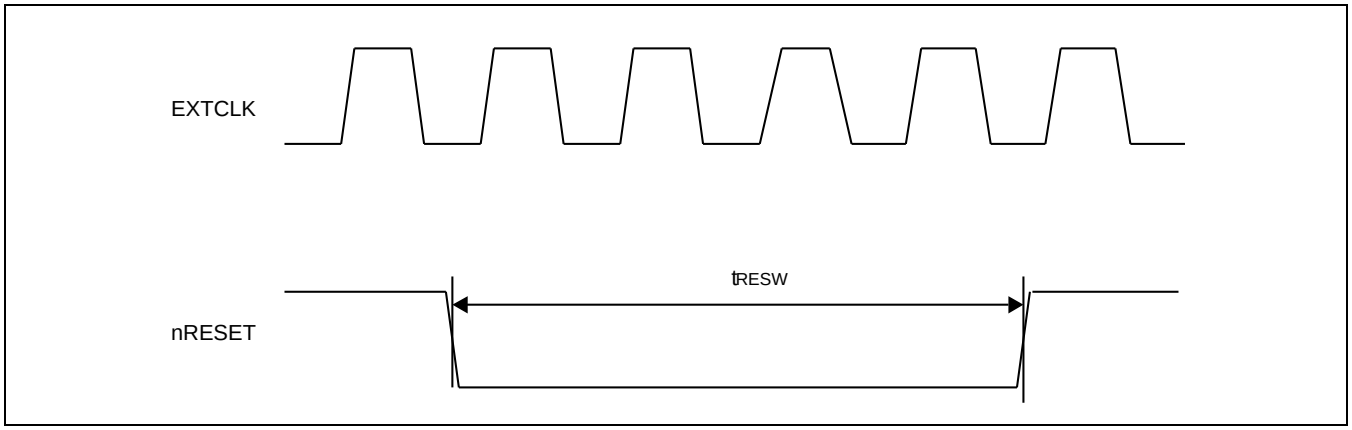


Figure 59-3 Manual Reset Input Timing

Figure 59-4 illustrates the power-on reset sequence.

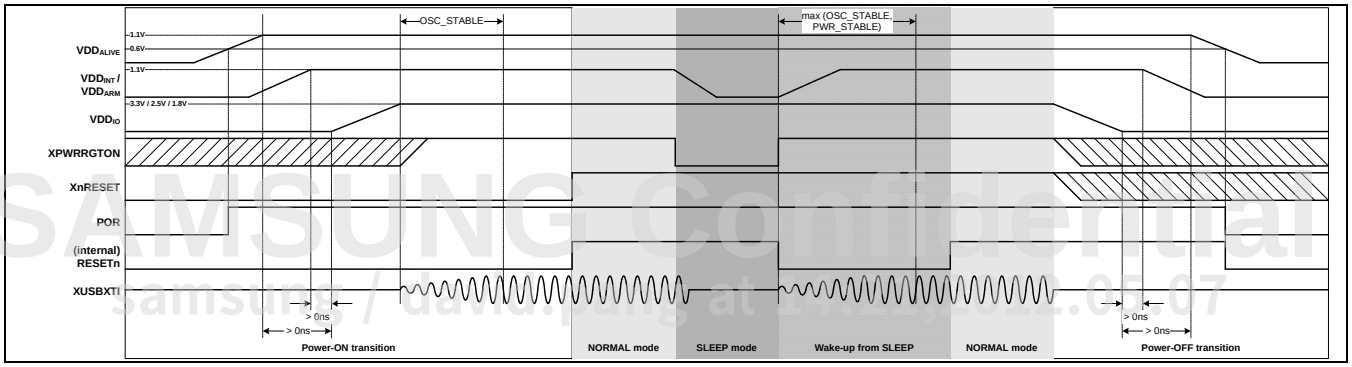


Figure 59-4 Power-On Reset Sequence

OSC STABLE in Figure 59-4 indicates the time required for the oscillator pad to be stabilized.

Table 59-10 lists the power-on reset timing specifications.

Table 59-10 Power-On Reset Timing Specifications

(Ta=-25 ~ 85°C and Tj=-25 ~ 125°C, VDDSYS =1.8 V ± 0.15 V)

Parameter	Symbol	Min.	Typ.	Max.	Unit
Reset assert time after clock stabilization	t_{RESW}	4	–	–	XTIpll or EXTCLK

59.6 ROM/SRAM AC Electrical Characteristics

Figure 59-5 illustrates the ROM/SRAM timing.

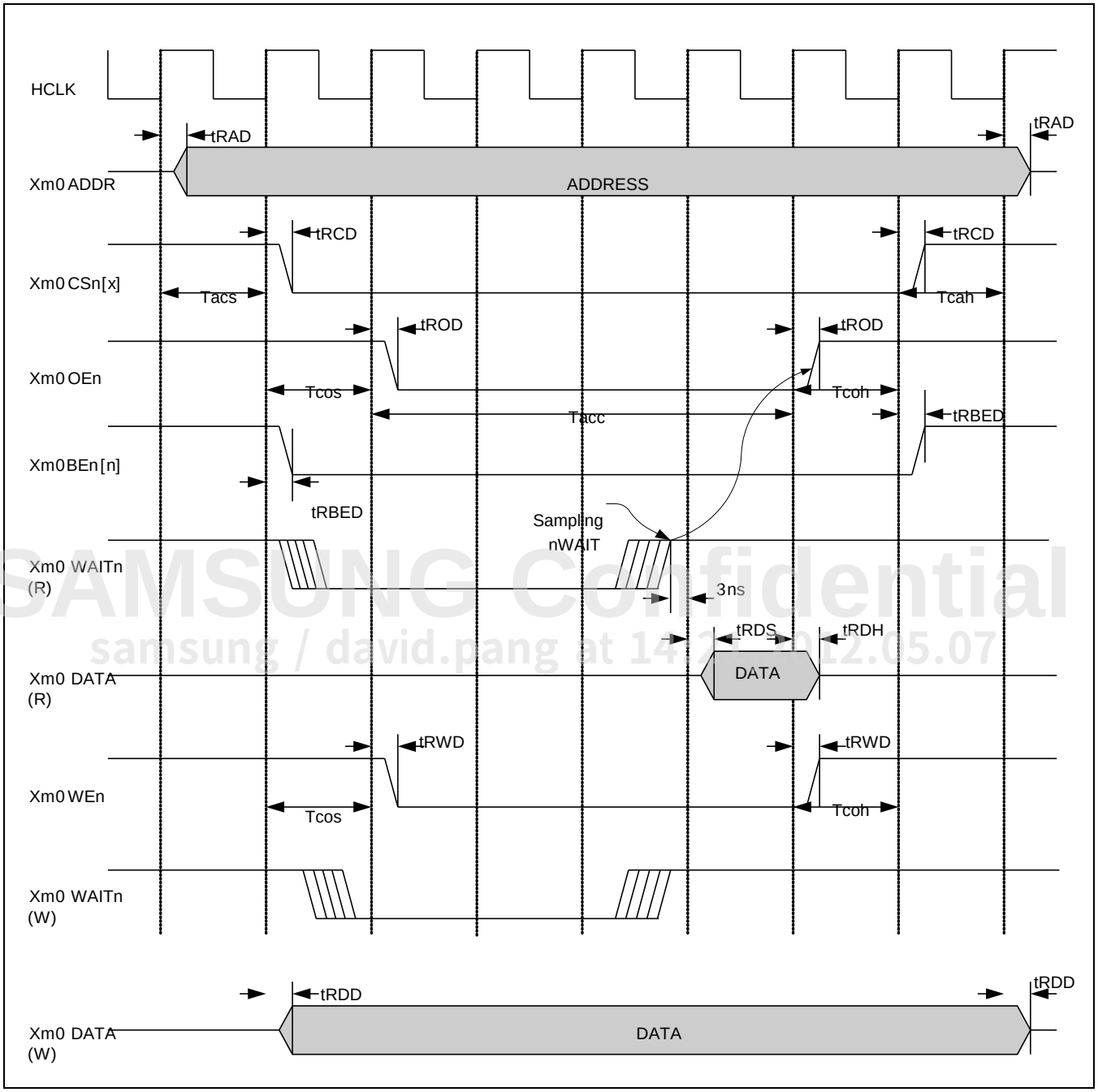


Figure 59-5 ROM/SRAM Timing

NOTE: Tacs = 0, Tcos = 0, Tacc = 2, Tcoh = 0, Tcah = 0, PMC = 0, ST = 0, DW = 16-bit

Table 59-11 lists the ROM/SRAM bus timing constants.

