Freescale Semiconductor
Addendum

Document Number: MPC8379ERMAD
Rev. 1.2, 04/2011

Errata to MPC8379E PowerQUICC II Pro Integrated Host Processor Family Reference Manual, Rev. 1

This errata describes corrections to the *MPC8379E PowerQUICC II Pro Integrated Host Processor Family Reference Manual*, Revision 1. For convenience, the section number and page number of the errata item in the reference manual are provided. Items in bold are new since the last revision of this document.

To locate any published updates for this document, visit our website listed on the back cover of this document.

Section, Page No. **Changes**

4.3.3.3.1, 4-31 Add the following column on the right of the “1010” column in Table 4-26, “Examples For Hard-Coded Reset Configuration Words Usage.”

Table 4-26. Examples For Hard-Coded Reset Configuration Words Usage

CFG_RESET_SOURCE[0:3]	1000	1001	1010	1011	1101	1110	1111
PCI_CLK (MHz)	66	33	33	50	66	33	66
csb_clk (MHz)	333	133	266	200	266	133	133
Core clock (MHz)	667	533	400	400	533	400	400
DDR Controller clock (data rate)	333	266	266	400	266	266	266
eLBC Controller clock	333	266	266	400	266	266	266

4.4.3, 4-33 Change $\overline{\text{CFG\_CLKIN\_DIV}}$ to CFG_CLKIN_DIV.

4.5.1.6, 4-38 Change the first sentence to the following:
“RCR, shown in Figure 4-12, can be used by software to initiate a hard reset sequence.”

4.5.2, 4-39 Update Output Clock Control Register (OCCR) reset value from “0x0000_FFF8” to “0x0000_FFFC” in Table 4-34, “Clock Configuration Registers Memory Map.”

4.5.2.1, 4-40 In Figure 4-14, “System PLL Mode Register” add “SVCOD” at bits 2 and 3. The updated figure is shown below:

Address 0x0_0A00 Access: Read only

	0	1	2 3	4 7	8	9 15	16 31
R	LBCM	DDRCM	SVCOD	SPMF	CKID	COREPLL	—
W							
Reset	*n*	*n*	*n* *n*	*n n n n*	*n*	*n n n n n n n*	*n n n n n n n n n n n n n n n n*[1]

Figure 4-14 System PLL Mode Register

[1] See Table 4-35 for reset values.

In Table 4-35, “System PLL Mode Register Bit Settings,” modify the row for bits 2–3 as follows:

Table 4-35. System PLL Mode Register Bit Settings

Bits	Name	Meaning	Description
2–3	SVCOD	System PLL VCO division	Section 4.3.2.1.1, “System PLL VCO Division”

4.5.2.3, 4-43 In Table 4-37, “SCCR Bit Descriptions,” replace SCCR[ENCCM] (bits 6–7) field descriptions with the following:

“Encryption core and I2C1 clock mode.

00 Encryption core clock is disabled.

01 Encryption core clock/csb_clk ratio is 1:1 (for csb_clk < 166 MHz).

10 Encryption core clock/csb_clk ratio is 1:2 (csb_clk has higher frequency than the encryption core).

11 Encryption core clock/csb_clk ratio is 1:3 (csb_clk has higher frequency than the encryption core).

The encryption core must have the same clock ratio as the USB unit, unless one of them has its clock disabled.”

4.5.2.3, 4-43 In Table 4-37, “SCCR Bit Descriptions,” modify the first sentence of the “Description” column for the “ENCCM” bit as follows:

“Encryption core, JTAG, and I^{2}C1 clock mode.”
The updated row now reads as follows:

Table 4-37. SCCR Bit Descriptions

Bits	Name	Description
6–7	ENCCM	Encryption core, JTAG, and I^{2}C1 clock mode. 00 Encryption core clock is disabled. 01 Encryption core clock/*csb_clk* ratio is 1:1 (for *csb_clk* < 166 MHz). 10 Encryption core clock/*csb_clk* ratio is 1:2 (*csb_clk* has higher frequency than the encryption core). 11 Encryption core clock/*csb_clk* ratio is 1:3 (*csb_clk* has higher frequency than the encryption core). **Note:** The encryption core must have the same clock ratio as the USB unit, unless one of them has its clock disabled.

5.3.2.5, 5-26 In Table 5-32, “SICRL Bit Settings,” change the SICRL bit settings for bit 16 and bit 17 to read as follows.

SICRL[Bits] Value		0b0/0b00	0b1/0b01	0b10	0b11
Bits	**Group**	**Pin Function 0**	**Pin Function 1**	**Pin Function 2**	**Pin Function 3**
16	GPIO1_K	GPIO1[10]	$\overline{\text{GTM1\_TGATE4/GTM2\_TGATE3}}$	—	—
17	GPIO1_J	GPIO1[9]	GTM1_TIN4/GTM2_TIN3	—	—

5.3.2.6, 5-28 In the second paragraph, change “A value of 0b11 is illegal for all groups.,” to “A value of 0b11 selects GPIO mode of the appropriate pin.”

5.3.2.6, 5-29 Change the reset value of bits 5–8 from “0000” to “1111” in Table 5-33, “SICRH Bit Settings.”

5.3.2.9, 5-33 In Figure 5-21, “DDR Debug Status Register (DDRDSR),” change the access from “Read” to “Read-only”

5.4.5.2, 5-42 Remove the following sentence from the first bullet list item:

“This is the default value after soft reset.”

Section, Page No.	**Changes**
5.5.5.5, 5-48	In Table 5-49, “RTEVR Bit Settings,” replace AIF (bit 30) bit field description with the following: “Alarm interrupt flag bit. Used to indicate the alarm interrupt. The bit is set if the RTC issues an interrupt when the RTC counter value equals RTALR[ALRM].”
5.6.5.5, 5-55	In Table 5-58, “PTEVR Bit Settings,” change bit field description of PTEVR[PIF] (bit 31) from: “Periodic interrupt flag bit. Used to indicate the periodic interrupt. It is asserted if the PIT issues an interrupt after the SPMPIT counter counts to zero. This status bit should be cleared by software.” to: “Periodic interrupt flag bit. It is asserted after the SPMPIT counter counts to zero. This status bit should be cleared by software.”
5.7.2, 5-58	Change the list bullet items from: •Maximum period of ~50 msecond (at 333-MHz bus clock and prescaler = 256) for 16-bit timer •Maximum period of ~12.8 seconds (at 333-MHz bus clock and prescaler = 256) for 32-bit timer to: •Maximum period of ~206 seconds (at 333-MHz bus clock in slow go mode, primary and secondary prescaler = 256) for 16-bit timer •Maximum period of ~3298 seconds (at 333-MHz bus clock and prescaler = 256) for 32-bit timer
5.7.4, 5-61	In Table 5-60, “GTM External Signals—Detailed Signal Descriptions,” change the State Meaning description of signal TGATEn, from “In a reset gate mode...” to “In a restart gate mode...”. In addition, for the State Meaning description of signal TOUTn, change (list #2) from: “TOUTn changes occur on the rising edge of the system clock” to: “TOUTn begins or stops counting, depending on the signal state and the configured mode.”, and for TOUTn signal Timing, change “system clock” to “timer input clock.”
5.7.6.1, 5-70	In the second paragraph, following the bulleted list, change “65,537” to “65,536.”

Section, Page No.	Changes
5.7.6.1, 5-70	Change the last sentence in the second paragraph from: “The maximum period (when the reference value is all ones and the prescaler divides by 256) for one 16-bit timer is ~50 ms at 333 MHz to: “The maximum period (when the reference value is all ones and the prescaler divides by 256) for one 16-bit timer is ~206 s at 333 MHz.”
5.7.6.3, 5-71	Replace the Note with the following: **NOTE** $\overline{\text{TGATE}}$ is internally synchronized to the system clock. If $\overline{\text{TGATE}}$ meets the asynchronous input setup time, the counter begins or stops counting after one system clock when working with the internal clock.
6.2.5, 6-7	Change the following from: “Arbiter mask register (AMR) is used to mask interrupts or reset requests. Setting a mask bit enables the corresponding interrupt or reset request; clearing a bit masks it. Regular interrupts, MCP interrupts and reset requests can be masked by AMR register.” to: “Arbiter mask register (AMR) is used to mask interrupts. Setting a mask bit enables the corresponding interrupt; clearing a bit masks it. Regular interrupts and MCP interrupts can be masked by the AMR register, except for scenarios that results in a transfer error when the master ID is 0.”
6.2.6, 6-8	Change the access mentioned in Figure 6-6, “Arbiter Event Attributes Register (AEATR),” from “User Read/ Write” to “User Read-only.”
6.2.7, 6-10	Change the access mentioned in Figure 6-7, “Arbiter Event Address Register (AEADR),” from “User Read/ Write” to “User Read-only.”
6.3.1.3, 6-14	Change the last sentence in the paragraph from: “After the completion of snoop copyback, the arbiter grants the bus back to the master that had its transaction ARTRYed” to: “After the completion of snoop copyback, the arbiter grants the bus to the most ahead master among those masters which have an active bus request signal at that time, which may or may not be the same master that had its transaction ARTRYed. Only when a transaction address phase is completed with no ARTRY (and no repeat conditions), the master moves to the end of the line.”
7.4.1.3.3, 7-22	In Table 7-3, “e300 HID0 Bit Descriptions,” add the following note to EBA (bit 2) and EBD (bit 3) field descriptions: “Do not set this bit; the CSB does not have parity signals.”

Section, Page No.	Changes
8.5.13, 8-23	In Figure 8-16, “System External Interrupt Mask Register (SEMSR),” change the following text appearing under the reset values of 0–15 bits from: “The reset values of implemented bits reflect the values of the external IRQ signals. Reserved bits are zeros.” to “All zeros”. In addition, delete the second footnote that says: “The user should drive all IRQ inputs to an inactive state prior to reset negation”. The updated figure is shown below:

Offset 0x38

Access: Read/write

	0	1	2	3	4	5	6	7	8–15
R W	IRQ0[1]	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7	—
Reset	All zeros								

	16	17–31
R W	SIRQ0	—
Reset	All zeros	

[1] This bit is valid only if the IRQ0 signal is configured as an external maskable interrupt (SEMSR[SIRQ0] = 0)

Figure 8-16 System External Interrupt Mask Register (SEMSR)

Section, Page No.	Changes
9.3.2.1, 9-6	Change the signal description of MA[14:0] in Table 9-3, “Memory Interface Signals—Detailed Signal Descriptions” from: “Assertion/Negation—The address is always driven when the memory controller is enabled. It is valid when a transaction is driven to DRAM (when $\overline{\text{MCS}n}$ is active).” to: “Assertion/Negation—The address lines are only driven when the controller has a command scheduled to issue on the address/CMD bus; otherwise they will be at high-Z. It is valid when a transaction is driven to DRAM (when $\overline{\text{MCS}n}$ is active).”
9.5, 9-38	Change the second sentence in the third paragraph from: “Bank sizes up to 2 Gbytes (maximum total physical memory size of 4 Gbytes) are supported, providing up to a maximum of 4 Gbytes of DDR main memory.” to: “Bank sizes up to 2 Gbits (maximum total physical memory size of 4 Gbytes) are supported, providing up to a maximum of 4 Gbits of DDR main memory per chip select.”
9.5.12, 9-65	Replace the first sentence of the third paragraph to say the following: “If a multi-bit error is detected for a read, the DDR memory controller logs the error and generates the interrupt, and transfer error acknowledge (TEA) is asserted internally on the CSB bus (if enabled, as described in Section 9.4.1.25, “Memory Error Disable (ERR_DISABLE)”).”
Chapter 9	Replace Chapter 9, “DDR Memory Controller” with the following updated chapter:

Chapter 9
DDR Memory Controller

9.1 Introduction

The fully programmable DDR SDRAM controller supports most JEDEC standard ×8, ×16, ×32, or ×64 DDR2 and DDR memories available. In addition, unbuffered and registered DIMMs are supported. However, mixing different memory types or unbuffered and registered DIMMs in the same system is not supported. Built-in error checking and correction (ECC) ensures very low bit-error rates for reliable high-frequency operation. Dynamic power management and auto-precharge modes simplify memory system design. A large set of special features, including ECC error injection, support rapid system debug.

NOTE

In this chapter, the word ‘bank’ refers to a physical bank specified by a chip select; ‘logical bank’ refers to one of the four or eight sub-banks in each SDRAM chip. A sub-bank is specified by the 2 or 3 bits on the bank address (MBA) pins during a memory access.

Figure 9-1 is a high-level block diagram of the DDR memory controller with its associated interfaces. Section 9.5, “Functional Description,” contains detailed figures of the controller.

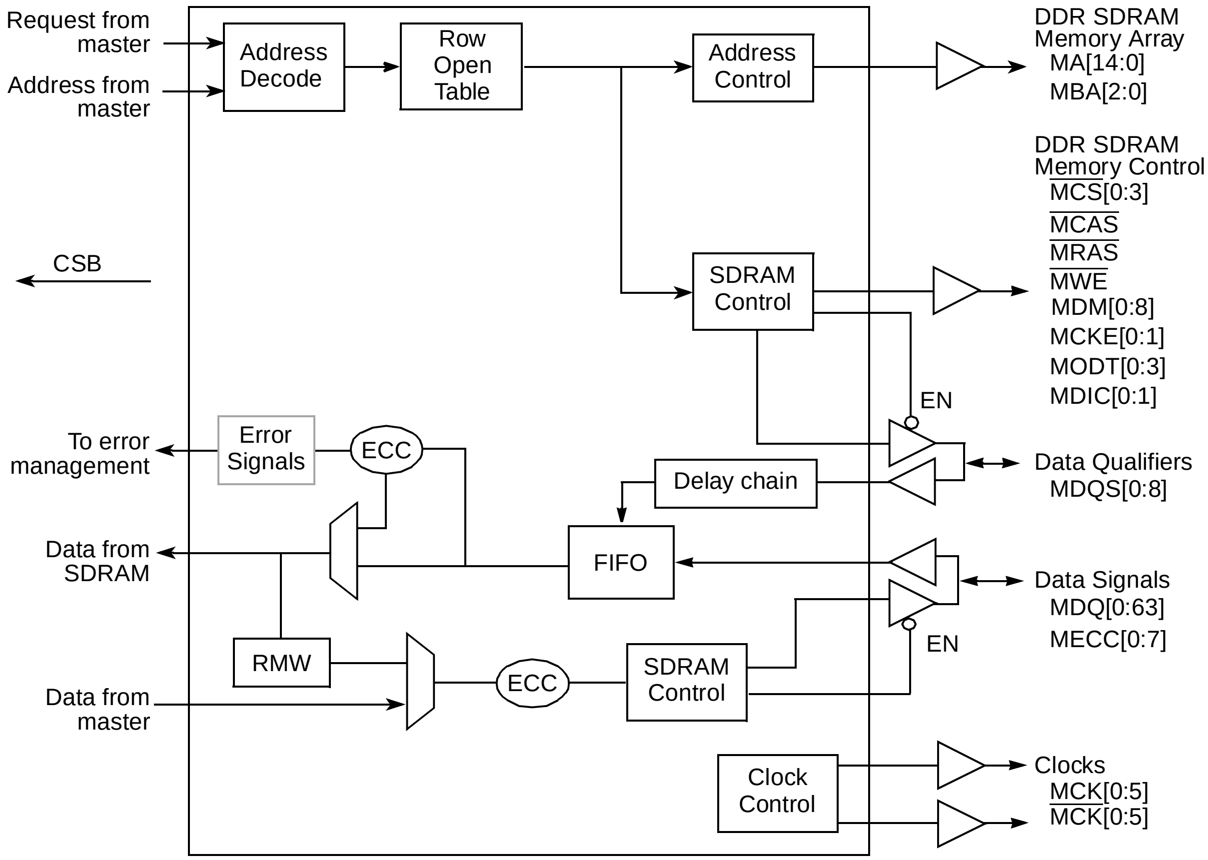


Figure 9-1. DDR Memory Controller Simplified Block Diagram

9.2 Features

The DDR memory controller includes these distinctive features:

- Support for DDR2 and DDR SDRAM
- 64-/72-bit SDRAM data bus, for DDR and DDR2
- Programmable settings for meeting all SDRAM timing parameters
- Support for the following SDRAM configurations:
  - As many as four physical banks (chip selects), each bank independently addressable
    - 64-Mbit to -Gbit devices depending on internal device configuration with ×8/×16/×32 data ports (no direct ×4 support)
  - Unbuffered and registered DIMMs
- Chip select interleaving support
- Support for data mask signals and read-modify-write for sub-double-word writes. Note that a read-modify-write sequence is only necessary when ECC is enabled.

- Support for double-bit error detection and single-bit error correction ECC (8-bit check word across or 64-bit data)
- Open page management (dedicated entry for each logical bank)
- Automatic DRAM initialization sequence or software-controlled initialization sequence
- Automatic DRAM data initialization
- Support for up to eight posted refreshes
- Memory controller clock frequency of two times the SDRAM clock with support for sleep power management
- Support for error injection

9.2.1 Modes of Operation

The DDR memory controller supports the following modes:

- Dynamic power management mode. The DDR memory controller can reduce power consumption by negating the SDRAM CKE signal when no transactions are pending to the SDRAM.
- Auto-precharge mode. Clearing DDR_SDRAM_INTERVAL[BSTOPRE] causes the memory controller to issue an auto-precharge command with every read or write transaction. Auto-precharge mode can be enabled for separate chip selects by setting CS*n*_CONFIG[AP_*n*_EN].

9.3 External Signal Descriptions

This section provides descriptions of the DDR memory controller's external signals. It describes each signal's behavior when the signal is asserted or negated and when the signal is an input or an output.

9.3.1 Signals Overview

Memory controller signals are grouped as follows:

- Memory interface signals
- Clock signals
- Debug signals

Table 9-1 shows how DDR memory controller external signals are grouped. The device hardware specification has a pinout diagram showing pin numbers. It also lists all electrical and mechanical specifications.

Table 9-1. DDR Memory Interface Signal Summary

Name	Function/Description	Reset	Pins	I/O
MDQ[0:63]	Data bus	All zeros	64	I/O
MDQS[0:8]	Data strobes	All zeros	9	I/O
MECC[0:7]	Error checking and correcting	All zeros	8	I/O

Table 9-1. DDR Memory Interface Signal Summary (continued)

Name	Function/Description	Reset	Pins	I/O
$\overline{\text{MCAS}}$	Column address strobe	One	1	O
MA[14:0]	Address bus	All zeros	15	O
MBA[2:0]	Logical bank address	All zeros	3	O
$\overline{\text{MCS}}$[0:3]	Chip selects	All ones	4	O
$\overline{\text{MWE}}$	Write enable	One	1	O
$\overline{\text{MRAS}}$	Row address strobe	One	1	O
MDM[0:8]	Data mask	All zeros	9	O
MCK[0:5]	DRAM clock outputs	All zeros	6	O
$\overline{\text{MCK}}$[0:5]	DRAM clock outputs (complement)	All zeros	6	O
MCKE[0:1]	DRAM clock enable	All zeros	2	O
MODT[0:3]	DRAM on-die termination external control.	All zeros	4	O
MDVAL	Memory debug data valid	Zero	1	O
MSRCID[0:4]	Memory debug source ID	All zeros	5	O
MDIC[0:1]	Driver impedance calibration	High Z	2	I/O

Table 9-2 shows the memory address signal mappings.

Table 9-2. Memory Address Signal Mappings

Signal Name (Outputs)		JEDEC DDR DIMM Signals (Inputs)
msb	MA14	A14
	MA13	A13
	MA12	A12
	MA11	A11
	MA10	A10 (AP for DDR)[1]
	MA9	A9
	MA8	A8 (alternate AP for DDR)[2]
	MA7	A7
	MA6	A6
	MA5	A5
	MA4	A4
	MA3	A3
	MA2	A2
	MA1	A1
lsb	MA0	A0
msb	MBA2	MBA2

Table 9-2. Memory Address Signal Mappings (continued)

Signal Name (Outputs)		JEDEC DDR DIMM Signals (Inputs)
	MBA1	MBA1
lsb	MBA0	MBA0

[1] Auto-precharge for DDR signaled on A10 when DDR_SDRAM_CFG[PCHB8] = 0.

[2] Auto-precharge for DDR signaled on A8 when DDR_SDRAM_CFG[PCHB8] = 1.

9.3.2 Detailed Signal Descriptions

The following sections describe the DDR SDRAM controller input and output signals, the meaning of their different states, and relative timing information for assertion and negation.

9.3.2.1 Memory Interface Signals

Table 9-3 describes the DDR controller memory interface signals.

Table 9-3. Memory Interface Signals—Detailed Signal Descriptions

Signal	I/O	Description	
MDQ[0:63]	I/O	Data bus. Both input and output signals on the DDR memory controller.	
	O	As outputs for the bidirectional data bus, these signals operate as described below.	
		State Meaning	Asserted/Negated—Represent the value of data being driven by the DDR memory controller.
		Timing	Assertion/Negation—Driven coincident with corresponding data strobes (MDQS) signal. High impedance—No READ or WRITE command is in progress; data is not being driven by the memory controller or the DRAM.
	I	As inputs for the bidirectional data bus, these signals operate as described below.	
		State Meaning	Asserted/Negated—Represents the state of data being driven by the external DDR SDRAMs.
		Timing	Assertion/Negation—The DDR SDRAM drives data during a READ transaction. High impedance—No READ or WRITE command in progress; data is not being driven by the memory controller or the DRAM.

Table 9-3. Memory Interface Signals—Detailed Signal Descriptions (continued)

Signal	I/O	Description	
MDQS[0:8]	I/O	Data strobes. Inputs with read data, outputs with write data.	
	O	As outputs, the data strobes are driven by the DDR memory controller during a write transaction. The memory controller always drives these signals low unless a read has been issued and incoming data strobes are expected. This keeps the data strobes from floating high when there are no transactions on the DRAM interface.	
		State Meaning	Asserted/Negated—Driven high when positive capture data is transmitted and driven low when negative capture data is transmitted. Centered in the data “eye” for writes; coincident with the data eye for reads. Treated as a clock. Data is valid when signals toggle. See Table 9-36 for byte lane assignments.
		Timing	Assertion/Negation (for DDR1)—If a WRITE command is registered at clock edge *n*, data strobes at the DRAM assert centered in the data eye on clock edge *n* + 1. See the JEDEC DDR SDRAM specification for more information. Assertion/Negation (for DDR2)—If a WRITE command is registered at clock edge *n*, and the latency is programmed in TIMING_CFG_2[WR_LAT] to be *m* clocks, data strobes at the DRAM assert coincident with the data on clock edge *n* + *m*. See the JEDEC DDR SDRAM specification for more information.
	I	As inputs, the data strobes are driven by the external DDR SDRAMs during a read transaction. The data strobes are used by the memory controller to synchronize data latching.	
		State Meaning	Asserted/Negated—Driven high when positive capture data is received and driven low when negative capture data is received. Centered in the data eye for writes; coincident with the data eye for reads. Treated as a clock. Data is valid when signals toggle. See Table 9-36 for byte lane assignments.
		Timing	Assertion/Negation—If a READ command is registered at clock edge *n*, and the latency is programmed in TIMING_CFG_1[CASLAT] to be *m* clocks, data strobes at the DRAM assert coincident with the data on clock edge *n* + *m*. See the JEDEC DDR SDRAM specification for more information.
MECC[0:7]	I/O	Error checking and correcting codes. Input and output signals for the DDR controller’s bidirectional ECC bus. MECC[0:5] function in both normal and debug modes.	
	O	As normal mode outputs, the ECC signals represent the state of ECC driven by the DDR controller on writes. As debug mode outputs MECC[0:5] provide source ID and data-valid information. See Section 9.5.10, “Error Checking and Correcting (ECC),” for more details.	
		State Meaning	Asserted/Negated—Represents the state of ECC being driven by the DDR controller on writes.
		Timing	Assertion/Negation—Same timing as MDQ High impedance—Same timing as MDQ
	I	As inputs, the ECC signals represent the state of ECC driven by the SDRAM devices on reads.	
		State Meaning	Asserted/Negated—Represents the state of ECC being driven by the DDR SDRAMs on reads.
		Timing	Assertion/Negation—Same timing as MDQ High impedance—Same timing as MDQ

Table 9-3. Memory Interface Signals—Detailed Signal Descriptions (continued)

Signal	I/O	Description	
MA[14:0]	O	Address bus. Memory controller outputs for the address to the DRAM. MA[14:0] carry 15 of the address bits for the DDR memory interface corresponding to the row and column address bits. MA0 is the lsb of the address output from the memory controller.	
		State Meaning	Asserted/Negated—Represents the address driven by the DDR memory controller. Contains different portions of the address depending on the memory size and the DRAM command being issued by the memory controller. See Table 9-40 Table 9-41 for a complete description of the mapping of these signals.
		Timing	Assertion/Negation—The address lines are only driven when the controller has a command scheduled to issue on the address/CMD bus; otherwise they will be at high-Z. It is valid when a transaction is driven to DRAM (when $\overline{\text{MCS}n}$ is active). High impedance—When the memory controller is disabled
MBA[2:0]	O	Logical bank address. Outputs that drive the logical (or internal) bank address pins of the SDRAM. Each SDRAM supports four or eight addressable logical sub-banks. Bit zero of the memory controller’s output bank address must be connected to bit zero of the SDRAM’s input bank address. MBA0, the least-significant bit of the three bank address signals, is asserted during the mode register set command to specify the extended mode register.	
		State Meaning	Asserted/Negated—Selects the DDR SDRAM logical (or internal) bank to be activated during the row address phase and selects the SDRAM internal bank for the read or write operation during the column address phase of the memory access. Table 9-40 Table 9-41 describes the mapping of these signals in all cases.
		Timing	Assertion/Negation—Same timing as MA*n* High impedance—Same timing as MA*n*
$\overline{\text{MCAS}}$	O	Column address strobe. Active-low SDRAM address multiplexing signal. $\overline{\text{MCAS}}$ is asserted for read or write transactions and for mode register set, refresh, and precharge commands.	
		State Meaning	Asserted—Indicates that a valid SDRAM column address is on the address bus for read and write transactions. See Table 9-48 for more information on the states required on $\overline{\text{MCAS}}$ for various other SDRAM commands. Negated—The column address is not guaranteed to be valid.
		Timing	Assertion/Negation—Assertion and negation timing is directed by the values described in Section 9.4.1.4, “DDR SDRAM Timing Configuration 0 (TIMING_CFG_0),” Section 9.4.1.5, “DDR SDRAM Timing Configuration 1 (TIMING_CFG_1),” Section 9.4.1.6, “DDR SDRAM Timing Configuration 2 (TIMING_CFG_2),” and Section 9.4.1.3, “DDR SDRAM Timing Configuration 3 (TIMING_CFG_3).” High impedance—$\overline{\text{MCAS}}$ is always driven unless the memory controller is disabled.
$\overline{\text{MRAS}}$	O	Row address strobe. Active-low SDRAM address multiplexing signal. Asserted for activate commands. In addition; used for mode register set commands and refresh commands.	
		State Meaning	Asserted—Indicates that a valid SDRAM row address is on the address bus for read and write transactions. See Table 9-48 for more information on the states required on $\overline{\text{MRAS}}$ for various other SDRAM commands. Negated—The row address is not guaranteed to be valid.
		Timing	Assertion/Negation—Assertion and negation timing is directed by the values described in Section 9.4.1.4, “DDR SDRAM Timing Configuration 0 (TIMING_CFG_0),” Section 9.4.1.5, “DDR SDRAM Timing Configuration 1 (TIMING_CFG_1),” Section 9.4.1.6, “DDR SDRAM Timing Configuration 2 (TIMING_CFG_2),” and Section 9.4.1.3, “DDR SDRAM Timing Configuration 3 (TIMING_CFG_3).” High impedance—$\overline{\text{MRAS}}$ is always driven unless the memory controller is disabled.

Table 9-3. Memory Interface Signals—Detailed Signal Descriptions (continued)

Signal	I/O	Description	
$\overline{\text{MCS}}$[0:3]	O	Chip selects. Four chip selects supported by the memory controller.	
		State Meaning	Asserted—Selects a physical SDRAM bank to perform a memory operation as described in Section 9.4.1.1, “Chip Select Memory Bounds (CSn_BNDS),” and Section 9.4.1.2, “Chip Select Configuration (CSn_CONFIG).” The DDR controller asserts one of the $\overline{\text{MCS}}$[0:3] signals to begin a memory cycle. Negated—Indicates no SDRAM action during the current cycle.
		Timing	Assertion/Negation—Asserted to signal any new transaction to the SDRAM. The transaction must adhere to the timing constraints set in TIMING_CFG_0–TIMING_CFG_3. High impedance—Always driven unless the memory controller is disabled.
$\overline{\text{MWE}}$	O	Write enable. Asserted when a write transaction is issued to the SDRAM. This is also used for mode registers set commands and precharge commands.	
		State Meaning	Asserted—Indicates a memory write operation. See Table 9-48 for more information on the states required on $\overline{\text{MWE}}$ for various other SDRAM commands. Negated—Indicates a memory read operation.
		Timing	Assertion/Negation—Similar timing as $\overline{\text{MRAS}}$ and $\overline{\text{MCAS}}$. Used for write commands. High impedance—$\overline{\text{MWE}}$ is always driven unless the memory controller is disabled.
MDM[0:8]	O	DDR SDRAM data output mask. Masks unwanted bytes of data transferred during a write. They are needed to support sub-burst-size transactions (such as single-byte writes) on SDRAM where all I/O occurs in multi-byte bursts. MDM0 corresponds to the most significant byte (MSB) and MDM7 corresponds to the LSB, while MDM8 corresponds to the ECC byte. Table 9-36 shows byte lane encodings.	
		State Meaning	Asserted—Prevents writing to DDR SDRAM. Asserted when data is written to DRAM if the corresponding byte(s) should be masked for the write. Note that the MDM*n* signals are active-high for the DDR controller. MDM*n* is part of the DDR command encoding. Negated—Allows the corresponding byte to be read from or written to the SDRAM.
		Timing	Assertion/Negation—Same timing as MDQx as outputs. High impedance—Always driven unless the memory controller is disabled.
MODT[0:3]	O	On-Die termination. Memory controller outputs for the ODT to the DRAM. MODT[0:3] represents the on-die termination for the associated data, data masks, ECC, and data strobes.	
		State Meaning	Asserted/Negated—Represents the ODT driven by the DDR memory controller.
		Timing	Assertion/Negation—Driven in accordance with JEDEC DRAM specifications for on-die termination timings. It is configured through the CS*n*_CONFIG[ODT_RD_CFG] and CS*n*_CONFIG[ODT_WR_CFG] fields. High impedance—Always driven.
MDIC[0:1]	I/O	Driver impedance calibration. Note that the MDIC signals require the use of resistors; MDIC0 must be pulled to GND, while MDIC1 must be pulled to GV_{DD}. See Section 5.3.2.8, “DDR Control Driver Register (DDRCDR),” for more information on these signals.	
		State Meaning	These pins are used for automatic calibration of the DDR IOs.
		Timing	These are driven for four DRAM cycles at a time while the DDR controller is executing the automatic driver compensation.

9.3.2.2 Clock Interface Signals

Table 9-4 contains the detailed descriptions of the clock signals of the DDR controller.

Table 9-4. Clock Signals—Detailed Signal Descriptions

Signal	I/O	Description	
MCK[0:5], $\overline{\text{MCK}}$[0:5]	O	DRAM clock outputs and their complements.	
		State Meaning	Asserted/Negated—The JEDEC DDR SDRAM specifications require true and complement clocks. A clock edge is seen by the SDRAM when the true and complement cross.
		Timing	Assertion/Negation—Timing is controlled by the DDR_CLK_CNTL register at offset 0x130.
MCKE[0:1]	O	Clock enable. Output signals used as the clock enables to the SDRAM. MCKE[0:1] can be negated to stop clocking the DDR SDRAM. The MCKE signals should be connected to the same rank of memory as the corresponding $\overline{\text{MCS}}$ and MODT signals. For example, MCKE[0] should be connected to the same rank of memory as $\overline{\text{MCS}}$[0] and MODT[0].	
		State Meaning	Asserted—Clocking to the SDRAM is enabled. Negated—Clocking to the SDRAM is disabled and the SDRAM should ignore signal transitions on MCK or $\overline{\text{MCK}}$. MCK/$\overline{\text{MCK}}$ are don't cares while MCKE[0:1] are negated.
		Timing	Assertion/Negation—Asserted when DDR_SDRAM_CFG[MEM_EN] is set. Can be negated when entering dynamic power management or self refresh. Are asserted again when exiting dynamic power management or self refresh. High impedance—Always driven.

9.3.2.3 Debug Signals

The debug signals MSRCID[0:4] and MDVAL have no function in normal DDR controller operation. A detailed description of these signals can be found in Section 5.3.2.7, "Debug Configuration."

9.4 Memory Map/Register Definition

Table 9-5 shows the register memory map for the DDR memory controller.