Table 59-11 ROM/SRAM Bus Timing Constants

(Ta=-25 ~ 85°C and Tj=-25 ~ 125°C, VDD m0 = 1.8 V ± 10 %)

Parameter	Symbol	Min.	Typ.	Max.	Unit
ROM/SRAM address delay	t_{RAD}	1.77	–	5.22	ns
ROM/SRAM chip select 0 delay	t_{RCD}	3.39	–	8.06	
ROM/SRAM chip select 1 delay	t_{RCD}	3.48	–	8.33	
ROM/SRAM chip select 2 delay	t_{RCD}	4.13	–	7.29	
ROM/SRAM chip select 3 delay	t_{RCD}	3.14	–	7.22	
ROM/SRAM chip select 4 delay	t_{RCD}	2.16	–	4.21	
ROM/SRAM chip select 5 delay	t_{RCD}	2.30	–	4.53	
ROM/SRAM nOE (output enable) delay	t_{ROD}	2.78	–	5.46	
ROM/SRAM new (write enable) delay	t_{RWD}	2.02	–	3.96	
ROM/SRAM byte enable delay	t_{RBED}	2.95	–	6.52	
ROM/SRAM output data delay	t_{RDD}	3.50	–	7.41	
ROM/SRAM read data setup time	t_{RDS}	2.00	–	–	
ROM/SRAM write data hold time	t_{RDH}	1.00	–	–	

59.7 NFCON AC Electrical Characteristics

Figure 59-6 illustrates the NAND flash timing.

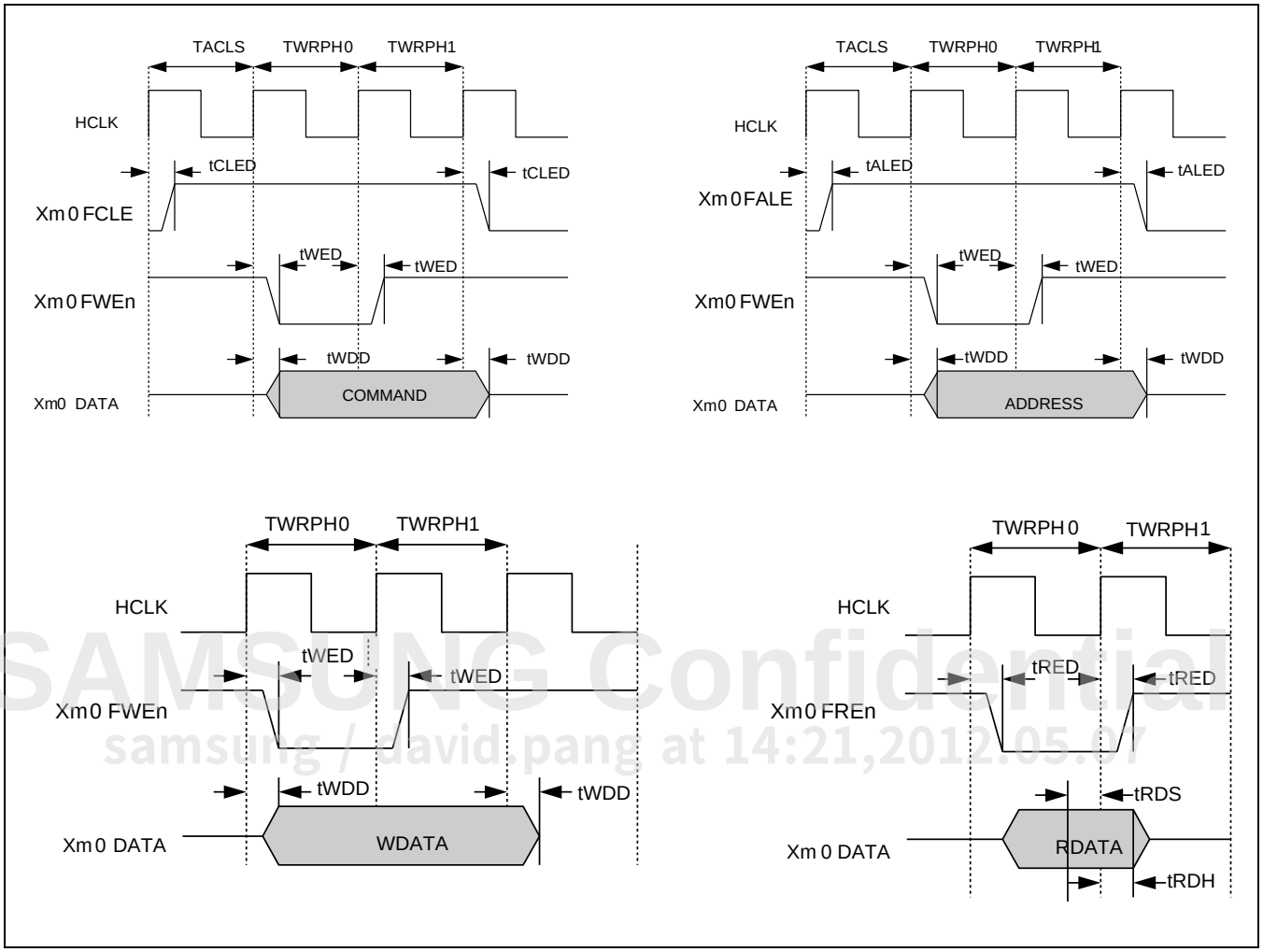


Figure 59-6 NAND Flash Timing

Table 59-12 lists the NFCON bus timing constants.

Table 59-12 NFCON Bus Timing Constants

(Ta=-25 ~ 85°C and Tj=-25 ~ 125°C, VDD m0 = 1.8 V ± 10 %)

Parameter	Symbol	Min.	Typ.	Max.	Unit
NFCON chip enable delay	t_{CED}	–	–	8.40	ns
NFCON CLE delay	t_{CLED}	–	–	4.45	
NFCON ALE delay	t_{ALED}	–	–	5.36	
NFCON write enable delay	t_{WED}	–	–	8.15	
NFCON read enable delay	t_{RED}	–	–	8.21	
NFCON write data delay	t_{WDD}	–	–	5.39	
NFCON read data setup requirement time	t_{RDS}	1.00	–	–	
NFCON read data hold requirement time	t_{RDH}	0.20	–	–	

59.8 LPDDR2 Electrical Characteristics

Refer to Chapter 18 DMC user's manual for more information.

59.9 LCD Controller AC Electrical Characteristics

Figure 59-7 illustrates the LCD controller timing.