In this table and in the register figures and field descriptions, the following access definitions apply:

- Reserved fields are always ignored for the purposes of determining access type.
- R/W, R, and W (read/write, read only, and write only) indicate that all the non-reserved fields in a register have the same access type.
- w1c indicates that all of the non-reserved fields in a register are cleared by writing ones to them.
- Mixed indicates a combination of access types.
- Special is used when no other category applies. In this case the register figure and field description table should be read carefully.

Table 9-5. DDR Memory Controller Memory Map

Offset	Register	Access	Reset	Section/Page
DDR Memory Controller—Block Base Address 0x0_2000				
0x000	CS0_BNDS—Chip select 0 memory bounds	R/W	All zeros	9.4.1.1/9-18
0x008	CS1_BNDS—Chip select 1 memory bounds	R/W	All zeros	9.4.1.1/9-18

Table 9-5. DDR Memory Controller Memory Map (continued)

Offset	Register	Access	Reset	Section/Page
0x010	CS2_BNDS—Chip select 2 memory bounds	R/W	All zeros	9.4.1.1/9-18
0x018	CS3_BNDS—Chip select 3 memory bounds	R/W	All zeros	9.4.1.1/9-18
0x080	CS0_CONFIG—Chip select 0 configuration	R/W	All zeros	9.4.1.2/9-18
0x084	CS1_CONFIG—Chip select 1 configuration	R/W	All zeros	9.4.1.2/9-18
0x088	CS2_CONFIG—Chip select 2 configuration	R/W	All zeros	9.4.1.2/9-18
0x08C	CS3_CONFIG—Chip select 3 configuration	R/W	All zeros	9.4.1.2/9-18
0x100	TIMING_CFG_3—DDR SDRAM timing configuration 3	R/W	All zeros	9.4.1.3/9-20
0x104	TIMING_CFG_0—DDR SDRAM timing configuration 0	R/W	0x0011_0105	9.4.1.4/9-21
0x108	TIMING_CFG_1—DDR SDRAM timing configuration 1	R/W	All zeros	9.4.1.5/9-22
0x10C	TIMING_CFG_2—DDR SDRAM timing configuration 2	R/W	All zeros	9.4.1.6/9-24
0x110	DDR_SDRAM_CFG—DDR SDRAM control configuration	R/W	0x0200_0000	9.4.1.7/9-26
0x114	DDR_SDRAM_CFG_2—DDR SDRAM control configuration 2	R/W	All zeros	9.4.1.8/9-29
0x118	DDR_SDRAM_MODE—DDR SDRAM mode configuration	R/W	All zeros	9.4.1.9/9-31
0x11C	DDR_SDRAM_MODE_2—DDR SDRAM mode configuration 2	R/W	All zeros	9.4.1.10/9-32
0x120	DDR_SDRAM_MD_CNTL—DDR SDRAM mode control	R/W	All zeros	9.4.1.11/9-32
0x124	DDR_SDRAM_INTERVAL—DDR SDRAM interval configuration	R/W	All zeros	9.4.1.12/9-35
0x128	DDR_DATA_INIT—DDR SDRAM data initialization	R/W	All zeros	9.4.1.13/9-35
0x130	DDR_SDRAM_CLK_CNTL—DDR SDRAM clock control	R/W	0x0200_0000	9.4.1.14/9-36
0x140–0x144	Reserved	—	—	—
0x148	DDR_INIT_ADDR—DDR training initialization address	R/W	All zeros	9.4.1.15/9-36
0x150–0xBF4	Reserved	—	—	—
0xBF8	DDR_IP_REV1—DDR IP block revision 1	R	0x*nnnn_nnnn*[1]	9.4.1.16/9-37
0xBFC	DDR_IP_REV2—DDR IP block revision 2	R	0x00*nn*_00*nn*[1]	9.4.1.17/9-37
0xE00	DATA_ERR_INJECT_HI—Memory data path error injection mask high	R/W	All zeros	9.4.1.18/9-38
0xE04	DATA_ERR_INJECT_LO—Memory data path error injection mask low	R/W	All zeros	9.4.1.19/9-38
0xE08	ERR_INJECT—Memory data path error injection mask ECC	R/W	All zeros	9.4.1.20/9-39
0xE20	CAPTURE_DATA_HI—Memory data path read capture high	R/W	All zeros	9.4.1.21/9-39
0xE24	CAPTURE_DATA_LO—Memory data path read capture low	R/W	All zeros	9.4.1.22/9-40
0xE28	CAPTURE_ECC—Memory data path read capture ECC	R/W	All zeros	9.4.1.23/9-40
0xE40	ERR_DETECT—Memory error detect	w1c	All zeros	9.4.1.24/9-40
0xE44	ERR_DISABLE—Memory error disable	R/W	All zeros	9.4.1.25/9-41
0xE48	ERR_INT_EN—Memory error interrupt enable	R/W	All zeros	9.4.1.26/9-42
0xE4C	CAPTURE_ATTRIBUTES—Memory error attributes capture	R/W	All zeros	9.4.1.27/9-43
0xE50	CAPTURE_ADDRESS—Memory error address capture	R/W	All zeros	9.4.1.28/9-44
0xE58	ERR_SBE—Single-Bit ECC memory error management	R/W	All zeros	9.4.1.29/9-44

---

[1] Implementation-dependent reset values are listed in specified section/page.

9.4.1 Register Descriptions

This section describes the DDR memory controller registers. Shading indicates reserved fields that should not be written.

9.4.1.1 Chip Select Memory Bounds (CS*n*_BNDS)

The chip select bounds registers (CS*n*_BNDS) define the starting and ending address of the memory space that corresponds to the individual chip selects. Note that the size specified in CS*n*_BNDS should equal the size of physical DRAM. Also, note that EA*n* must be greater than or equal to SA*n*.

If chip select interleaving is enabled, all fields in the lower interleaved chip select are used, and the other chip selects' bounds registers are unused. For example, if chip selects 0 and 1 are interleaved, all fields in CS0_BNDS are used, and all fields in CS1_BNDS are unused.

CS*n*_BNDS are shown in Figure 9-2.

Offset 0x000, 0x008, 0x010, 0x018 Access: Read/Write

	0–7	8–15	16–23	24–31
R W	—	SA*n*	—	EA*n*
Reset	All zeros			

Figure 9-2. Chip Select Bounds Registers (CS*n*_BNDS)

Table 9-6 describes the CS*n*_BNDS register fields.

Table 9-6. CS*n*_BNDS Field Descriptions

Bits	Name	Description
0–7	—	Reserved
8–15	SA*n*	Starting address for chip select (bank) *n.* This value is compared against the 8 msbs of the 32-bit address.
16–23	—	Reserved
24–31	EA*n*	Ending address for chip select (bank) *n.* This value is compared against the 8 msbs of the 32-bit address.

9.4.1.2 Chip Select Configuration (CS*n*_CONFIG)

The chip select configuration (CS*n*_CONFIG) registers shown in Figure 9-3 enable the DDR chip selects and set the number of row and column bits used for each chip select. These registers should be loaded with the correct number of row and column bits for each SDRAM. Because CS*n*_CONFIG[ROW_BITS_CS_*n*, COL_BITS_CS_*n*] establish address multiplexing, the user should take great care to set these values correctly.

If chip select interleaving is enabled, then all fields in the lower interleaved chip select are used, and the other registers' fields are unused, with the exception of the ODT_RD_CFG and ODT_WR_CFG fields. For example, if chip selects 0 and 1 are interleaved, all fields in CS0_CONFIG are used, but only the ODT_RD_CFG and ODT_WR_CFG fields in CS1_CONFIG are used.

Offset 0x080, 0x084, 0x088, 0x08C Access: Read/Write

	0	1–7	8	9–11	12	13–15
R W	CS_*n*_EN	—	AP_*n*_EN	ODT_RD_CFG	—	ODT_WR_CFG
Reset	All zeros					

	16–17	18–20	21–23	24–28	29–31
R W	BA_BITS_CS_*n*	—	ROW_BITS_CS_*n*	—	COL_BITS_CS_*n*
Reset	All zeros				

Figure 9-3. Chip Select Configuration Register (CS*n*_CONFIG)

Table 9-7 describes the CS*n*_CONFIG register fields.

Table 9-7. CS*n*_CONFIG Field Descriptions

Bits	Name	Description
0	CS_*n*_EN	Chip select *n* enable 0 Chip select *n* is not active 1 Chip select *n* is active and assumes the state set in CS*n*_BNDS.
1–7	—	Reserved
8	AP_*n*_EN	Chip select *n* auto-precharge enable 0 Chip select *n* is only auto-precharged if global auto-precharge mode is enabled (DDR_SDRAM_INTERVAL[BSTOPRE] = 0). 1 Chip select *n* always issues an auto-precharge for read and write transactions.
9–11	ODT_RD_CFG	ODT for reads configuration. Note that CAS latency plus additive latency must be at least 3 cycles for ODT_RD_CFG to be enabled. ODT should only be used with DDR2 memories. 000 Never assert ODT for reads 001 Assert ODT only during reads to CS*n* 010 Assert ODT only during reads to other chip selects 011 Assert ODT only during reads to other DIMM modules. It is assumed that CS0 and CS1 are on the same DIMM module, whereas CS2 and CS3 are on a separate DIMM module. 100 Assert ODT for all reads 101–111Reserved
12	—	Reserved
13–15	ODT_WR_CFG	ODT for writes configuration. Note that write latency plus additive latency must be at least 3 cycles for ODT _WR_CFG to be enabled. ODT should only be used with DDR2 memories. 000 Never assert ODT for writes 001 Assert ODT only during writes to CS*n* 010 Assert ODT only during writes to other chip selects 011 Assert ODT only during writes to other DIMM modules. It is assumed that CS0 and CS1 are on the same DIMM module, whereas CS2 and CS3 are on a separate DIMM module. 100 Assert ODT for all writes 101–111Reserved
16–17	BA_BITS_CS_*n*	Number of bank bits for SDRAM on chip select *n*. These bits correspond to the sub-bank bits driven on MBA*n* in Table 9-43. 00 2 logical bank bits 01 3 logical bank bits 10–11Reserved

Table 9-7. CS*n*_CONFIG Field Descriptions (continued)

Bits	Name	Description
18–20	—	Reserved
21–23	ROW_BITS_CS_*n*	Number of row bits for SDRAM on chip select *n*. See Table 9-43 for details. 000 12 row bits 001 13 row bits 010 14 row bits 011 15 row bits 000–111 Reserved
24–28	—	Reserved
29–31	COL_BITS_CS_*n*	Number of column bits for SDRAM on chip select *n*. For DDR, the decoding is as follows: 000 8 column bits 001 9 column bits 010 10 column bits 011 11 column bits 100–111 Reserved

9.4.1.3 DDR SDRAM Timing Configuration 3 (TIMING_CFG_3)

DDR SDRAM timing configuration register 3, shown in Figure 9-4, sets the extended refresh recovery time, which is combined with TIMING_CFG_1[REFREC] to determine the full refresh recovery time.

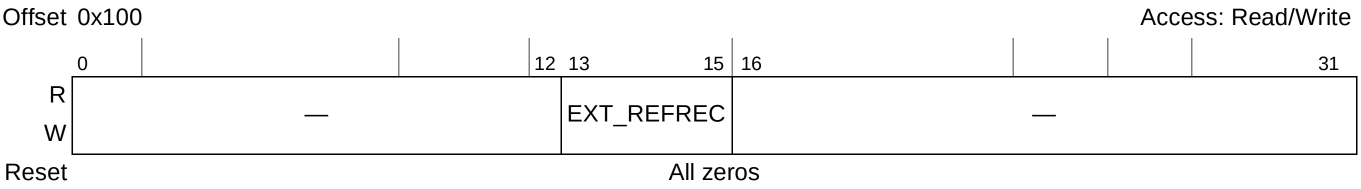


Figure 9-4. DDR SDRAM Timing Configuration 3 (TIMING_CFG_3)

Table 9-8 describes TIMING_CFG_3 fields.

Table 9-8. TIMING_CFG_3 Field Descriptions

Bits	Name	Description
0–12	—	Reserved
13–15	EXT_REFREC	Extended refresh recovery time (t_{RFC}). Controls the number of clock cycles from a refresh command until an activate command is allowed. This field is concatenated with TIMING_CFG_1[REFREC] to obtain a 7-bit value for the total refresh recovery. Note that hardware adds an additional 8 clock cycles to the final, 7-bit value of the refresh recovery. t_{RFC} = {EXT_REFREC || REFREC} + 8, such that t_{RFC} is calculated as follows: 000 0 clocks 001 16 clocks 010 32 clocks 011 48 clocks 100 64 clocks 101 80 clocks 110 96 clocks 111 112 clocks
16–31	—	Reserved

9.4.1.4 DDR SDRAM Timing Configuration 0 (TIMING_CFG_0)

DDR SDRAM timing configuration register 0, shown in Figure 9-5, sets the number of clock cycles between various SDRAM control commands.

Offset 0x104 Access: Read/Write

	0 1	2 3	4 5	6 7	8	9 11	12	13 15	16 19	20 23	24 27	28 31
R W	RWT	WRT	RRT	WWT	—	ACT_PD_EXIT	—	PRE_PD_EXIT	—	ODT_PD_EXIT	—	MRS_CYC
Reset	0 0	0 0	0 0	0 0	0	0 0 1	0	0 0 1	0 0 0 0	0 0 0 1	0 0 0 0	0 1 0 1

Figure 9-5. DDR SDRAM Timing Configuration 0 (TIMING_CFG_0)

Table 9-9 describes TIMING_CFG_0 fields.

Table 9-9. TIMING_CFG_0 Field Descriptions

Bits	Name	Description
0–1	RWT	Read-to-write turnaround (t_{RTW}). Specifies how many extra cycles are added between a read to write turnaround. If 0 clocks is chosen, then the DDR controller uses a fixed number based on the CAS latency and write latency. Choosing a value other than 0 adds extra cycles past this default calculation. As a default the DDR controller determines the read-to-write turnaround as CL – WL + BL ÷ 2 + 2. In this equation, CL is the CAS latency rounded up to the next integer, WL is the programmed write latency, and BL is the burst length. 00 0 clocks 01 1 clock 10 2 clocks 11 3 clocks
2–3	WRT	Write-to-read turnaround. Specifies how many extra cycles are added between a write to read turnaround. If 0 clocks is chosen, then the DDR controller uses a fixed number based on the, read latency, and write latency. Choosing a value other than 0 adds extra cycles past this default calculation. As a default, the DDR controller determines the write-to-read turnaround as WL – CL + BL ÷ 2 + 1. In this equation, CL is the CAS latency rounded down to the next integer, WL is the programmed write latency, and BL is the burst length. 00 0 clocks 01 1 clock 10 2 clocks 11 3 clocks
4–5	RRT	Read-to-read turnaround. Specifies how many extra cycles are added between reads to different chip selects. As a default, 3 cycles are required between read commands to different chip selects. Extra cycles may be added with this field. Note: If 8-beat bursts are enabled, then 5 cycles are the default. Note that DDR2 does not support 8-beat bursts. 00 0 clocks 01 1 clock 10 2 clocks 11 3 clocks
6–7	WWT	Write-to-write turnaround. Specifies how many extra cycles are added between writes to different chip selects. As a default, 2 cycles are required between write commands to different chip selects. Extra cycles may be added with this field. Note: If 8-beat bursts are enabled, then 4 cycles are the default. Note that DDR2 does not support 8-beat bursts. 00 0 clocks 01 1 clock 10 2 clocks 11 3 clocks
8	—	Reserved

Table 9-9. TIMING_CFG_0 Field Descriptions (continued)

Bits	Name	Description
9–11	ACT_PD_EXIT	Active powerdown exit timing (t_{XARD} and t_{XARDS}). Specifies how many clock cycles to wait after exiting active powerdown before issuing any command. 000 Reserved 001 1 clock 010 2 clocks 011 3 clocks 100 4 clocks 101 5 clocks 110 6 clocks 111 7 clocks
12	—	Reserved
13–15	PRE_PD_EXIT	Precharge powerdown exit timing (t_{XP}). Specifies how many clock cycles to wait after exiting precharge powerdown before issuing any command. 000 Reserved 001 1 clock 010 2 clocks 011 3 clocks 100 4 clocks 101 5 clocks 110 6 clocks 111 7 clocks
16–19	—	Reserved
20–23	ODT_PD_EXIT	ODT powerdown exit timing (t_{AXPD}). Specifies how many clocks must pass after exiting powerdown before ODT may be asserted. 0000 0 clock 0001 1 clock 0010 2 clocks 0011 3 clocks 0100 4 clocks 0101 5 clocks 0110 6 clocks 0111 7 clocks 1000 8 clocks 1001 9 clocks 1010 10 clocks 1011 11 clocks 1100 12 clocks 1101 13 clocks 1110 14 clocks 1111 15 clocks
24–27	—	Reserved
28–31	MRS_CYC	Mode register set cycle time (t_{MRD}). Specifies the number of cycles that must pass after a Mode Register Set command until any other command. 0000 Reserved 0001 1 clock 0010 2 clocks 0011 3 clocks 0100 4 clocks 0101 5 clocks 0110 6 clocks 0111 7 clocks 1000 8 clocks 1001 9 clocks 1010 10 clocks 1011 11 clocks 1100 12 clocks 1101 13 clocks 1110 14 clocks 1111 15 clocks

9.4.1.5 DDR SDRAM Timing Configuration 1 (TIMING_CFG_1)

DDR SDRAM timing configuration register 1, shown in Figure 9-6, sets the number of clock cycles between various SDRAM control commands.