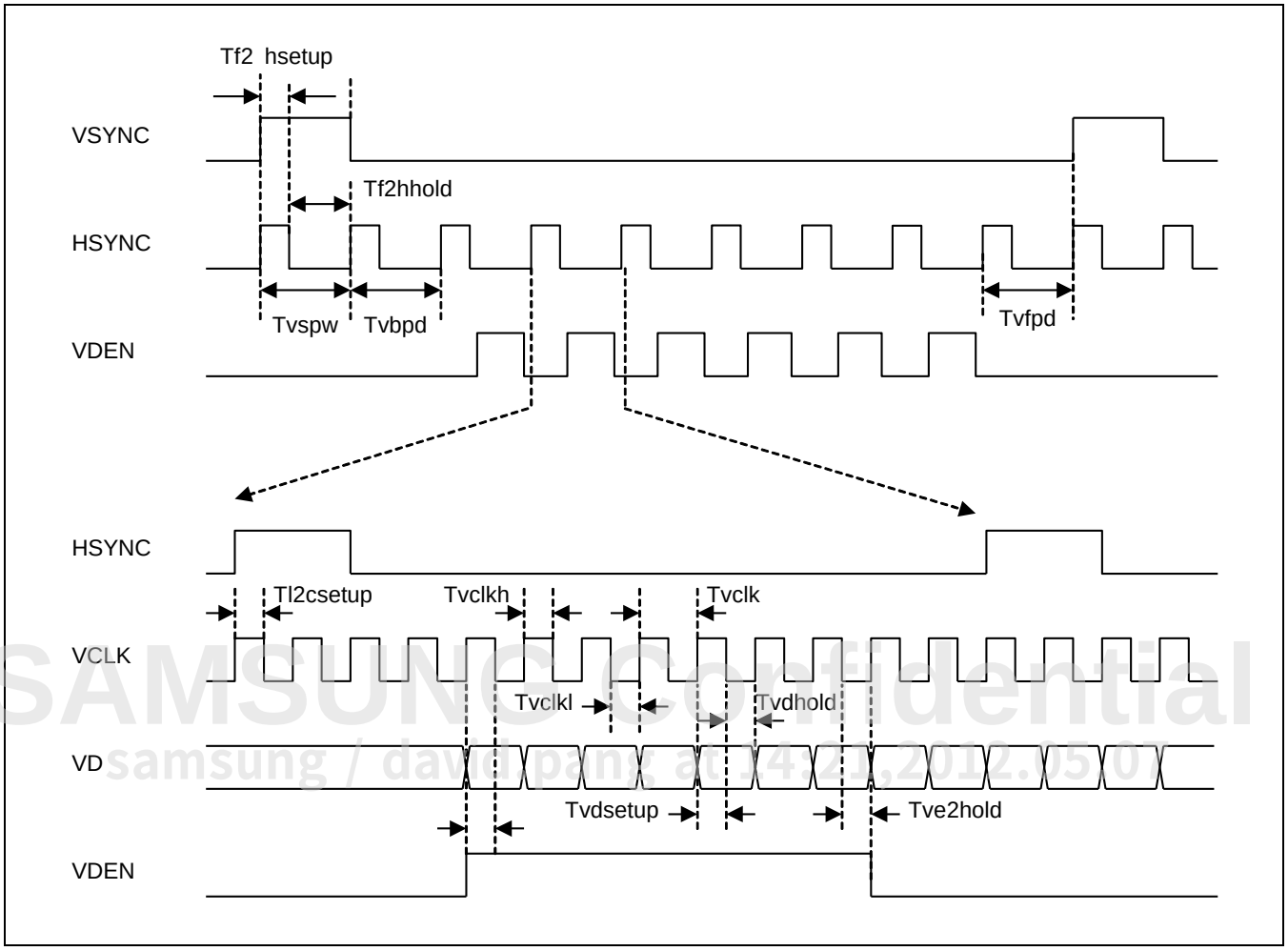


Figure 59-7 LCD Controller Timing

Table 59-13 lists the TFT LCD controller module signal timing constants.

Table 59-13 TFT LCD Controller Module Signal Timing Constants

(Ta=-25 ~ 85°C and Tj=-25 ~ 125°C, VDDlcd = 1.8 V ± 10 %)

Parameter	Symbol	Min.	Typ.	Max.	Unit
VCLK pulse width	Tvclk	12	–	–	ns
VCLK pulse width high	Tvclkh	0.3	–	–	Pvclk (1)
VCLK pulse width low	Tvclkl	0.3	–	–	Pvclk
Vertical sync pulse width	Tvspw	VSPW + 1	–	–	Phclk (2)
Vertical back porch delay	Tvbpd	VBPD + 1	–	–	Phclk
Vertical front porch delay	Tvfpd	VFPD + 1	–	–	Phclk
Hsync setup to VCLK falling edge	Tl2csetup	0.3	–	–	Pvclk
VDEN setup to VCLK falling edge	Tde2csetup	0.3	–	–	Pvclk
VDEN hold from VCLK falling edge	Tde2chold	0.3	–	–	Pvclk
VD setup to VCLK falling edge	Tvd2csetup	0.3	–	–	Pvclk
VD hold from VCLK falling edge	Tvd2chold	0.3	–	–	Pvclk
VSYNC setup to HSYNC falling edge	Tf2hsetup	HSPW + 1	–	–	Pvclk
VSYNC hold from HSYNC falling edge	Tf2hhold	HBPD + HFPD + HOZVAL + 3	–	–	Pvclk

NOTE:

1. VCLK period.
2. HSYNC period.

Figure 59-8 illustrates the LCD I80 interface timing.

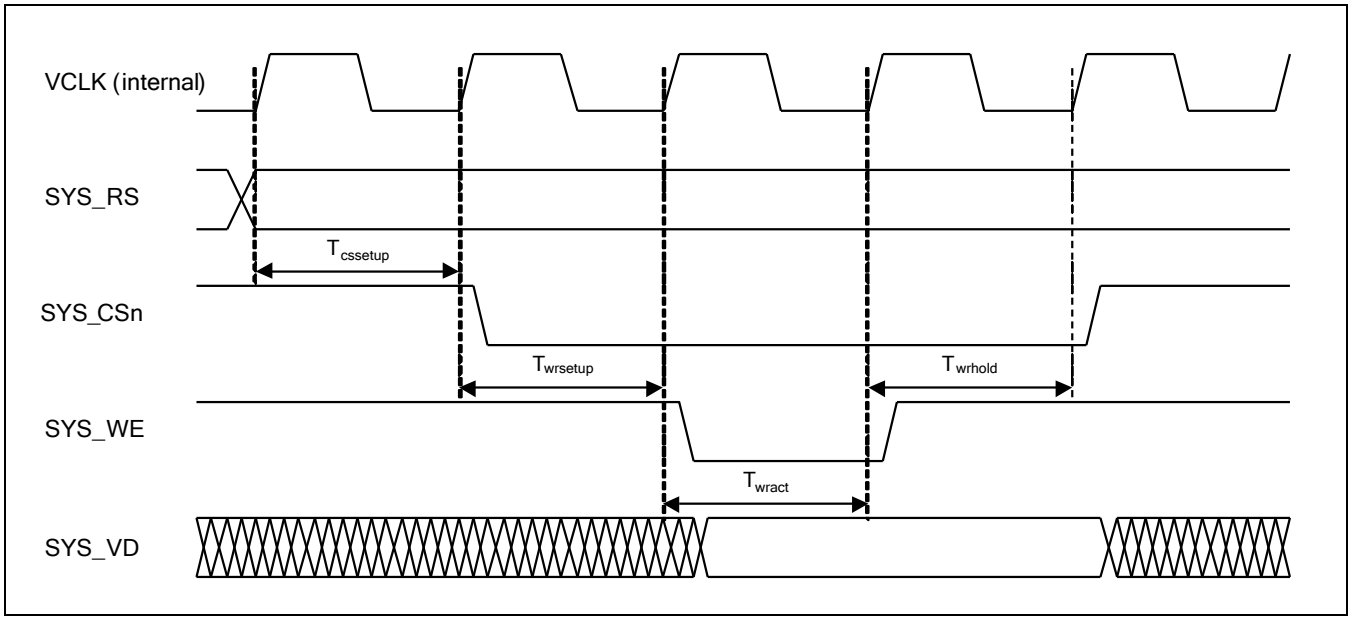


Figure 59-8 LCD I80 Interface Timing

Table 59-14 lists the LCD I80 interface signal timing constants.

Table 59-14 LCD I80 Interface Signal Timing Constants

(Ta=-25 ~ 85°C and Tj=-25 ~ 125°C, VDDlcd = 1.8 V ± 10 %)

Parameter	Symbol	Min.	Typ.	Max.	Unit
SYS_RS to SYS_CSn low	Tcssetup	–	LCD_CS_SETUP + 1	–	Pvclk (NOTE)
SYS_CSn low to SYS_WR low	Twrsetup	–	LCD_WR_SETUP + 1	–	Pvclk
SYS_WE pulse width	Twract	–	LCD_WR_ACT + 1	–	Pvclk
SYS_WE high to SYS_CSn high	Twrhold	–	LCD_WR_HOLD + 1	–	Pvclk

NOTE: Internal VCLK period.

59.10 Camera Interface AC Electrical Characteristics

Figure 59-9 illustrates the camera interface VSYNC timing.

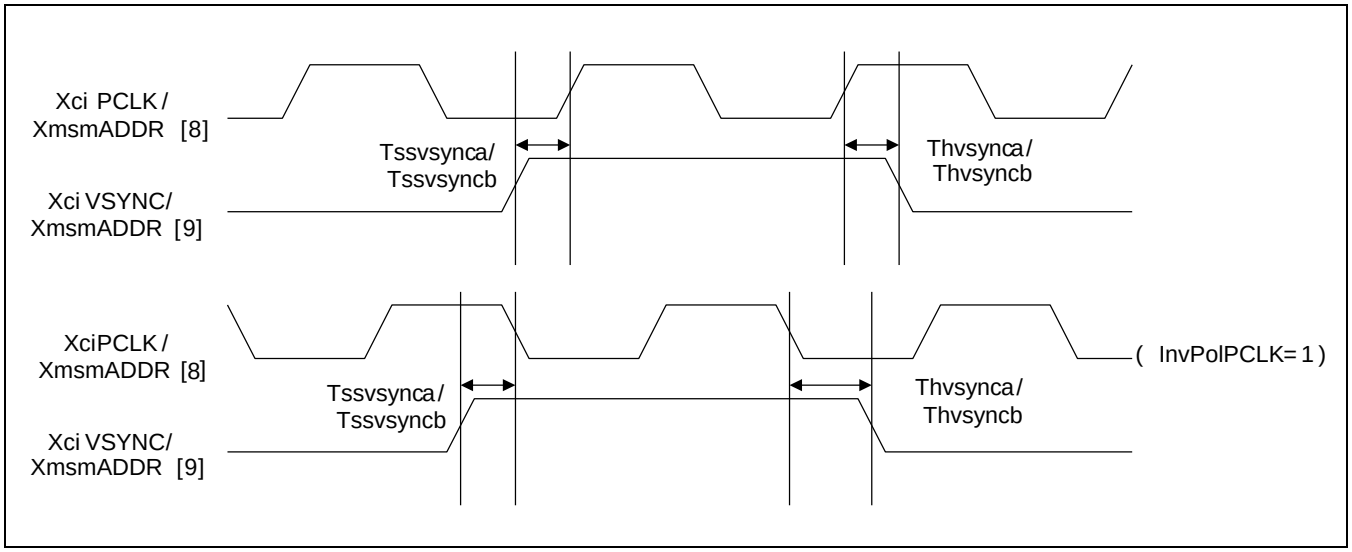


Figure 59-9 Camera Interface VSYNC Timing

Figure 59-10 illustrates the camera interface HREF timing.

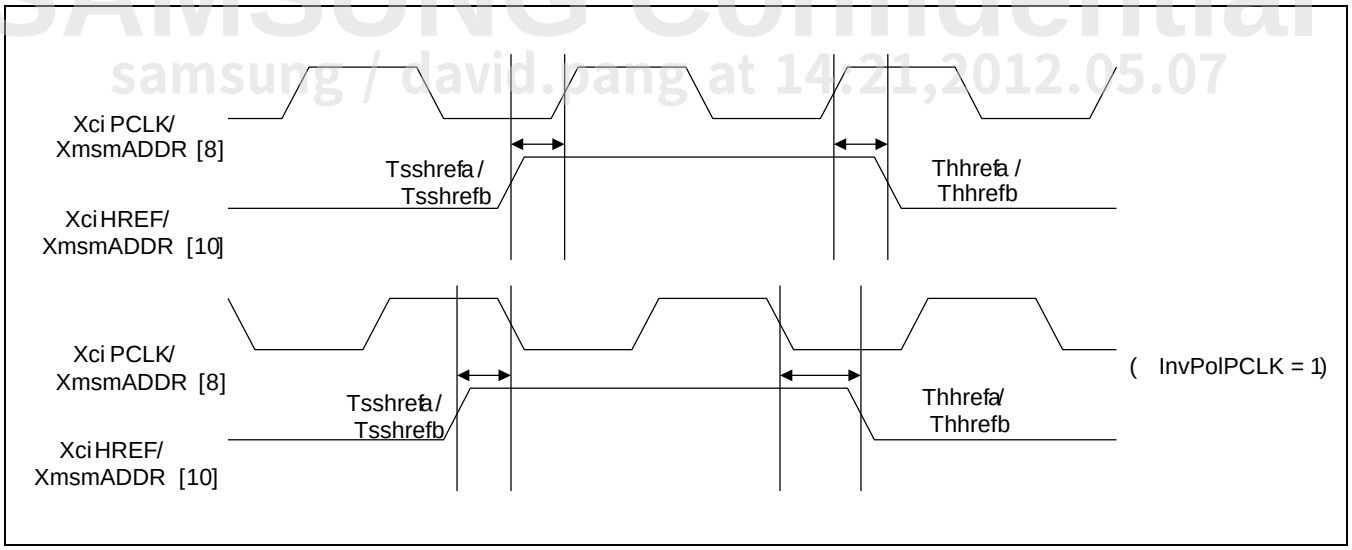


Figure 59-10 Camera Interface HREF Timing

Figure 59-11 illustrates the camera interface data timing.

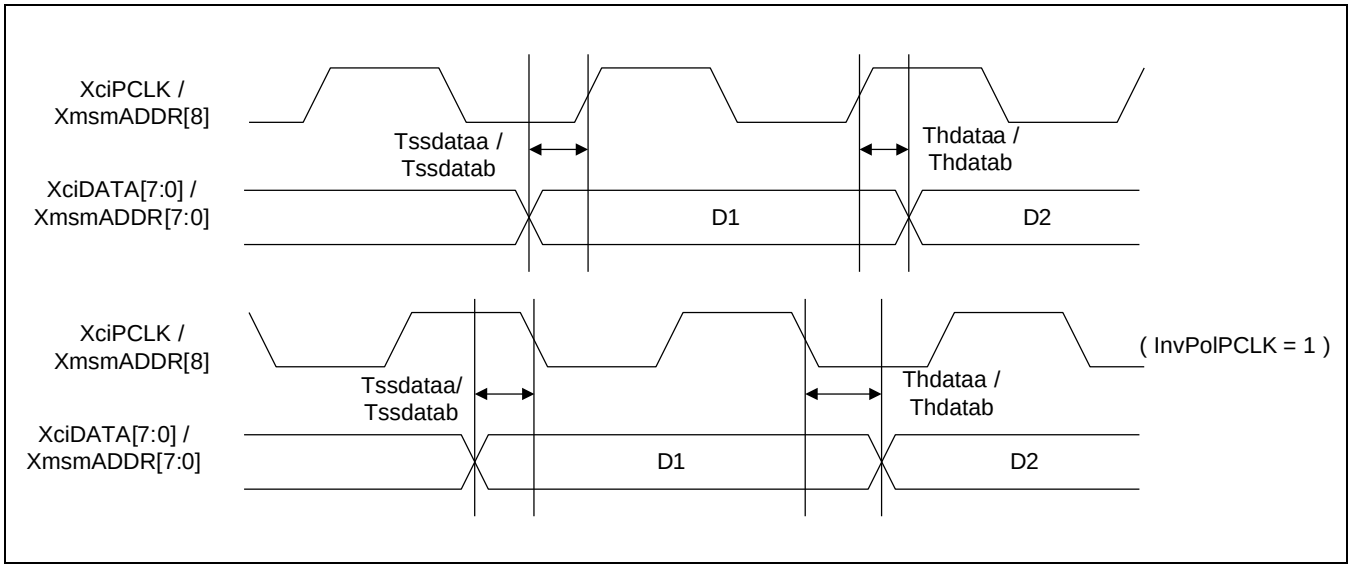
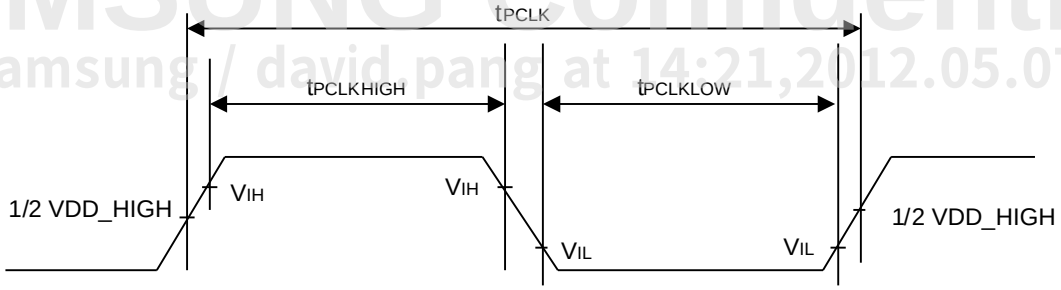


Figure 59-11 Camera Interface Data Timing

Figure 59-12 illustrates the camera interface PCLK timing.



NOTE: The clock input from the XciPCLK or XmsmADDR[8] pin.

Figure 59-12 Camera Interface PCLK Timing

Table 59-15 lists the camera controller module signal timing constants.