Offset 0x108 Access: Read/Write

	0	1 3	4 7	8	9 11	12 15	16 19	20	21 23	24	25 27	28	29 31
R W	—	PRETOACT	ACTTOPRE	—	ACTTORW	CASLAT	REFREC	—	WRREC	—	ACTTOACT	—	WRTORD
Reset	All zeros												

Figure 9-6. DDR SDRAM Timing Configuration 1 (TIMING_CFG_1)

Table 9-10 describes TIMING_CFG_1 fields.

Table 9-10. TIMING_CFG_1 Field Descriptions

Bits	Name	Description
0	—	Reserved
1–3	PRETOACT	Precharge-to-activate interval (t_{RP}). Determines the number of clock cycles from a precharge command until an activate or refresh command is allowed. 000 Reserved 001 1 clock 010 2 clocks 011 3 clocks 100 4 clocks 101 5 clocks 110 6 clocks 111 7 clocks
4–7	ACTTOPRE	Activate to precharge interval (t_{RAS}). Determines the number of clock cycles from an activate command until a precharge command is allowed. 0000 16 clocks 0001 17 clocks 0010 18 clocks 0011 19 clocks 0100 4 clocks 0101 5 clocks 0110 6 clocks 0111 7 clocks … 1111 15 clocks
8	—	Reserved
9–11	ACTTORW	Activate to read/write interval for SDRAM (t_{RCD}). Controls the number of clock cycles from an activate command until a read or write command is allowed. 000 Reserved 001 1 clock 010 2 clocks 011 3 clocks 100 4 clocks 101 5 clocks 110 6 clocks 111 7 clocks
12–15	CASLAT	$\overline{\text{MCAS}}$ latency from READ command. Number of clock cycles between registration of a READ command by the SDRAM and the availability of the first output data. If a READ command is registered at clock edge n and the latency is m clocks, data is available nominally coincident with clock edge $n + m$. This value must be programmed at initialization as described in Section 9.4.1.8, “DDR SDRAM Control Configuration 2 (DDR_SDRAM_CFG_2).”) 0000 Reserved 0001 1 clock 0010 1.5 clocks 0011 2 clocks 0100 2.5 clocks 0101 3 clocks 0110 3.5 clocks 0111 4 clocks 1000 4.5 clocks 1001 5 clocks 1010 5.5 clocks 1011 6 clocks 1100 6.5 clocks 1101 7 clocks 1110 7.5 clocks 1111 8 clocks

Table 9-10. TIMING_CFG_1 Field Descriptions (continued)

Bits	Name	Description
16–19	REFREC	Refresh recovery time (t_{RFC}). Controls the number of clock cycles from a refresh command until an activate command is allowed. This field is concatenated with TIMING_CFG_3[EXTREFREC] to obtain a 7-bit value for the total refresh recovery. Note that hardware adds an additional 8 clock cycles to the final, 7-bit value of the refresh recovery, such that t_{RFC} is calculated as follows: t_{RFC} = {EXT_REFREC || REFREC} + 8. 0000 8 clocks 0001 9 clocks 0010 10 clocks 0011 11 clocks ... 1111 23 clocks
20	—	Reserved
21–23	WRREC	Last data to precharge minimum interval (t_{WR}). Determines the number of clock cycles from the last data associated with a write command until a precharge command is allowed. 000 Reserved 001 1 clock 010 2 clocks 011 3 clocks 100 4 clocks 101 5 clocks 110 6 clocks 111 7 clocks
24	—	Reserved
25–27	ACTTOACT	Activate-to-activate interval (t_{RRD}). Number of clock cycles from an activate command until another activate command is allowed for a different logical bank in the same physical bank (chip select). 000 Reserved 001 1 clock 010 2 clocks 011 3 clocks 100 4 clocks 101 5 clocks 110 6 clocks 111 7 clocks
28	—	Reserved
29–31	WRTORD	Last write data pair to read command issue interval (t_{WTR}). Number of clock cycles between the last write data pair and the subsequent read command to the same physical bank. 000 Reserved 001 1 clock 010 2 clocks 011 3 clocks 100 4 clocks 101 5 clocks 110 6 clocks 111 7 clocks

9.4.1.6 DDR SDRAM Timing Configuration 2 (TIMING_CFG_2)

DDR SDRAM timing configuration 2, shown in Figure 9-7, sets the clock delay to data for writes.

Offset 0x10C Access: Read/Write

	0	1–3	4–8	9	10–12	13–15	16–18	19–21	22	23–25	26–31
R W	—	ADD_LAT	CPO	—	WR_LAT	—	RD_TO_PRE	WR_DATA_DELAY	—	CKE_PLS	FOUR_ACT
Reset	All zeros										

Figure 9-7. DDR SDRAM Timing Configuration 2 Register (TIMING_CFG_2)

Table 9-11 describes the TIMING_CFG_2 fields.

Table 9-11. TIMING_CFG_2 Field Descriptions

Bits	Name	Description
0	—	Reserved
1–3	ADD_LAT	Additive latency. The additive latency must be set to a value less than TIMING_CFG_1[ACTTORW]. (DDR2-specific) 000 0 clocks 001 1 clock 010 2 clocks 011 3 clocks 100 4 clocks 101 5 clocks 110 Reserved 111 Reserved
4–8	CPO[1]	$\overline{\text{MCAS}}$-to-preamble override. Defines the number of DRAM cycles between when a read is issued and when the corresponding DQS preamble is valid for the memory controller. For these decodings, “READ_LAT” is equal to the CAS latency plus the additive latency. 00000READ_LAT + 1 00001Reserved 00010READ_LAT 00011READ_LAT + 1/4 00100READ_LAT + 1/2 00101READ_LAT + 3/4 00110READ_LAT + 1 00111READ_LAT + 5/4 01000READ_LAT + 3/2 01001READ_LAT + 7/4 01010READ_LAT + 2 01011READ_LAT + 9/4 01100READ_LAT + 5/2 01101READ_LAT + 11/4 01110READ_LAT + 3 01111READ_LAT + 13/4 10000READ_LAT + 7/2 10001READ_LAT + 15/4 10010READ_LAT + 4 10011READ_LAT + 17/4 10100READ_LAT + 9/2 10101READ_LAT + 19/4 10110–11111 Reserved
9	—	Reserved
10–12	WR_LAT	Write latency. Note that the total write latency for DDR2 is equal to WR_LAT + ADD_LAT; the write latency for DDR1 is 1. 000 Reserved 001 1 clock 010 2 clocks 011 3 clocks 100 4 clocks 101 5 clocks 110 6 clocks 111 7 clocks
13–15	—	Reserved
16–18	RD_TO_PRE	Read to precharge (t_{RTP}). For DDR2, with a non-zero ADD_LAT value, takes a minimum of ADD_LAT + t_{RTP} cycles between read and precharge. For DDR1 with burst length of 4, must be set to 010; for DDR1 with burst length of 8, must be set to 100. 000 Reserved 001 1 cycle 010 2 cycles 011 3 cycles 100 4 cycles 101–111 Reserved

Table 9-11. TIMING_CFG_2 Field Descriptions (continued)

Bits	Name	Description
19–21	WR_DATA_DELAY	Write command to write data strobe timing adjustment. Controls the amount o delay applied to the data and data strobes for writes. See Section 9.5.6, “DDR SDRAM Write Timing Adjustments,” for details. 000 0 clock delay 001 1/4 clock delay 010 1/2 clock delay 011 3/4 clock delay 100 1 clock delay 101 5/4 clock delay 110 3/2 clock delay 111 Reserved
22	—	Reserved
23–25	CKE_PLS	Minimum CKE pulse width (t_{CKE})Can be set to 001 for DDR1. 000 Reserved 001 1 cycle 010 2 cycles 011 3 cycles 100 4 cycles 101–111 Reserved
26–31	FOUR_ACT	Window for four activates (t_{FAW}). This is applied to DDR2 with eight logical banks only. Must be set to 000001 for DDR1. 000000 Reserved 000001 1 cycle 000010 2 cycles 000011 3 cycles 000100 4 cycles ... 010011 19 cycles 010100 20 cycles 010101–111111 Reserved

[1] For CPO decodings other than 00000 and 11111, ‘READ_LAT’ is rounded up to the next integer value.

9.4.1.7 DDR SDRAM Control Configuration (DDR_SDRAM_CFG)

The DDR SDRAM control configuration register, shown in Figure 9-8, enables the interface logic and specifies certain operating features such as self refreshing, error checking and correcting, registered DIMMs, and dynamic power management.

Offset 0x110

Access: Read/Write

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	MEM_EN	SREN	ECC_EN	RD_EN	—	SDRAM_TYPE			—		DYN_PWR	—	32_BE	8_BE	—	
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

	16	17–23	24–25	26	27	28	29	30	31
R W	2T_EN	BA_INTLV_CTL	—	x32_EN	PCHB8	HSE	—	MEM_HALT	BI
Reset	All zeros								

Figure 9-8. DDR SDRAM Control Configuration Register (DDR_SDRAM_CFG)

Table 9-12 describes the DDR_SDRAM_CFG fields.

Table 9-12. DDR_SDRAM_CFG Field Descriptions

Bits	Name	Description
0	MEM_EN	DDR SDRAM interface logic enable. 0 SDRAM interface logic is disabled. 1 SDRAM interface logic is enabled. Must not be set until all other memory configuration parameters have been appropriately configured by initialization code.
1	SREN	Self refresh enable (during sleep). 0 SDRAM self refresh is disabled during sleep. Whenever self-refresh is disabled, the system is responsible for preserving the integrity of SDRAM during sleep. 1 SDRAM self refresh is enabled during sleep.
2	ECC_EN	ECC enable. Note that non-correctable read errors may cause an interrupt. 0 No ECC errors are reported. No ECC interrupts are generated. 1 ECC is enabled.
3	RD_EN	Registered DIMM enable. Specifies the type of DIMM used in the system. 0 Indicates unbuffered DIMMs. 1 Indicates registered DIMMs. **Note:** RD_EN and 2T_EN must not both be set at the same time.
4	—	Reserved
5–7	SDRAM_TYPE	Type of SDRAM device to be used. This field is used when issuing the automatic hardware initialization sequence to DRAM through Mode Register Set and Extended Mode Register Set commands. Default value is 010 designating DDR1 SDRAM. 000–001Reserved 010 DDR1 SDRAM 011 DDR2 SDRAM 100 Reserved 101 Reserved 110 Reserved 111 Reserved
8–9	—	Reserved
10	DYN_PWR	Dynamic power management mode 0 Dynamic power management mode is disabled. 1 Dynamic power management mode is enabled. If there is no ongoing memory activity, the SDRAM CKE signal is negated.
11	—	Reserved
12	32_BE	32-bit bus enable. 0 64-bit bus is used. 1 32-bit bus is used.
13	8_BE	8-beat burst enable. 0 4-beat bursts are used on the DRAM interface. 1 8-beat bursts are used on the DRAM interface. **Note:** DDR1 must use 8-beat bursts when using 32-bit bus mode (32_BE = 1) and 4-beat bursts when using 64-bit bus mode **Note:** DDR2 must use 4-beat bursts when using 32/64-bit bus mode
14–15	—	Reserved

Table 9-12. DDR_SDRAM_CFG Field Descriptions (continued)

Bits	Name	Description
16	2T_EN	Enable 2T timing. 0 1T timing is enabled. The DRAM command/address is held for only 1 cycle on the DRAM bus. 1 2T timing is enabled. The DRAM command/address are held for 2 full cycles on the DRAM bus for every DRAM transaction. However, the chip select is only held for the second cycle. **Note:** RD_EN and 2T_EN must not both be set at the same time.
17–23	BA_INTLV_CTL	Bank (chip select) interleaving control. Set this field only if you wish to use bank interleaving. (‘*x*’ denotes a don’t care bit value. All unlisted field values are reserved.) 0000000No external memory banks are interleaved 1000000External memory banks 0 and 1 are interleaved 0100000External memory banks 2 and 3 are interleaved 1100000External memory banks 0 and 1 are interleaved together and banks 2 and 3 are interleaved together *xx*00100External memory banks 0 through 3 are all interleaved together
24–25	—	Reserved
26	x32_EN	x32 enable. 0 Either x8 or x16 discrete DRAM chips are used. In this mode, each data byte has a dedicated corresponding data strobe. 1 x32 discrete DRAM chips are used. In this mode, DQS0 is used to capture DQ[0:31], DQS4 is used to capture DQ[32:63] and DQS8 is used to capture ECC[0:7].
27	PCHB8	Precharge bit 8 enable. 0 MA[10] is used to indicate the auto-precharge and precharge all commands. 1 MA[8] is used to indicate the auto-precharge and precharge all commands. If x32_EN is cleared, then PCHB8 should be cleared as well.
28	HSE	Global half-strength override Sets I/O driver impedance to half strength. This impedance is used by the MDIC, address/command, data, and clock impedance values, but only if automatic hardware calibration is disabled and the corresponding group's software override is disabled in the DDR control driver register(s) described in Section 5.4.4.12, “DDR Control Driver Register (DDRCDR).” This bit should be cleared if using automatic hardware calibration. 0 I/O driver impedance is configured to full strength. 1 I/O driver impedance is configured to half strength.
29	—	Reserved

Table 9-12. DDR_SDRAM_CFG Field Descriptions (continued)

Bits	Name	Description
30	MEM_HALT	DDR memory controller halt. When this bit is set, the memory controller does not accept any new data read/write transactions to DDR SDRAM until the bit is cleared again. This can be used when bypassing initialization and forcing MODE REGISTER SET commands through software. 0 DDR controller accepts new transactions. 1 DDR controller finishes any remaining transactions, and then it remains halted until this bit is cleared by software.
31	BI	Bypass initialization 0 DDR controller cycles through initialization routine based on SDRAM_TYPE 1 Initialization routine is bypassed. Software is responsible for initializing memory through DDR_SDRAM_MODE2 register. If software is initializing memory, then the MEM_HALT bit can be set to prevent the DDR controller from issuing transactions during the initialization sequence. Note that the DDR controller does not issue a DLL reset to the DRAMs when bypassing the initialization routine, regardless of the value of DDR_SDRAM_CFG[DLL_RST_DIS]. If a DLL reset is required, then the controller should be forced to enter and exit self-refresh after the controller is enabled. See Section 9.4.1.15, “DDR Initialization Address (DDR_INIT_ADDR),” for details on avoiding ECC errors in this mode.

9.4.1.8 DDR SDRAM Control Configuration 2 (DDR_SDRAM_CFG_2)

The DDR SDRAM control configuration register 2, shown in Figure 9-9, provides more control configuration for the DDR controller.

Offset 0x114 Access: Read/Write

	0	1	2	3	4 5	6 8	9 10	11 15
R W	FRC_SR	SR_IE	DLL_RST_DIS	—	DQS_CFG	—	ODT_CFG	—
Reset	All zeros							

	16 19	20 26	27	28 31
R W	NUM_PR	—	D_INIT	—
Reset	All zeros			

Figure 9-9. DDR SDRAM Control Configuration Register 2 (DDR_SDRAM_CFG_2)

Table 9-13 describes the DDR_SDRAM_CFG_2 fields.