Table 59-15 Camera Controller Module Signal Timing Constants

(Ta=-25 ~ 85°C and Tj=-25 ~ 125°C, VDDext = 1.8 V ± 10 %)

Parameter	Symbol	Min.	Typ.	Max.	Unit
XciVSYNC (CAM_VSYNC_A) input setup time	Tssvsynca	0	–	PA - 0.26	ns
XciVSYNC (CAM_VSYNC_A) input hold time	Thvsynca	0.26	–	PA	ns
XciHREF (CAM_HREF_A) input setup time	Tsshrefa	0.35	–	PA	ns
XciHREF (CAM_HREF_A) input hold time	Thhrefa	0	–	PA - 0.35	ns
XciDATA (CAM_DATA_A) input setup time	Tssdataa	0	–	PA - 4.8	ns
XciDATA (CAM_DATA_A) input hold time	Thdataa	4.8	–	PA	ns
XmsmADDR[9] (CAM_VSYNC_B) input setup time	Tssvsyncb	0	–	PB	ns
XmsmADDR[9] (CAM_VSYNC_B) input hold time	Thvsyncb	0	–	PB	ns
XmsmADDR[10] (CAM_HREF_B) input setup time	Tsshrefb	0.08	–	PB	ns
XmsmADDR[10] (CAM_HREF_B) input hold time	Thhrefb	0	–	PB - 0.08	ns
XmsmADDR[7:0] (CAM_DATA_B) input setup time	Tssdatab	0	–	PB - 4.93	ns
XmsmADDR[7:0] (CAM_DATA_B) input hold time	Thdatab	4.93	–	PB	ns
PCLK period	t_{PCLK}	12	–	–	ns
PCLK input high level pulse width	$t_{PCLKHIGH}$	3	–	–	ns
PCLK input low level pulse width	$t_{PCLKLOW}$	3	–	–	ns

NOTE:

1. PA denotes period (ns) of CAM_PCLK_A.
2. PB denotes period (ns) of CAM_PCLK_B.

59.11 SDMMC AC Electrical Characteristics

Figure 59-13 illustrates the high speed SDMMC interface timing.

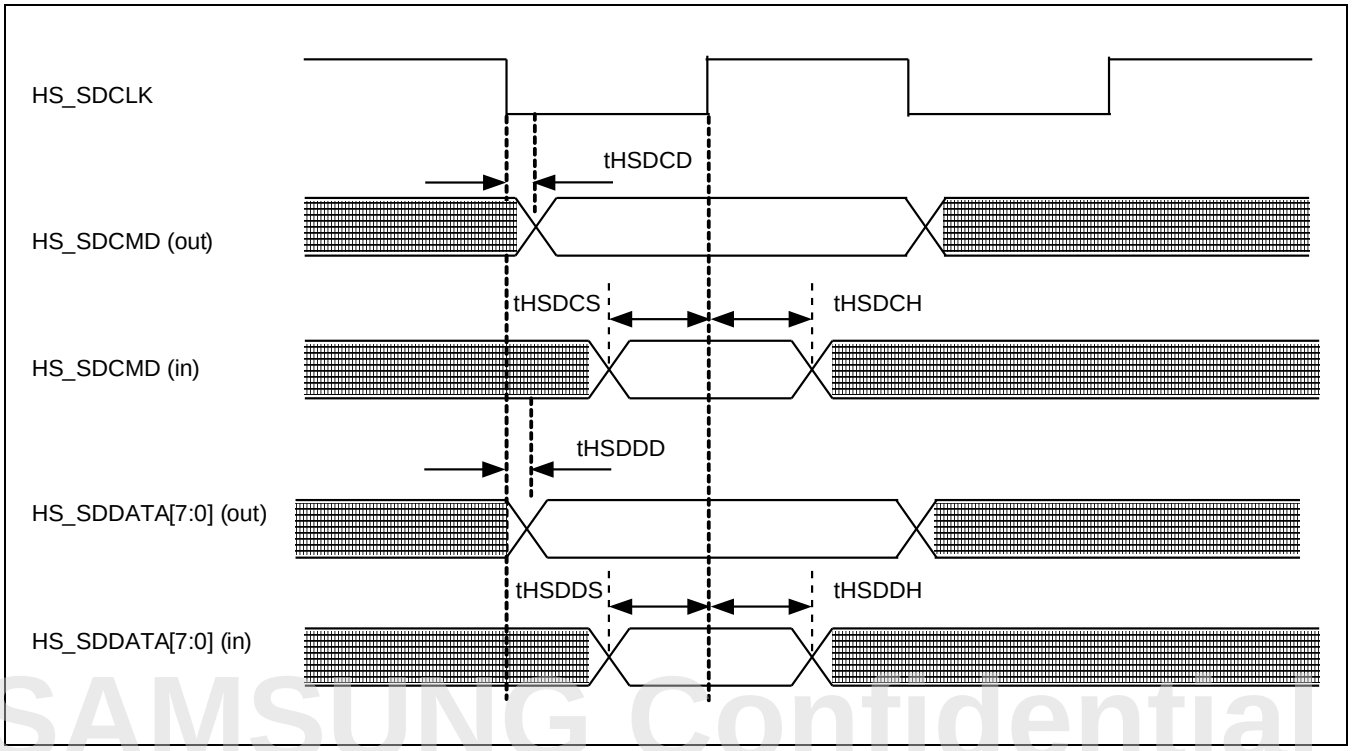


Figure 59-13 High Speed SDMMC Interface Timing

Table 59-16 lists the high speed SDMMC interface Transmit/Receive timing constants.

Table 59-16 High Speed SDMMC Interface Transmit/Receive Timing Constants

(Ta=-25 ~ 85°C and Tj=-25 ~ 125°C, VDDQ_MMC01/2 = 3.3 V ± 5 %, 2.5 V ± 5 %, 1.8 V ± 5 %, VDDQ_MMC3 = 1.8 V ± 5 %)

Parameter	Symbol	Min.	Typ.	Max.	Unit
SD command output delay time	t_{SDCD}	–	–	4.0	ns
SD command input setup time	t_{SDCS}	4.0	–	–	ns
SD command input hold time	t_{SDCH}	0	–	–	ns
SD data output delay time	t_{SDDD}	–	–	4.0	ns
SD data input setup time	t_{SDDS}	4.0	–	–	ns
SD data input hold time	t_{SDDH}	0	–	–	ns

59.12 SPI AC Electrical Characteristics

Figure 59-14 illustrates the SPI interface timing (CPHA = 0, CPOL = 1).