Table 9-13. DDR_SDRAM_CFG_2 Field Descriptions

Bits	Name	Description
0	FRC_SR	Force self-refresh 0 DDR controller operates in normal mode. 1 DDR controller enters self-refresh mode.
1	SR_IE	Self-refresh interrupt enable. The DDR controller can be placed into self refresh mode if DDR_SR_REQ (IRQ1) is asserted. This is considered a 'panic interrupt' by the DDR controller, and it will enter self refresh as soon as possible. DDR_SDRAM_CFG[SREN] must also be set if the panic interrupt is used. 0 DDR controller will not enter self-refresh mode if panic interrupt is asserted. 1 DDR controller will enter self-refresh mode if panic interrupt is asserted.
2	DLL_RST_DIS	DLL reset disable. The DDR controller typically issues a DLL reset to the DRAMs when exiting self refresh. However, this function may be disabled by setting this bit during initialization. 0 DDR controller issues a DLL reset to the DRAMs when exiting self refresh. 1 DDR controller does not issue a DLL reset to the DRAMs when exiting self refresh.
3	—	Reserved
4–5	DQS_CFG	DQS configuration 00 Only true DQS signals are used. 01 Reserved 10 Reserved 11 Reserved
6–8	—	Reserved
9–10	ODT_CFG	ODT configuration This field defines how ODT is driven to the on-chip IOs. See Section 5.4.4.12, "DDR Control Driver Register (DDRCDR)," which defines the termination value that is used. (DDR2-specific, must be cleared for DDR1) 00 Never assert ODT to internal IOs 01 Assert ODT to internal IOs only during writes to DRAM 10 Assert ODT to internal IOs only during reads to DRAM 11 Always keep ODT asserted to internal IOs
11–15	—	Reserved.
16–19	NUM_PR	Number of posted refreshes This determines how many posted refreshes, if any, can be issued at one time. Note that if posted refreshes are used, then this field, along with DDR_SDRAM_INTERVAL[REFINT], must be programmed such that the maximum t_{ras} specification cannot be violated. For example, some DDR1 SDRAMs are not able to use more than 3 posted refreshes because the required refresh interval could then exceed the maximum constraint for t_{ras}. 0000 Reserved 0001 1 refresh is issued at a time 0010 2 refreshes is issued at a time 0011 3 refreshes is issued at a time ... 1000 8 refreshes is issued at a time 1001–1111 Reserved
20–24	—	Reserved
20–26	—	Reserved

Table 9-13. DDR_SDRAM_CFG_2 Field Descriptions (continued)

Bits	Name	Description
27	D_INIT	DRAM data initialization This bit is set by software, and it is cleared by hardware. If software sets this bit before the memory controller is enabled, the controller automatically initializes DRAM after it is enabled. This bit is automatically cleared by hardware once the initialization is completed. This data initialization bit should only be set when the controller is idle. 0 There is not data initialization in progress, and no data initialization is scheduled 1 The memory controller initializes memory once it is enabled. This bit remains asserted until the initialization is complete. The value in DDR_DATA_INIT register is used to initialize memory.
28–31	—	Reserved

9.4.1.9 DDR SDRAM Mode Configuration (DDR_SDRAM_MODE)

The DDR SDRAM mode configuration register, shown in Figure 9-10, sets the values loaded into the DDR's mode registers.

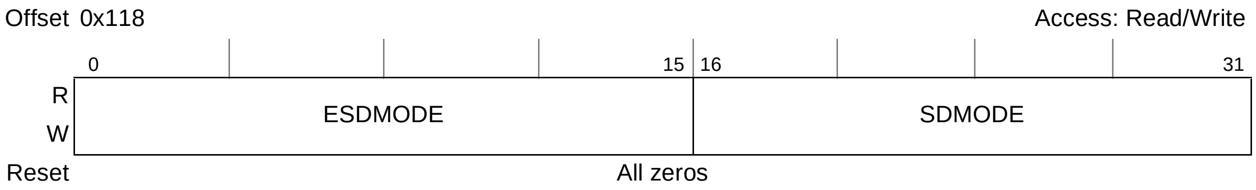


Figure 9-10. DDR SDRAM Mode Configuration Register (DDR_SDRAM_MODE)

Table 9-14 describes the DDR_SDRAM_MODE fields.

Table 9-14. DDR_SDRAM_MODE Field Descriptions

Bits	Name	Description
0–15	ESDMODE	Extended SDRAM mode Specifies the initial value loaded into the DDR SDRAM extended mode register. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb of ESDMODE, which, in the big-endian convention shown in Figure 9-10, corresponds to ESDMODE[15]. The msb of the SDRAM extended mode register value must be stored at ESDMODE[0].
16–31	SDMODE	SDRAM mode Specifies the initial value loaded into the DDR SDRAM mode register. The range of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during DDR SDRAM initialization), MA[0] presents the lsb of SDMODE, which, in the big-endian convention shown in Figure 9-10, corresponds to SDMODE[15]. The msb of the SDRAM mode register value must be stored at SDMODE[0]. Because the memory controller forces SDMODE[7] to certain values depending on the state of the initialization sequence, (for resetting the SDRAM's DLL) the corresponding bits of this field are ignored by the memory controller. Note that SDMODE[7] is mapped to MA[8].

9.4.1.10 DDR SDRAM Mode 2 Configuration (DDR_SDRAM_MODE_2)

The DDR SDRAM mode 2 configuration register, shown in Figure 9-11, sets the values loaded into the DDR's extended mode 2 and 3 registers (for DDR2).

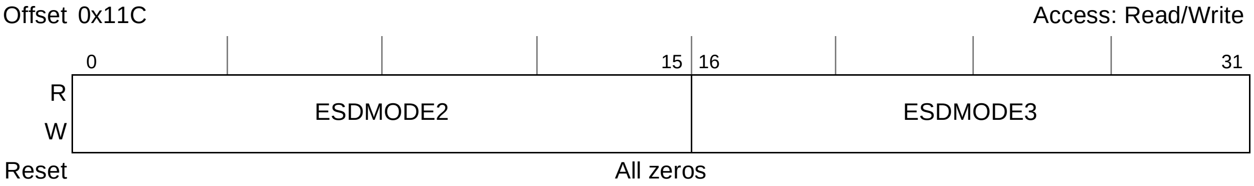


Figure 9-11. DDR SDRAM Mode 2 Configuration Register (DDR_SDRAM_MODE_2)

Table 9-15 describes the DDR_SDRAM_MODE_2 fields.

Table 9-15. DDR_SDRAM_MODE_2 Field Descriptions

Bits	Name	Description
0–15	ESDMODE2	Extended SDRAM mode 2 Specifies the initial value loaded into the DDR SDRAM extended 2 mode register. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE2, which, in the big-endian convention shown in Figure 9-11, corresponds to ESDMODE2[15]. The msb of the SDRAM extended mode 2 register value must be stored at ESDMODE2[0].
16–31	ESDMODE3	Extended SDRAM mode 3 Specifies the initial value loaded into the DDR SDRAM extended 3 mode register. The range of legal values of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during DDR SDRAM initialization), MA[0] presents the lsb of ESDMODE3, which, in the big-endian convention shown in Figure 9-11, corresponds to ESDMODE3[15]. The msb of the SDRAM extended mode 3 register value must be stored at ESDMODE3[0].

9.4.1.11 DDR SDRAM Mode Control Register (DDR_SDRAM_MD_CNTL)

The DDR SDRAM mode control register, shown in Figure 9-12, allows the user to carry out the following tasks:

- Issue a mode register set command to a particular chip select
- Issue an immediate refresh to a particular chip select
- Issue an immediate precharge or precharge all command to a particular chip select
- Force the CKE signals to a specific value

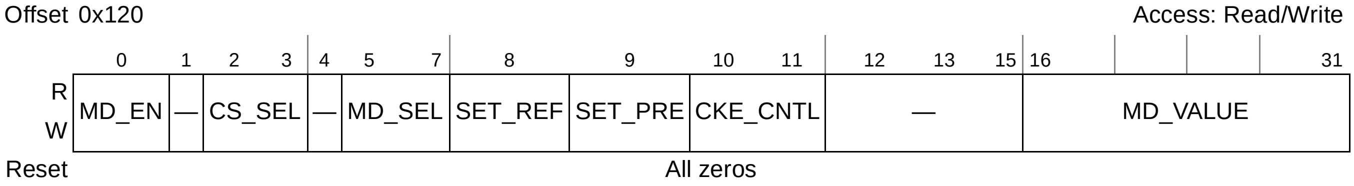


Figure 9-12. DDR SDRAM Mode Control Register (DDR_SDRAM_MD_CNTL)

Table 9-16 describes the fields of this register. Table 9-17 shows the user how to set the fields of this register to accomplish the above tasks.

NOTE

Note that MD_EN, SET_REF, and SET_PRE are mutually exclusive; only one of these fields can be set at a time.

Table 9-16. DDR_SDRAM_MD_CNTL Field Descriptions

Bits	Name	Description
0	MD_EN	Mode enable Setting this bit specifies that valid data in MD_VALUE is ready to be written to DRAM as one of the following commands: • MODE REGISTER SET • EXTENDED MODE REGISTER SET • EXTENDED MODE REGISTER SET 2 • EXTENDED MODE REGISTER SET 3 The specific command to be executed is selected by setting MD_SEL. In addition, the chip select must be chosen by setting CS_SEL. MD_EN is set by software and cleared by hardware once the command has been issued. 0 Indicates that no mode register set command needs to be issued. 1 Indicates that valid data contained in the register is ready to be issued as a mode register set command.
1	—	Reserved
2–3	CS_SEL	Select chip select Specifies the chip select that is driven active due to any command forced by software in DDR_SDRAM_MD_CNTL. 00 Chip select 0 is active 01 Chip select 1 is active 10 Chip select 2 is active 11 Chip select 3 is active
4	—	Reserved
5–7	MD_SEL	Mode register select MD_SEL specifies one of the following: • During a mode select command, selects the SDRAM mode register to be changed • During a precharge command, selects the SDRAM logical bank to be precharged. A precharge all command ignores this field. • During a refresh command, this field is ignored. Note that MD_SEL contains the value that is presented onto the memory bank address pins (MBA*n*) of the DDR controller. 000 MR 001 EMR 010 EMR2 011 EMR3
8	SET_REF	Set refresh Forces an immediate refresh to be issued to the chip select specified by DDR_SDRAM_MD_CNTL[CS_SEL]. This bit is set by software and cleared by hardware once the command has been issued. 0 Indicates that no refresh command needs to be issued. 1 Indicates that a refresh command is ready to be issued.

Table 9-16. DDR_SDRAM_MD_CNTL Field Descriptions (continued)

Bits	Name	Description
9	SET_PRE	Set precharge Forces a precharge or precharge all to be issued to the chip select specified by DDR_SDRAM_MD_CNTL[CS_SEL]. This bit is set by software and cleared by hardware once the command has been issued. 0 Indicates that no precharge all command needs to be issued. 1 Indicates that a precharge all command is ready to be issued.
10–11	CKE_CNTL	Clock enable control Allows software to globally clear or set all CKE signals issued to DRAM. Once software has forced the value driven on CKE, that value continues to be forced until software clears the CKE_CNTL bits. At that time, the DDR controller continues to drive the CKE signals to the same value forced by software until another event causes the CKE signals to change (such as, self refresh entry/exit, power down entry/exit). 00 CKE signals are not forced by software. 01 CKE signals are forced to a low value by software. 10 CKE signals are forced to a high value by software. 11 Reserved
12–15	—	Reserved
16–31	MD_VALUE	Mode register value This field, which specifies the value that is presented on the memory address pins of the DDR controller during a mode register set command, is significant only when this register is used to issue a mode register set command or a precharge or precharge all command. For a mode register set command, this field contains the data to be written to the selected mode register. For a precharge command, only bit five is significant: 0 Issue a precharge command; MD_SEL selects the logical bank to be precharged 1 Issue a precharge all command; all logical banks are precharged

Table 9-17 shows how DDR_SDRAM_MD_CNTL fields should be set for each of the tasks described above.

Table 9-17. Settings of DDR_SDRAM_MD_CNTL Fields

Field	Mode Register Set	Refresh	Precharge	Clock Enable Signals Control
MD_EN	1	0	0	—
SET_REF	0	1	0	—
SET_PRE	0	0	1	—
CS_SEL	Chooses chip select (CS)			—
MD_SEL	Select mode register. See Table 9-16.	—	Selects logical bank	—
MD_VALUE	Value written to mode register	—	Only bit five is significant. See Table 9-16.	—
CKE_CNTL	0	0	0	See Table 9-16.

9.4.1.12 DDR SDRAM Interval Configuration (DDR_SDRAM_INTERVAL)

The DDR SDRAM interval configuration register, shown in Figure 9-13, sets the number of DRAM clock cycles between bank refreshes issued to the DDR SDRAMs. In addition, the number of DRAM cycles that a page is maintained after it is accessed is provided here.

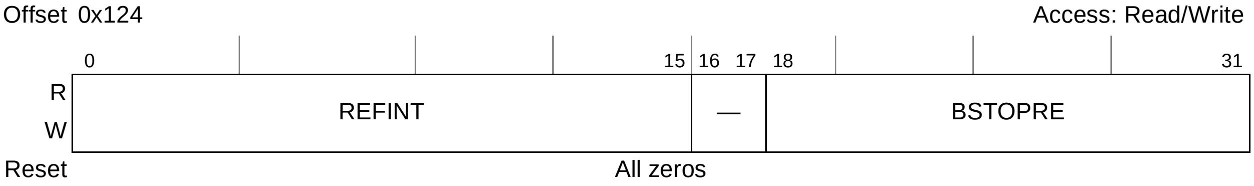


Figure 9-13. DDR SDRAM Interval Configuration Register (DDR_SDRAM_INTERVAL)

Table 9-18 describes the DDR_SDRAM_INTERVAL fields.

Table 9-18. DDR_SDRAM_INTERVAL Field Descriptions

Bits	Name	Description
0–15	REFINT	Refresh interval Represents the number of memory bus clock cycles between refresh cycles. Depending on DDR_SDRAM_CFG_2[NUM_PR], some number of rows are refreshed in each DDR SDRAM physical bank during each refresh cycle. The value for REFINT depends on the specific SDRAMs used and the interface clock frequency. Refreshes are not issued when the REFINT is set to all 0s.
16–17	—	Reserved
18–31	BSTOPRE	Precharge interval Sets the duration (in memory bus clocks) that a page is retained after a DDR SDRAM access. If BSTOPRE is zero, the DDR memory controller uses auto-precharge read and write commands rather than operating in page mode. This is called global auto-precharge mode.

9.4.1.13 DDR SDRAM Data Initialization (DDR_DATA_INIT)

The DDR SDRAM data initialization register, shown in Figure 9-14, provides the value that is used to initialize memory if DDR_SDRAM_CFG2[D_INIT] is set.

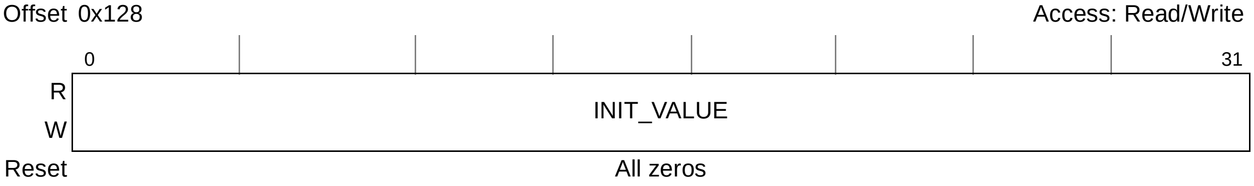


Figure 9-14. DDR SDRAM Data Initialization Configuration Register (DDR_DATA_INIT)

Table 9-19 describes the DDR_DATA_INIT fields.

Table 9-19. DDR_DATA_INIT Field Descriptions

Bits	Name	Description
0–31	INIT_VALUE	Initialization value. Represents the value that DRAM is initialized with if DDR_SDRAM_CFG2[D_INIT] is set.

9.4.1.14 DDR SDRAM Clock Control (DDR_SDRAM_CLK_CNTL)

The DDR SDRAM clock control configuration register, shown in Figure 9-15, provides a 1/4-cycle clock adjustment.

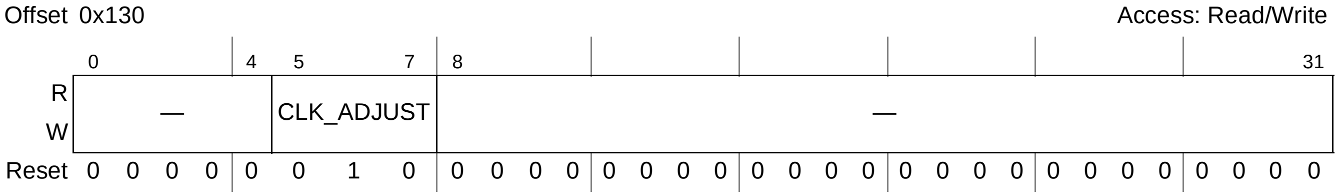


Figure 9-15. DDR SDRAM Clock Control Configuration Register (DDR_SDRAM_CLK_CNTL)

Table 9-20 describes the DDR_SDRAM_CLK_CNTL fields.

Table 9-20. DDR_SDRAM_CLK_CNTL Field Descriptions

Bits	Name	Description
0–4	—	Reserved
5–7	CLK_ADJUST	Clock adjust. 000 Clock is launched aligned with address/command 001 Clock is launched 1/4 applied cycle after address/command 010 Clock is launched 1/2 applied cycle after address/command 011 Clock is launched 3/4 applied cycle after address/command 100 Clock is launched 1 applied cycle after address/command 101–111Reserved
8	—	Reserved
9–31	—	Reserved

9.4.1.15 DDR Initialization Address (DDR_INIT_ADDR)

The DDR SDRAM initialization address register, shown in Figure 9-16, provides the address that is used for the automatic $\overline{\text{CAS}}$ to preamble calibration after POR.

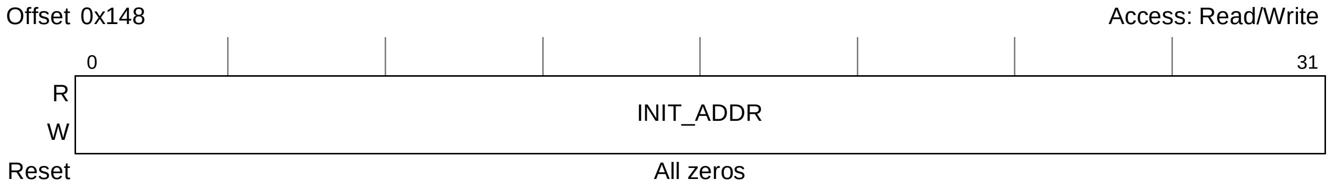


Figure 9-16. DDR Initialization Address Configuration Register (DDR_INIT_ADDR)

Table 9-21 describes the DDR_INIT_ADDR fields.

Table 9-21. DDR_INIT_ADDR Field Descriptions

Bits	Name	Description
0–31	INIT_ADDR	Initialization address. Represents the address that is used for the automatic CAS to preamble calibration at POR.

9.4.1.16 DDR IP Block Revision 1 (DDR_IP_REV1)

The DDR IP block revision 1 register, shown in Figure 9-17, provides read-only fields with the IP block ID, along with major and minor revision information.

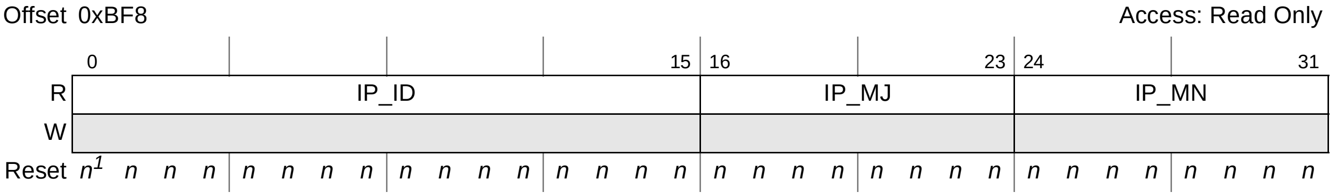


Figure 9-17. DDR IP Block Revision 1 (DDR_IP_REV1)

[1] For reset values, see Table 9-22.

Table 9-22 describes the DDR_IP_REV1 fields.

Table 9-22. DDR_IP_REV1 Field Descriptions

Bits	Name	Description
0–15	IP_ID	IP block ID. For the DDR controller, this value is 0x0002.
16–23	IP_MJ	Major revision. This is currently set to 0x02.
24–31	IP_MN	Minor revision. This is currently set to 0x00.

9.4.1.17 DDR IP Block Revision 2 (DDR_IP_REV2)

The DDR IP block revision 2 register, shown in Figure 9-18, provides read-only fields with the IP block integration and configuration options.

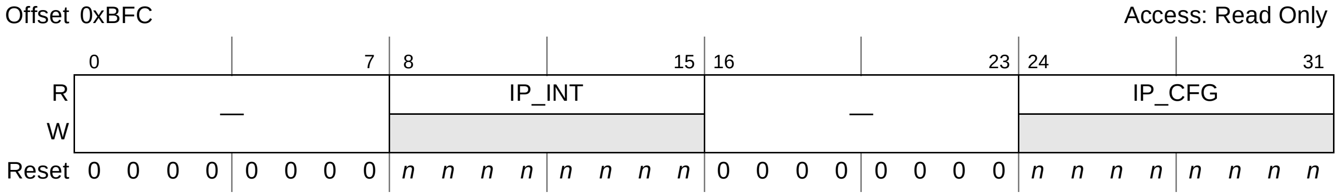


Figure 9-18. DDR IP Block Revision 2 (DDR_IP_REV2)

Table 9-23 describes the DDR_IP_REV2 fields.

Table 9-23. DDR_IP_REV2 Field Descriptions

Bits	Name	Description
0–7	—	Reserved
8–15	IP_INT	IP block integration options
16–23	—	Reserved
24–31	IP_CFG	IP block configuration options

9.4.1.18 Memory Data Path Error Injection Mask High (DATA_ERR_INJECT_HI)

The memory data path error injection mask high register is shown in Figure 9-19.

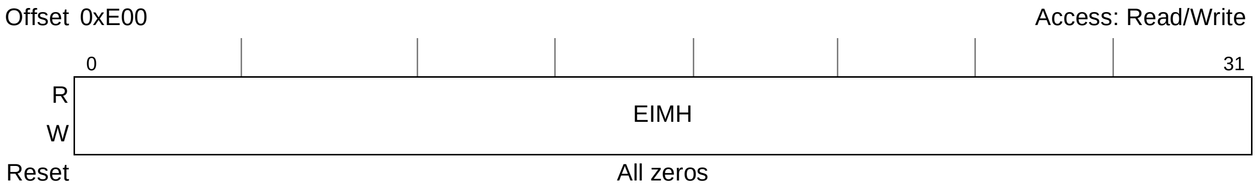


Figure 9-19. Memory Data Path Error Injection Mask High Register (DATA_ERR_INJECT_HI)

Table 9-24 describes the DATA_ERR_INJECT_HI fields.

Table 9-24. DATA_ERR_INJECT_HI Field Descriptions

Bits	Name	Description
0–31	EIMH	Error injection mask high data path Used to test ECC by forcing errors on the high word of the data path. Setting a bit causes the corresponding data path bit to be inverted on memory bus writes.

9.4.1.19 Memory Data Path Error Injection Mask Low (DATA_ERR_INJECT_LO)

The memory data path error injection mask low register is shown in Figure 9-20.

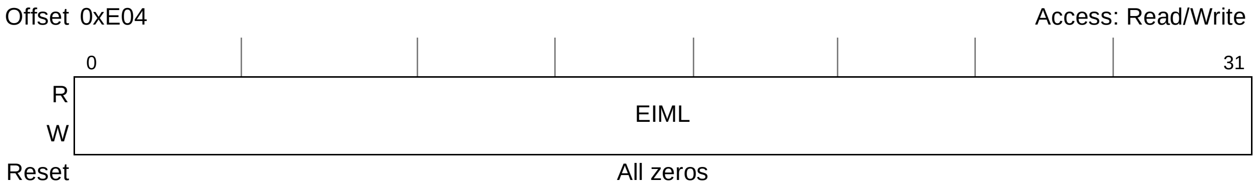


Figure 9-20. Memory Data Path Error Injection Mask Low Register (DATA_ERR_INJECT_LO)

Table 9-25 describes the DATA_ERR_INJECT_LO fields.

Table 9-25. DATA_ERR_INJECT_LO Field Descriptions

Bits	Name	Description
0–31	EIML	Error injection mask low data path Used to test ECC by forcing errors on the low word of the data path. Setting a bit causes the corresponding data path bit to be inverted on memory bus writes.

9.4.1.20 Memory Data Path Error Injection Mask ECC (ERR_INJECT)

The memory data path error injection mask ECC register, shown in Figure 9-21, sets the ECC mask, enables errors to be written to ECC memory, and allows the ECC byte to mirror the most significant data byte.

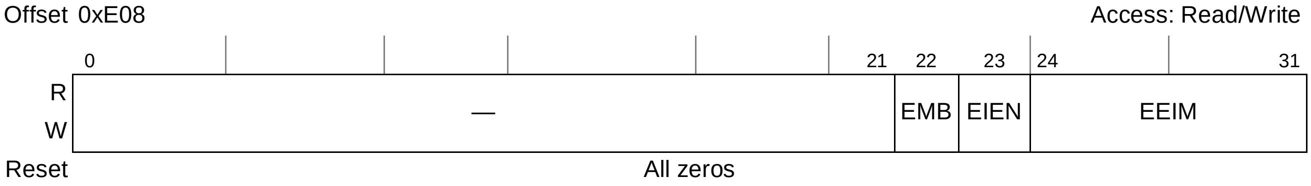


Figure 9-21. Memory Data Path Error Injection Mask ECC Register (ERR_INJECT)

Table 9-26 describes the ERR_INJECT fields.

Table 9-26. ERR_INJECT Field Descriptions

Bits	Name	Description
0–21	—	Reserved
22	EMB	ECC mirror byte 0 Mirror byte functionality disabled. 1 Mirror the most significant data path byte onto the ECC byte.
23	EIEN	Error injection enable 0 Error injection disabled. 1 Error injection enabled. This applies to the data mask bits, the ECC mask bits, and the ECC mirror bit. Note that error injection should not be enabled until the memory controller has been enabled through DDR_SDRAM_CFG[MEM_EN].
24–31	EEIM	ECC error injection mask. Setting a mask bit causes the corresponding ECC bit to be inverted on memory bus writes.

9.4.1.21 Memory Data Path Read Capture High (CAPTURE_DATA_HI)

The memory data path read capture high register, shown in Figure 9-22, stores the high word of the read data path during error capture.

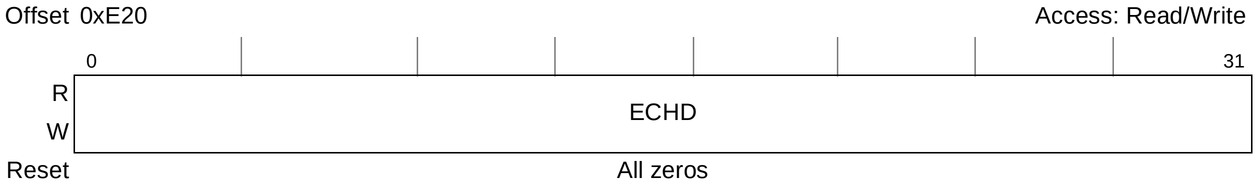


Figure 9-22. Memory Data Path Read Capture High Register (CAPTURE_DATA_HI)

Table 9-27 describes the CAPTURE_DATA_HI fields.

Table 9-27. CAPTURE_DATA_HI Field Descriptions

Bits	Name	Description
0–31	ECHD	Error capture high data path. Captures the high word of the data path when errors are detected.

9.4.1.22 Memory Data Path Read Capture Low (CAPTURE_DATA_LO)

The memory data path read capture low register, shown in Figure 9-23, stores the low word of the read data path during error capture.

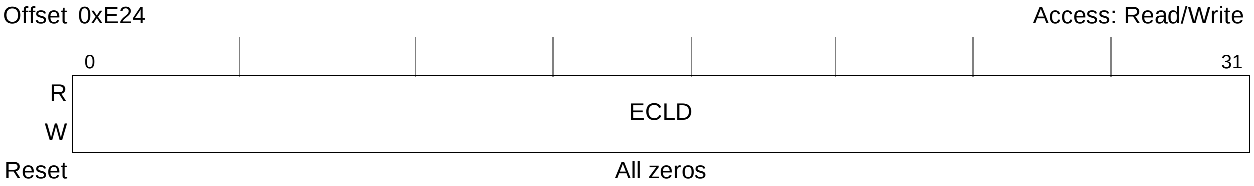


Figure 9-23. Memory Data Path Read Capture Low Register (CAPTURE_DATA_LO)

Table 9-28 describes the CAPTURE_DATA_LO fields.

Table 9-28. CAPTURE_DATA_LO Field Descriptions

Bits	Name	Description
0–31	ECLD	Error capture low data path. Captures the low word of the data path when errors are detected.

9.4.1.23 Memory Data Path Read Capture ECC (CAPTURE_ECC)

The memory data path read capture ECC register, shown in Figure 9-24, stores the ECC syndrome bits that were on the data bus when an error was detected.

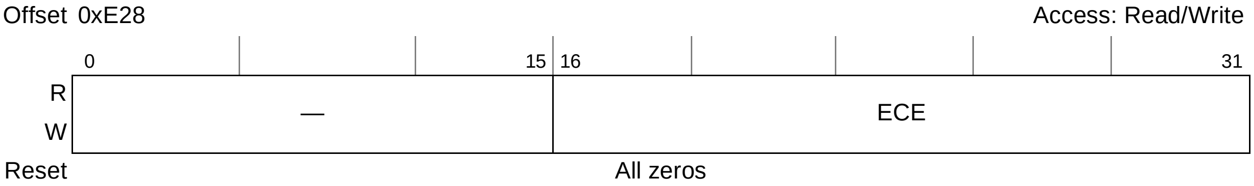


Figure 9-24. Memory Data Path Read Capture ECC Register (CAPTURE_ECC)

Table 9-29 describes the CAPTURE_ECC fields.

Table 9-29. CAPTURE_ECC Field Descriptions

Bits	Name	Description
0–15	—	Reserved
16–31	ECE	Error capture ECC. Captures the ECC bits on the data path whenever errors are detected. 16:23—8-bit ECC code for 1st 32 bits 24:31—8-bit ECC code for 2nd 32 bits **Note:** In 64-bit mode, only 24:31 should be used, although 16:23 shows the 8-bit ECC code replicated. ; all 64-bits in 64-bit bus mode

9.4.1.24 Memory Error Detect (ERR_DETECT)

The memory error detect register stores the detection bits for multiple memory errors, single- and multiple-bit ECC errors, and memory select errors. It is a read/write register. A bit can be cleared by writing a one to the bit. System software can determine the type of memory error by examining the contents of this register. If an error is disabled with ERR_DISABLE, the corresponding error is never detected or captured in ERR_DETECT.

ERR_DETECT is shown in Figure 9-25.

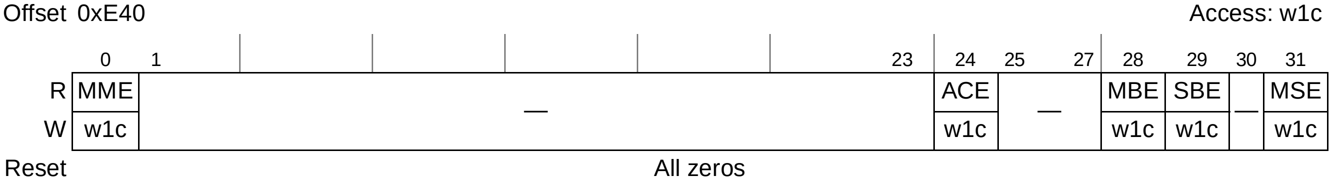


Figure 9-25. Memory Error Detect Register (ERR_DETECT)

Table 9-30 describes the ERR_DETECT fields.

Table 9-30. ERR_DETECT Field Descriptions

Bits	Name	Description
0	MME	Multiple memory errors. This bit is cleared by software writing a 1. 0 Multiple memory errors of the same type were not detected. 1 Multiple memory errors of the same type were detected.
1–23	—	Reserved
24	ACE	Automatic calibration error. This bit is cleared by software writing a 1. 0 An automatic calibration error has not been detected. 1 An automatic calibration error has been detected.
25–27	—	Reserved
28	MBE	Multiple-bit error. This bit is cleared by software writing a 1. 0 A multiple-bit error has not been detected. 1 A multiple-bit error has been detected.
29	SBE	Single-bit ECC error. This bit is cleared by software writing a 1. 0 The number of single-bit ECC errors detected has not crossed the threshold set in ERR_SBE[SBET]. 1 The number of single-bit ECC errors detected crossed the threshold set in ERR_SBE[SBET].
30	—	Reserved
31	MSE	Memory select error. This bit is cleared by software writing a 1. 0 A memory select error has not been detected. 1 A memory select error has been detected.

9.4.1.25 Memory Error Disable (ERR_DISABLE)

The memory error disable register, shown in Figure 9-26, allows selective disabling of the DDR controller's error detection circuitry. Disabled errors are not detected or reported.