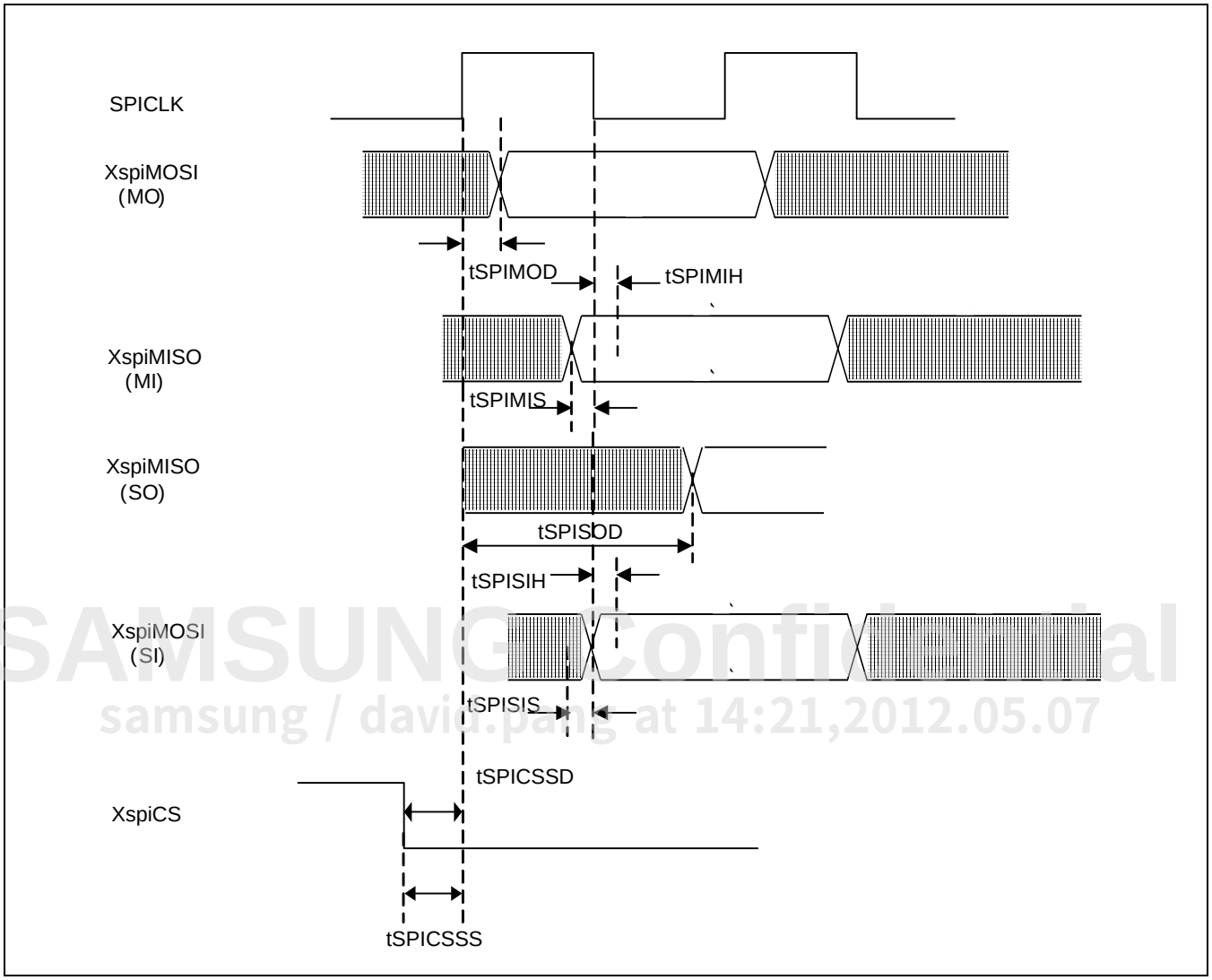


Figure 59-14 SPI Interface Timing (CPHA = 0, CPOL = 1)

Table 59-17 lists the SPI interface Transmit/Receive timing constants (VDDext = 1.8 V).

Table 59-17 SPI Interface Transmit/Receive Timing Constants (VDDext = 1.8 V)

(Ta=-25 ~ 85°C and Tj=-25 ~ 125°C, VDDext = 1.8 V ± 10 %, Load = 15 pF)

Parameter		Symbol	Min.	Typ.	Max.	Unit
Ch 0	SPI MOSI master output delay time	t_{SPIMOD}	–	–	5	ns
	SPI MISO master input setup time (FB_CLK_SEL = 00)	t_{SPIMIS}	12	–	–	ns
	SPI MISO master input setup time (FB_CLK_SEL = 01)		7	–	–	ns
	SPI MISO master input setup time (FB_CLK_SEL = 10)		2	–	–	ns
	SPI MISO master input setup time (FB_CLK_SEL = 11)		- 3	–	–	ns
	SPI MISO master input hold time	t_{SPIMIH}	5	–	–	ns
	SPI MOSI slave input setup time	t_{SPISIS}	2	–	–	ns
	SPI MOSI slave input hold time	t_{SPISIH}	5	–	–	ns
	SPI MISO slave output delay time	t_{SPISOD}	–	–	17	ns
	SPI nSS master output delay time	$t_{SPICSSD}$	7	–	–	ns
	SPI nSS slave input setup time	$t_{SPICSSS}$	5	–	–	ns
Ch 1	SPI MOSI master output delay time	t_{SPIMOD}	–	–	4	ns
	SPI MISO master input setup time (FB_CLK_SEL = 00)	t_{SPIMIS}	13	–	–	ns
	SPI MISO master input setup time (FB_CLK_SEL = 01)		8	–	–	ns
	SPI MISO master input setup time (FB_CLK_SEL = 10)		3	–	–	ns
	SPI MISO master input setup time (FB_CLK_SEL = 11)		- 2	–	–	ns
	SPI MISO master input hold time	t_{SPIMIH}	5	–	–	ns
	SPI MOSI slave input setup time	t_{SPISIS}	3	–	–	ns
	SPI MOSI slave input hold time	t_{SPISIH}	5	–	–	ns
	SPI MISO slave output delay time	t_{SPISOD}	–	–	18	ns
	SPI nSS master output delay time	$t_{SPICSSD}$	7	–	–	ns
	SPI nSS slave input setup time	$t_{SPICSSS}$	5	–	–	ns

NOTE: SPICLKout = 50 MHz

$t_{SPIMIS,CH0}$ = 12 – (Cycle period/4) × FB_CLK_SEL

$t_{SPIMIS,CH1}$ = 13 – (Cycle period/4) × FB_CLK_SEL

59.13 I2C AC Electrical Characteristics

Figure 59-15 illustrates the IIC interface timing.

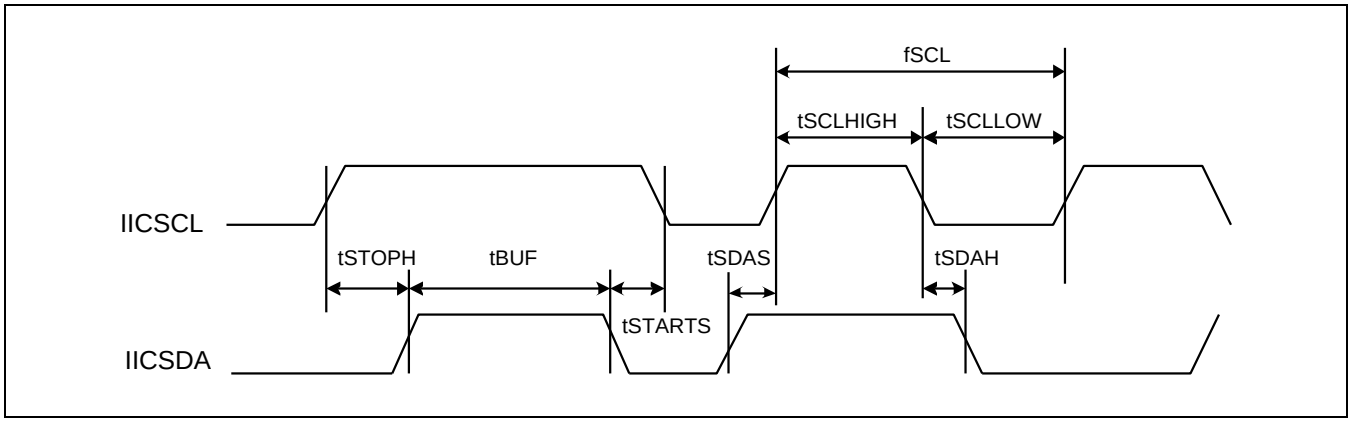


Figure 59-15 IIC Interface Timing

Table 59-18 lists the IIC BUS controller module signal timing.

Table 59-18 IIC BUS Controller Module Signal Timing

(Ta=-25 ~ 85°C and Tj=-25 ~ 125°C, VDDext = 1.8 V ± 10 %)

Parameter	Symbol	Min.	Typ.	Max.	Unit
SCL clock frequency	f_{SCL}	–	–	std. 100 fast 400	kHz
SCL high level pulse width	$t_{SCLHIGH}$	std. 4.0 fast 0.6	–	–	µs
SCL low level pulse width	t_{SCLLOW}	std. 4.7 fast 1.3	–	–	µs
Bus free time between stop and start	t_{BUF}	std 4.7 fast 1.3	–	–	µs
START hold time	t_{STARTS}	std. 4.0 fast 0.6	–	–	µs
SDA hold time	t_{SDAH}	std. 0 fast 0	–	std.-fast 0.9	µs
SDA setup time	t_{SDAS}	std. 250 fast 100	–	–	ns
STOP setup time	t_{STOPH}	std. 4.0 fast 0.6	–	–	µs

NOTE: std. refers to standard mode and fast refers to fast mode.

1. The IIC data hold time (t_{SDAH}) is minimum 0 ns. (IIC data hold time is minimum 0 ns for standard/ fast bus mode IIC specification Version 2.1)
   Verify whether the data hold time of your IIC device is 0 ns or not.
2. The IIC controller supports IIC bus device only (standard/fast bus mode), and does not support C bus device

60 Mechanical Data

This chapter describes the Mechanical Data of Exynos 4412 SCP.