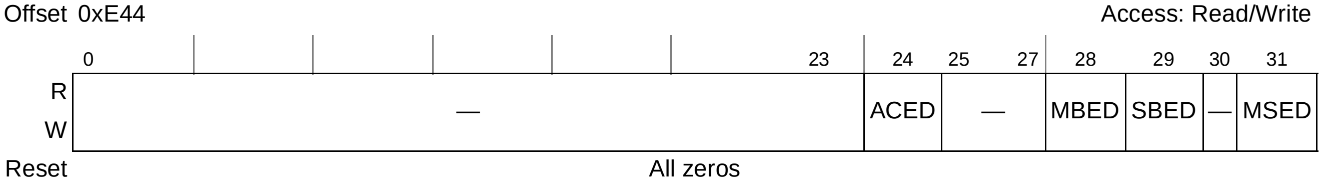


Figure 9-26. Memory Error Disable Register (ERR_DISABLE)

Table 9-31 describes the ERR_DISABLE fields.

Table 9-31. ERR_DISABLE Field Descriptions

Bits	Name	Description
0–23	—	Reserved
24	ACED	Automatic calibration error disable 0 Automatic calibration errors are enabled. 1 Automatic calibration errors are disabled.
25–27	—	Reserved
28	MBED	Multiple-bit ECC error disable 0 Multiple-bit ECC errors are detected if DDR_SDRAM_CFG[ECC_EN] is set. They are reported if ERR_INT_EN[MBEE] is set. See Section 9.5.11, “Error Management,” for more information. 1 Multiple-bit ECC errors are not detected or reported.
29	SBED	Single-bit ECC error disable 0 Single-bit ECC errors are enabled. 1 Single-bit ECC errors are disabled.
30	—	Reserved
31	MSED	Memory select error disable 0 Memory select errors are enabled. 1 Memory select errors are disabled.

9.4.1.26 Memory Error Interrupt Enable (ERR_INT_EN)

The memory error interrupt enable register, shown in Figure 9-27, enables ECC interrupts or memory select error interrupts. When an enabled interrupt condition occurs, the internal $\overline{int}$ signal is asserted to the programmable interrupt controller (PIC).

Offset 0xE48 Access: Read/Write

	0–23	24	25–27	28	29	30	31
R W	—	ACEE	—	MBEE	SBEE	—	MSEE
Reset	All zeros						

Figure 9-27. Memory Error Interrupt Enable Register (ERR_INT_EN)

Table 9-32 describes the ERR_INT_EN fields.

Table 9-32. ERR_INT_EN Field Descriptions

Bits	Name	Description
0–23	—	Reserved
24	ACEE	Automatic calibration error interrupt enable 0 Automatic calibration errors cannot generate interrupts. 1 Automatic calibration errors generate interrupts.
25–27	—	Reserved
28	MBEE	Multiple-bit ECC error interrupt enable. See Section 9.5.11, “Error Management,” for more information. 0 Multiple-bit ECC errors cannot generate interrupts. 1 Multiple-bit ECC errors generate interrupts.

Table 9-32. ERR_INT_EN Field Descriptions (continued)

Bits	Name	Description
29	SBEE	Single-bit ECC error interrupt enable 0 Single-bit ECC errors cannot generate interrupts. 1 Single-bit ECC errors generate interrupts.
30	—	Reserved
31	MSEE	Memory select error interrupt enable 0 Memory select errors do not cause interrupts. 1 Memory select errors generate interrupts.

9.4.1.27 Memory Error Attributes Capture (CAPTURE_ATTRIBUTES)

The memory error attributes capture register, shown in Figure 9-28, sets attributes for errors including type, size, source, and others.

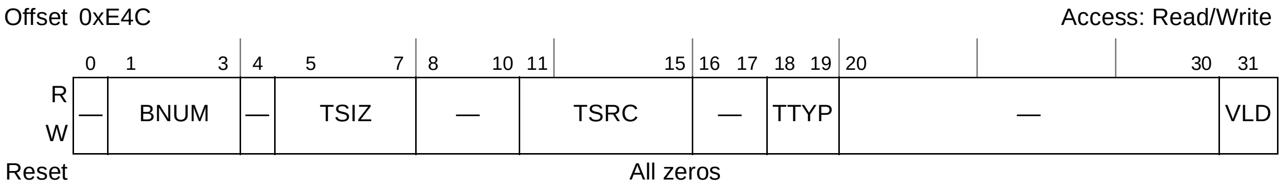


Figure 9-28. Memory Error Attributes Capture Register (CAPTURE_ATTRIBUTES)

Table 9-33 describes the CAPTURE_ATTRIBUTES fields.

Table 9-33. CAPTURE_ATTRIBUTES Field Descriptions

Bits	Name	Description
0	—	Reserved
1–3	BNUM	Data beat number. Captures the double-word number for the detected error. Relevant only for ECC errors.
4	—	Reserved
5–7	TSIZ	Transaction size for the error. Captures the transaction size in double words. 000 4 double words 001 1 double word 010 2 double words 011 3 double words Others Reserved
8–10	—	Reserved

Table 9-33. CAPTURE_ATTRIBUTES Field Descriptions (continued)

Bits	Name	Description	
11–15	TSRC	Transaction source for the error 00000 e300 core data transaction 00001 Reserved 00010 e300 core instruction fetch 00011 Reserved 00100 eTSEC 1 00101 eTSEC 2 00110 Reserved 00111 USB DR 01000 Encryption core 01001I^{2}C (boot sequencer) 01010JTAG 01011 Reserved	01100eSDHC 01101PCI 01110Reserved 01111DMA 10000–10111 Reserved 11000 SATA1 11001 SATA2 11010 SATA3 11011 SATA4 11100 Reserved 11101 PCI Express 1 11110 PCI Express 2 11111 Reserved
16–17	—	Reserved	
18–19	TTYP	Transaction type for the error. 00 Reserved 01 Write 10 Read 11 Read-modify-write	
20–30	—	Reserved	
31	VLD	Valid. Set as soon as valid information is captured in the error capture registers.	

9.4.1.28 Memory Error Address Capture (CAPTURE_ADDRESS)

The memory error address capture register, shown in Figure 9-29, holds the 32 lsbs of a transaction when a DDR ECC error is detected.

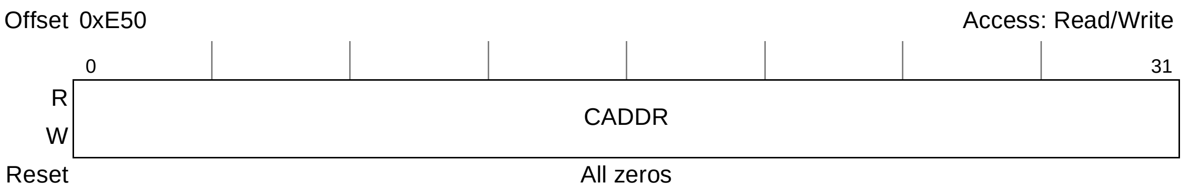


Figure 9-29. Memory Error Address Capture Register (CAPTURE_ADDRESS)

Table 9-34 describes the CAPTURE_ADDRESS fields.

Table 9-34. CAPTURE_ADDRESS Field Descriptions

Bits	Name	Description
0–31	CADDR	Captured address. Captures the 32 lsbs of the transaction address when an error is detected.

9.4.1.29 Single-Bit ECC Memory Error Management (ERR_SBE)

The single-bit ECC memory error management register, shown in Figure 9-30, stores the threshold value for reporting single-bit errors and the number of single-bit errors counted since the last error report. When the counter field reaches the threshold, it wraps back to the reset value (0). If necessary, software must clear the counter after it has managed the error.

Offset 0xE58 Access: Read/Write

	0 … 7	8 … 15	16 … 23	24 … 31
R W	—	SBET	—	SBEC
Reset	All zeros			

Figure 9-30. Single-Bit ECC Memory Error Management Register (ERR_SBE)

Table 9-35 describes the ERR_SBE fields.

Table 9-35. ERR_SBE Field Descriptions

Bits	Name	Description
0–7	—	Reserved
8–15	SBET	Single-bit error threshold. Establishes the number of single-bit errors that must be detected before an error condition is reported.
16–23	—	Reserved
24–31	SBEC	Single-bit error counter. Indicates the number of single-bit errors detected and corrected since the last error report. If single-bit error reporting is enabled, an error is reported and an interrupt is generated when this value equals SBET. SBEC is automatically cleared when the threshold value is reached.

9.5 Functional Description

The DDR SDRAM controller controls processor and I/O interactions with system memory. It provides support for JEDEC-compliant DDR2 and DDR SDRAM. The memory system allows a wide range of memory devices to be mapped to any arbitrary chip select, and support is provided for registered DIMMs and unbuffered DIMMs. However, registered DIMMs cannot be mixed with unbuffered DIMMs.

Figure 9-31 is a high-level block diagram of the DDR memory controller. Requests are received from the internal mastering device and the address is decoded to generate the physical bank, logical bank, row, and column addresses. The transaction is compared with values in the row open table to determine if the address maps to an open page. If the transaction does not map to an open page, an active command is issued.

The memory interface supports the following configurations:

- as many as four physical banks of 64-/72-bit wide or 32-/40-bit wide memory
- bank sizes up to 2 Gbytes

Programmable parameters allow for a variety of memory organizations and timings. Optional error checking and correcting (ECC) protection is provided for the DDR SDRAM data bus. Using ECC, the DDR memory controller detects and corrects all single-bit errors within the 64- or 32-bit data bus, detects all double-bit errors within the 64- or 32-bit data bus, and detects all errors within a nibble. The controller allows as many as 32 pages to be open simultaneously. The amount of time (in clock cycles) the pages remain open is programmable with DDR_SDRAM_INTERVAL[BSTOPRE].

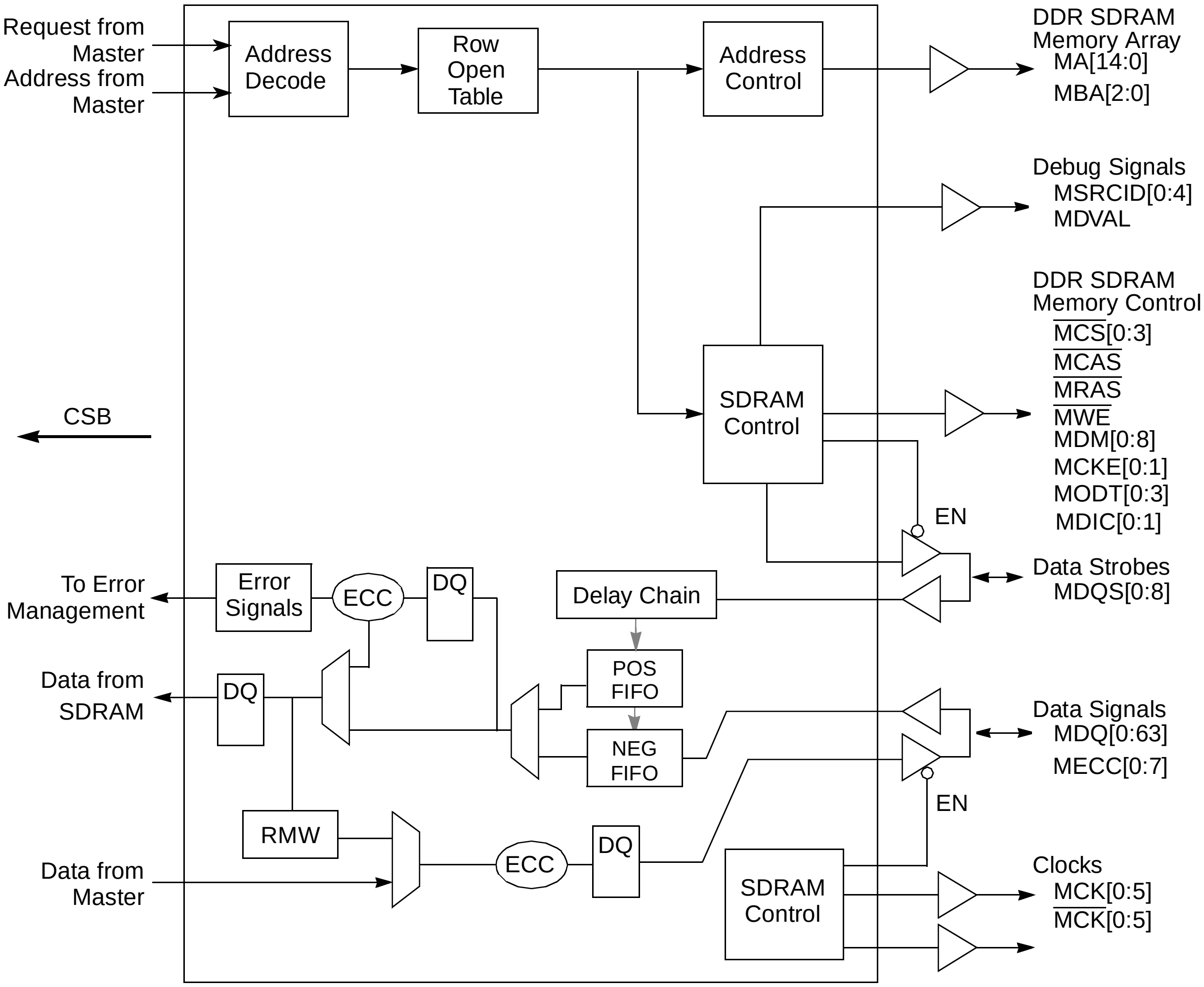


Figure 9-31. DDR Memory Controller Block Diagram

Read and write accesses to memory are burst oriented; accesses start at a selected location and continue for a programmed number of higher locations (4 or 8) in a programmed sequence. Accesses to closed pages start with the registration of an ACTIVE command followed by a READ or WRITE. (Accessing open pages does not require an ACTIVE command.) The address bits registered coincident with the activate command specifies the logical bank and row to be accessed. The address coincident with the READ or WRITE command specify the logical bank and starting column for the burst access.

The data interface is source synchronous, meaning whatever sources the data also provides a clocking signal to synchronize data reception. These bidirectional data strobes (MDQS[0:8]) are inputs to the controller during reads and outputs during writes. The DDR SDRAM specification requires the data strobe signals to be centered within the data tenure during writes and to be offset by the controller to the center of the data tenure during reads. This delay is implemented in the controller for both reads and writes.

When ECC is enabled, 1 clock cycle is added to the read path to check ECC and correct single-bit errors. ECC generation does not add a cycle to the write path.

The address and command interface is also source synchronous, although 1/8 cycle adjustments are provided for adjusting the clock alignment.

Figure 9-32 shows an example DDR SDRAM configuration with four logical banks.

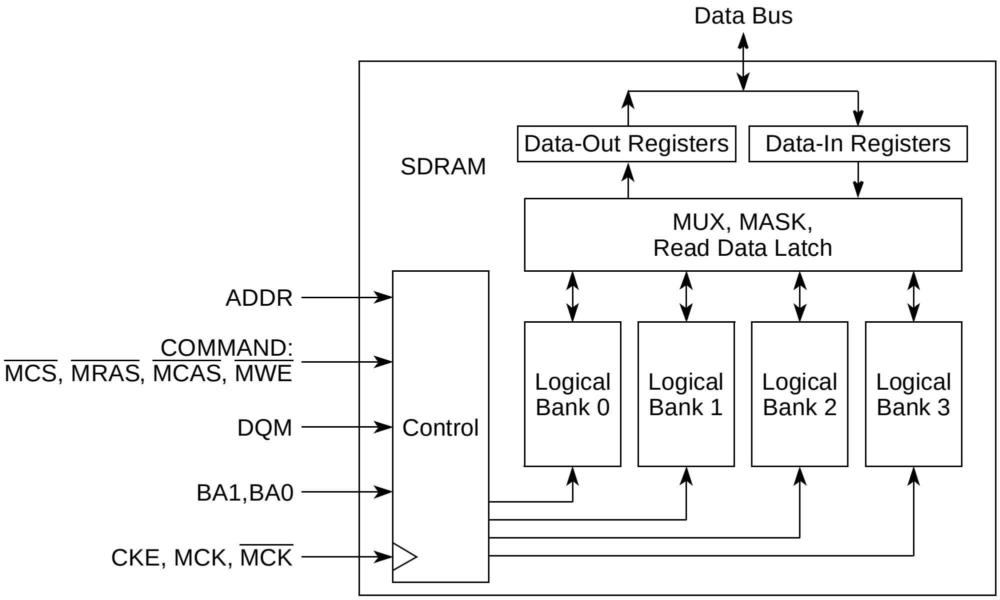


Figure 9-32. Typical Dual Data Rate SDRAM Internal Organization

Figure 9-33 shows some typical signal connections.