60.1 Physical Dimension

This section includes:

- Pin Assignment Diagram 786-ball FCFBGA (SCP, Single Chip Package)
- Package Dimension

60.1.1 Pin Assignment Diagram 786-ball FCFBGA (SCP)

Figure 60-1 illustrates the bottom view of Exynos 4412 SCP pin assignment (786-FCFBGA).

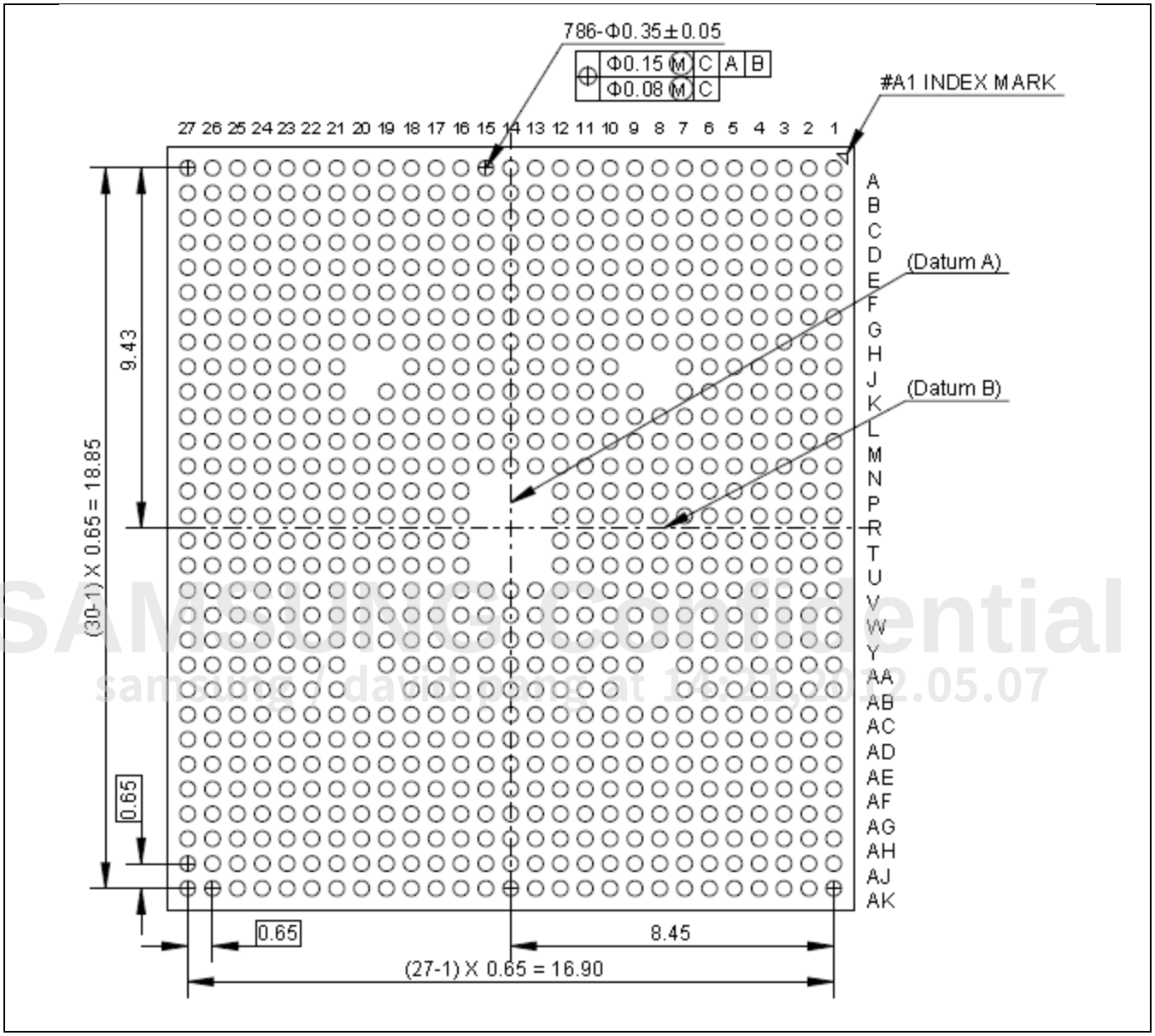


Figure 60-1 Exynos 4412 SCP Pin Assignment (784-FCFBGA) Bottom View

60.1.2 Package Dimension

Figure 60-2 illustrates the top view of Exynos 4412 SCP package dimension (784-FCFBGA).

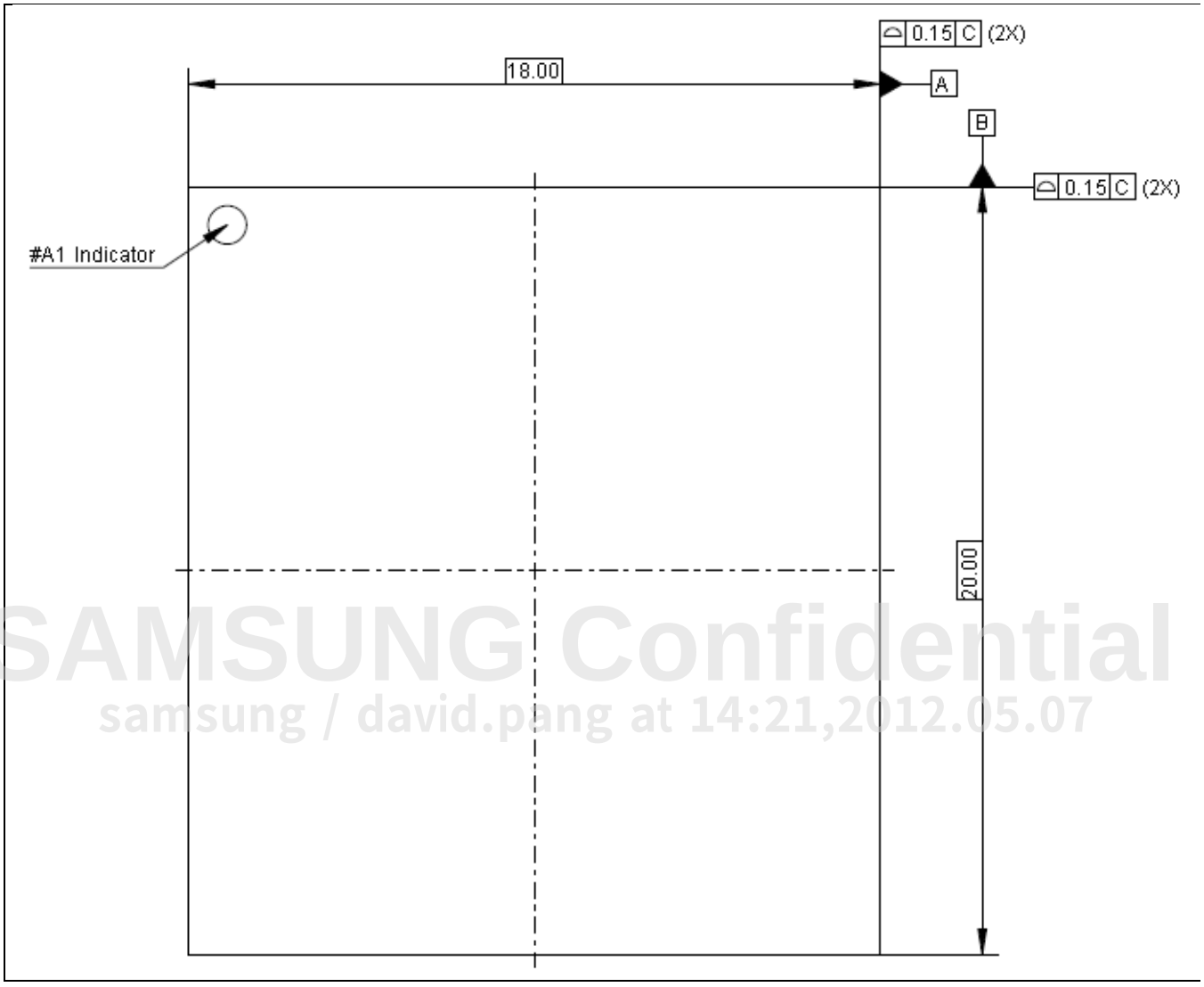


Figure 60-2 Exynos 4412 SCP Package Dimension (804-FCMSP) Top View

Figure 60-3 illustrates the side view of Exynos 4412 SCP package dimension (784-FCFBGA).

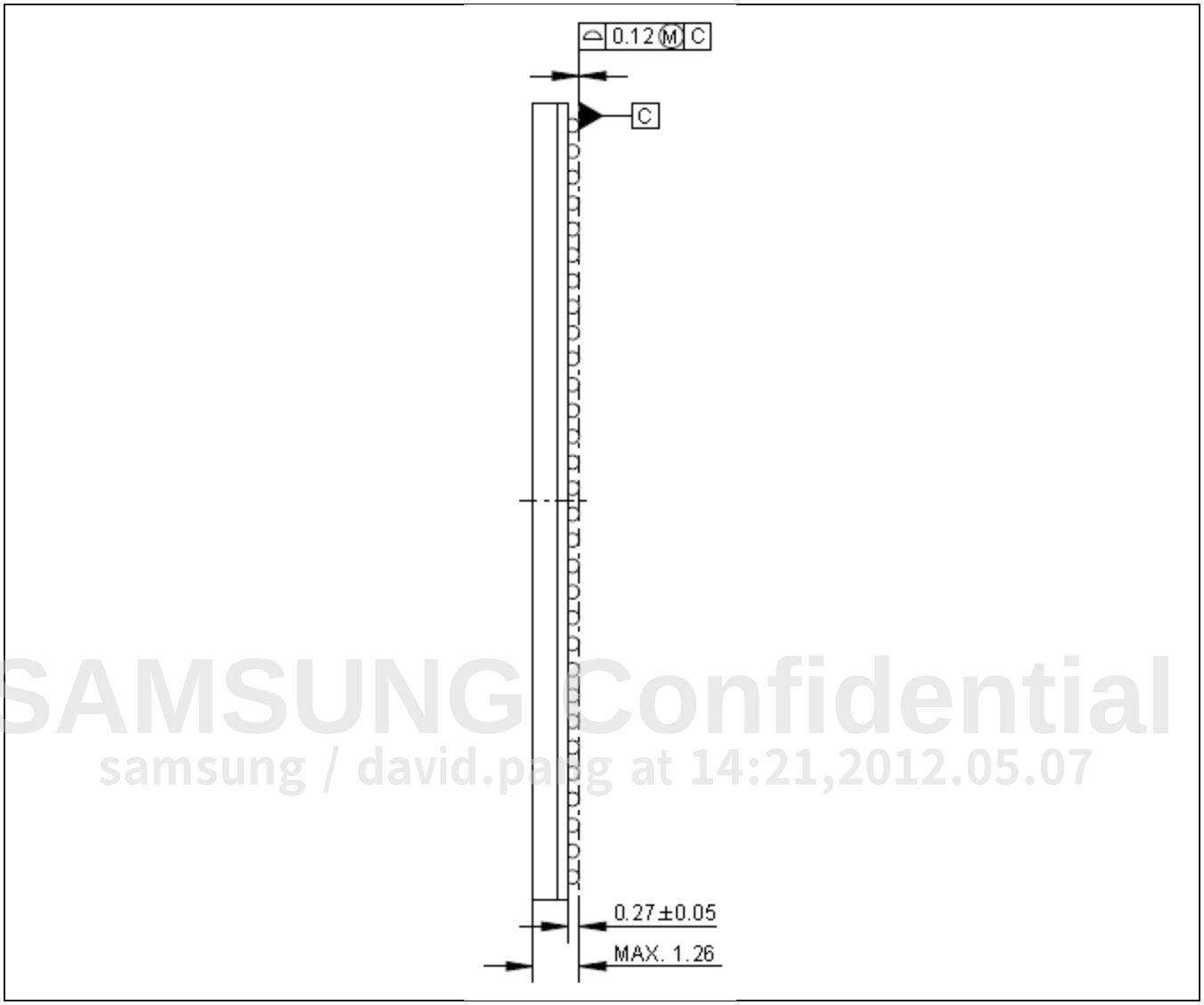


Figure 60-3 Exynos 4412 SCP Package Dimension (784-FCFBGA) Side View

60.2 Package Marking

Figure 60-4 illustrates the marking-top view of Exynos 4412 SCP package.

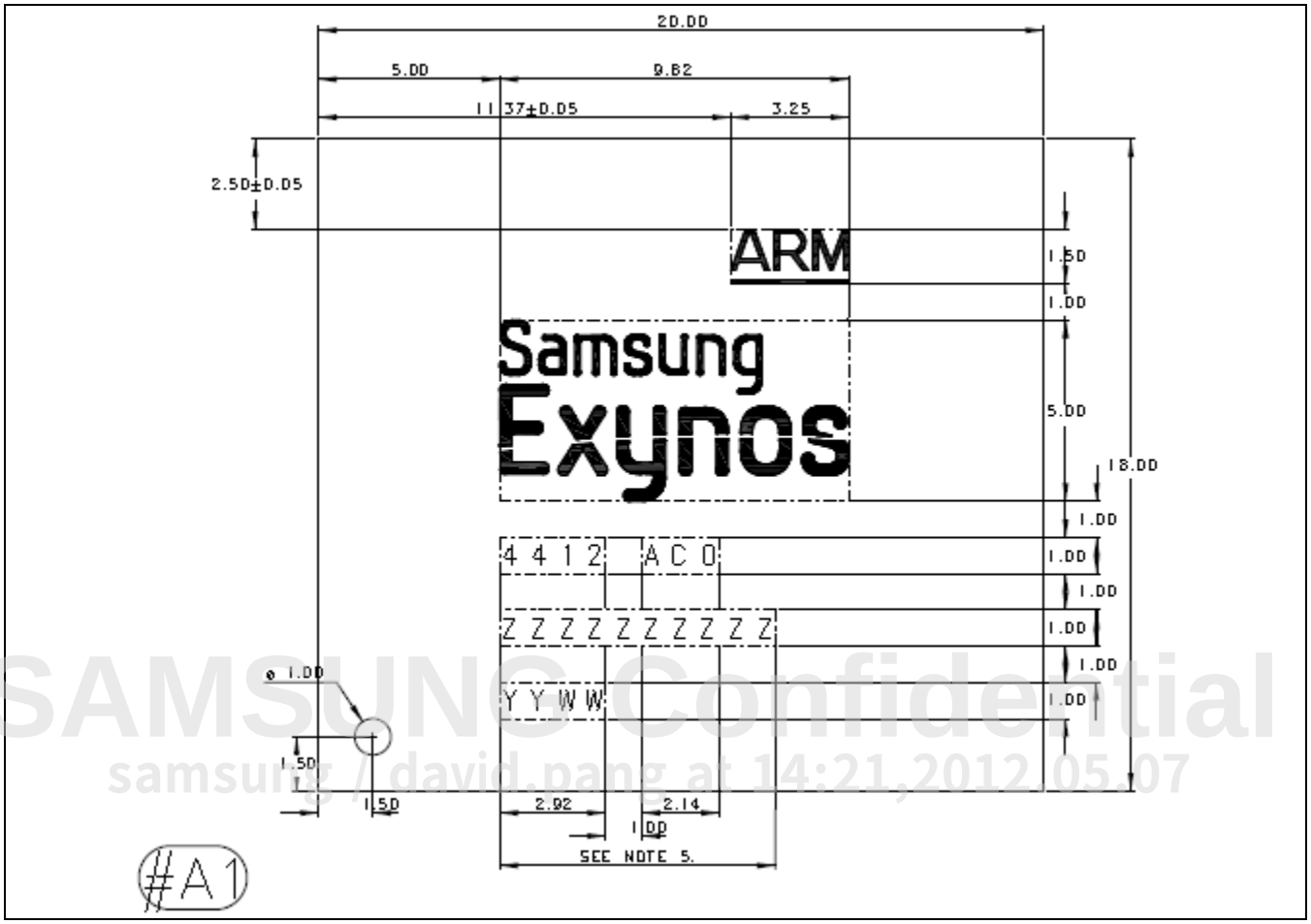


Figure 60-4 Exynos 4412 SCP Package Marking-Top View

60.2.1 Package Marking Line Description

Table 60-1 describes the package marking line description.

Table 60-1 Package Marking Line Description

No	Marking	Description
1	4412	Exynos Code
2	AD0	Device Option
3	ZZZZZZZZZZ	Lot number
4	YYWW	Work week YY = year (two digit) WW = week (two digit, based on calendar year)