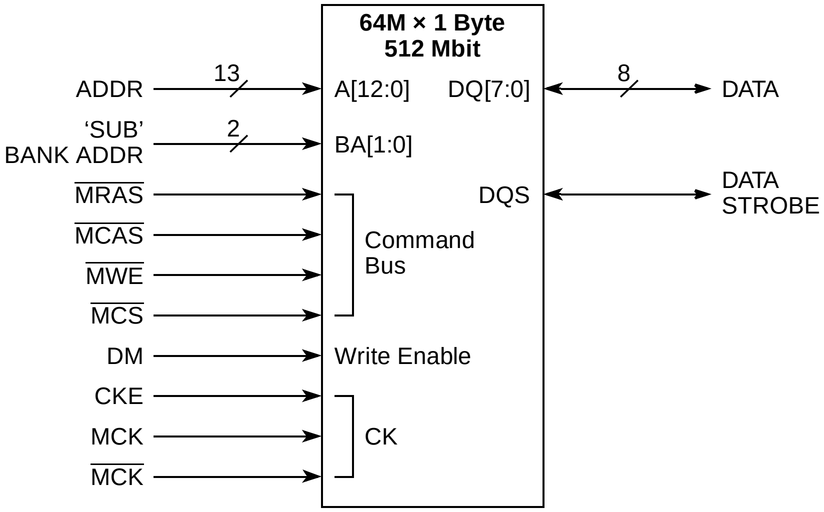
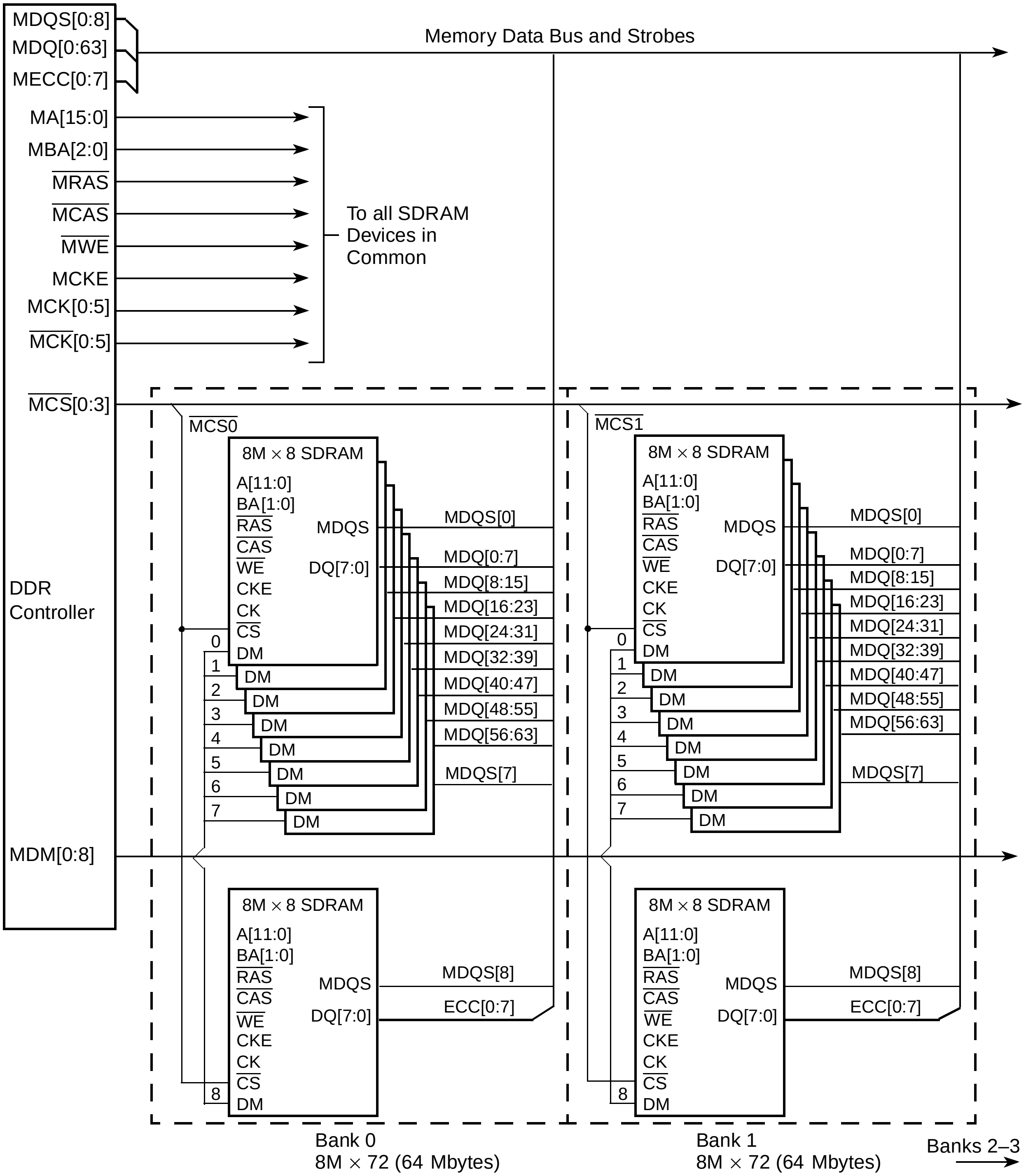


Figure 9-33. Typical DDR SDRAM Interface Signals

Figure 9-34 shows an example DDR SDRAM configuration with four physical banks each comprised of nine 8Mbyte × 8 DDR modules for a total of 256 Mbytes of system memory. One of the nine modules is used for the memory's ECC checking function. Certain address and control lines may require buffering. Analysis of the device's AC timing specifications, desired memory operating frequency, capacitive loads, and board routing loads can assist the system designer in deciding signal buffering requirements. The DDR memory controller drives 15 address pins, but in this example the DDR SDRAM devices use only 12 bits.



1. All signals are connected in common (in parallel) except for $\overline{\text{MCS}}$[0:3], MCK*n*, MDM[0:8], and the data bus signals.
2. Each of the $\overline{\text{MCS}}$[0:3] signals correspond with a separate physical bank of memory.
3. Buffering may be needed if large memory arrays are used.
4. MCK*n* may be apportioned among all memory devices. Complementary bus is not shown.

Figure 9-34. Example 256-Mbyte DDR SDRAM Configuration With ECC

Section 9.5.11, “Error Management,” explains how the DDR memory controller handles errors.

9.5.1 DDR SDRAM Interface Operation

The DDR memory controller supports many different DDR SDRAM configurations. SDRAMs with different sizes can be used in the same system. Fifteen multiplexed address signals and three logical bank select signals support device densities from 64 Mbits to 4 Gbits. Four chip select ($\overline{\text{CS}}$) signals support up to two DIMMs of memory. The DDR SDRAM physical banks can be built from standard memory modules or directly-attached memory devices. The data path to individual physical banks is up to 64 or 32 bits wide, 72 or 40 bits with ECC. The DDR memory controller supports physical bank sizes from 16 Mbytes to 4 Gbytes. The physical banks can be constructed using ×8, ×16, or ×32 memory devices. The memory technologies supported are 64 Mbits, 128 Mbits, 256 Mbits, 512 Mbits, 1 Gbit, 2 Gbits, and 4Gbits. Nine data qualifier (DQM) signals provide byte selection for memory accesses.

NOTE

> An 8-bit DDR SDRAM device has a DQM signal and eight data signals (DQ[0:7]). A 16-bit DDR SDRAM device has two DQM signals associated with specific halves of the 16 data signals (DQ[0:7] and DQ[8:15]).

When ECC is enabled, all memory accesses are performed on double-word boundaries (that is, all DQM signals are set simultaneously). However, when ECC is disabled, the memory system uses the DQM signals for byte lane selection.

Table 9-36 shows the DDR memory controller’s relationships between data byte lane 0–7, MDM[0:7], MDQS[0:7], and MDQ[0:63] when DDR SDRAM memories are used with ×8 or ×16 devices.

Table 9-36. Byte Lane to Data Relationship

Data Byte Lane	Data Bus Mask	Data Bus Strobe	Data Bus 64-Bit Mode
0 (MSB)	MDM[0]	MDQS[0]	MDQ[0:7]
1	MDM[1]	MDQS[1]	MDQ[8:15]
2	MDM[2]	MDQS[2]	MDQ[16:23]
4	MDM[4]	MDQS[4]	MDQ[32:39]
5	MDM[5]	MDQS[5]	MDQ[40:47]
6	MDM[6]	MDQS[6]	MDQ[48:55]
7 (LSB)	MDM[7]	MDQS[7]	MDQ[56:63]

9.5.1.1 Supported DDR SDRAM Organizations

Although the DDR memory controller multiplexes row and column address bits onto 15 memory address signals and 3 logical bank select signals, a physical bank may be implemented with memory devices requiring fewer than 31 address bits. The physical bank may be configured to provide from 12 to 15 row address bits, plus 2 or 3 logical bank-select bits and from 8–11 column address bits.

The following tables describe DDR SDRAM device configurations supported by the DDR memory controller.

NOTE

DDR SDRAM is limited to 30 total address bits.

Table 9-37. Supported DDR1 SDRAM Device Configurations

SDRAM Device	Device Configuration	Row x Column x Sub-Bank Bits	64-Bit Bank Size	Four Banks of Memory
64 Mbits	8 Mbits x 8	12 x 9 x 2	64 Mbytes	256 Mbytes
64 Mbits[1]	4 Mbits x 16	12 x 8 x 2	32 Mbytes	128 Mbytes
128 Mbits	16 Mbits x 8	12 x 10 x 2	128 Mbytes	512 Mbytes
128 Mbits	8 Mbits x 16	12 x 9 x 2	64 Mbytes	256 Mbytes
256 Mbits	32 Mbits x 8	13 x 10 x 2	256 Mbytes	1 Gbyte
256 Mbits	16 Mbits x 16	13 x 9 x 2	128 Mbytes	512 Mbytes
512 Mbits	64 Mbits x 8	13 x 11 x 2	512 Mbytes	2 Gbytes
512 Mbits	32 Mbits x 16	13 x 10 x 2	256 Mbytes	1 Gbyte
1 Gbit	128 Mbits x 8	14 x 11 x 2	1 Gbyte	4 Gbytes
1 Gbit	64 Mbits x 16	14 x 10 x 2	512 Mbytes	2 Gbytes
2 Gbits	256 Mbits x 8	15 x 11 x 2	2 Gbytes	8 Gbytes (split into two banks)
2 Gbits	128 Mbits x 16	15 x 10 x 2	1 Gbyte	4 Gbytes

[1] This configuration is not supported in 16-bit bus mode.

Table 9-38. Supported DDR2 SDRAM Device Configurations

SDRAM Device	Device Configuration	Row x Column x Sub-Bank Bits	64-Bit Bank Size	Four Banks of Memory
256 Mbits	32 Mbits x 8	13 x 10 x 2	256 Mbytes	1 Gbyte
256 Mbits	16 Mbits x 16	13 x 9 x 2	128 Mbytes	512 Mbytes
512 Mbits	64 Mbits x 8	14 x 10 x 2	512 Mbytes	2 Gbytes
512 Mbits	32 Mbits x 16	13 x 10 x 2	256 Mbytes	1 Gbyte
1 Gbit	128 Mbits x 8	14 x 10 x 3	1 Gbyte	4 Gbytes
1 Gbit	64 Mbits x 16	13 x 10 x 3	512 Mbytes	2 Gbytes
2 Gbits	256 Mbits x 8	15 x 10 x 3	2 Gbytes	8 Gbytes (split into two banks)
2 Gbits	128Mbits x 16	14 x 10 x 3	1 Gbyte	4 Gbytes
4 Gbits	256 Mbits x 16	15 x 10 x 3	2 Gbytes	8 Gbytes (split into two banks)

If a transaction request is issued to the DDR memory controller and the address does not lie within any of the programmed address ranges for an enabled chip select, a memory select error is flagged. Errors are described in detail in Section 9.5.11, “Error Management.”

Using a memory-polling algorithm at power-on reset or by querying the JEDEC serial presence detect capability of memory modules, system firmware uses the memory-boundary registers to configure the DDR memory controller to map the size of each bank in memory. The memory controller uses its bank map to assert the appropriate $\overline{\text{MCS}n}$ signal for memory accesses according to the provided bank starting and ending addresses. The memory banks are not required to be mapped to a contiguous address space.

9.5.2 DDR SDRAM Address Multiplexing

The following tables show the address bit encodings for each DDR SDRAM configuration. The address presented at the memory controller signals MA[14:0] use MA[14] as the msb and MA[0] as the lsb. Also, MA[10] is used as the auto-precharge bit in DDR1/DDR2 modes for reads and writes, so the column address can never use MA[10].

Table 9-39. DDR1 Address Multiplexing for 64-Bit Data Bus with Interleaving Disabled

Row x Col		msb	Address from Core Master																											lsb
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29–31
15 x 11 x 2	$\overline{\text{MRAS}}$		14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																	1	0												
	$\overline{\text{MCAS}}$																			11	9	8	7	6	5	4	3	2	1	0	
15 x 10 x 2	$\overline{\text{MRAS}}$			14	13	12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																		1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
14 x 11 x 2	$\overline{\text{MRAS}}$			13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																	1	0												
	$\overline{\text{MCAS}}$																			11	9	8	7	6	5	4	3	2	1	0	
14 x 10 x 2	$\overline{\text{MRAS}}$				13	12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																		1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
13 x 11 x 2	$\overline{\text{MRAS}}$				12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																	1	0												
	$\overline{\text{MCAS}}$																			11	9	8	7	6	5	4	3	2	1	0	
13 x 10 x 2	$\overline{\text{MRAS}}$					12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																		1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
13 x 9 x 2	$\overline{\text{MRAS}}$						12	11	10	9	8	7	6	5	4	3	2	1	0												
	MBA																			1	0										
	$\overline{\text{MCAS}}$																					8	7	6	5	4	3	2	1	0	

Table 9-39. DDR1 Address Multiplexing for 64-Bit Data Bus with Interleaving Disabled (continued)

Row x Col		msb	Address from Core Master																												lsb
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29–31
12 x 10 x 2	$\overline{\text{MRAS}}$						11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																		1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
12 x 9 x 2	$\overline{\text{MRAS}}$							11	10	9	8	7	6	5	4	3	2	1	0												
	MBA																			1	0										
	$\overline{\text{MCAS}}$																					8	7	6	5	4	3	2	1	0	
12 x 8 x 2	$\overline{\text{MRAS}}$								11	10	9	8	7	6	5	4	3	2	1	0											
	MBA																				1	0									
	$\overline{\text{MCAS}}$																						7	6	5	4	3	2	1	0	

Table 9-40. DDR1 Address Multiplexing for 32-Bit Data Bus with Interleaving Disabled

Row x Col		msb	Address from Core Master																													lsb
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30–31
15 x 11 x 2	$\overline{\text{MRAS}}$			14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																		1	0												
	$\overline{\text{MCAS}}$																				11	9	8	7	6	5	4	3	2	1	0	
15 x 10 x 2	$\overline{\text{MRAS}}$				14	13	12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																			1	0											
	$\overline{\text{MCAS}}$																					9	8	7	6	5	4	3	2	1	0	
14 x 11 x 2	$\overline{\text{MRAS}}$				13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																		1	0												
	$\overline{\text{MCAS}}$																				11	9	8	7	6	5	4	3	2	1	0	
14 x 10 x 2	$\overline{\text{MRAS}}$					13	12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																			1	0											
	$\overline{\text{MCAS}}$																					9	8	7	6	5	4	3	2	1	0	
13 x 11 x 2	$\overline{\text{MRAS}}$					12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																		1	0												
	$\overline{\text{MCAS}}$																				11	9	8	7	6	5	4	3	2	1	0	
13 x 10 x 2	$\overline{\text{MRAS}}$						12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																			1	0											
	$\overline{\text{MCAS}}$																					9	8	7	6	5	4	3	2	1	0	
13 x 9 x 2	$\overline{\text{MRAS}}$							12	11	10	9	8	7	6	5	4	3	2	1	0												
	MBA																				1	0										
	$\overline{\text{MCAS}}$																						8	7	6	5	4	3	2	1	0	

Table 9-40. DDR1 Address Multiplexing for 32-Bit Data Bus with Interleaving Disabled (continued)

Row x Col		msb	Address from Core Master																													lsb
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30–31
12 x 10 x 2	$\overline{\text{MRAS}}$							11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																			1	0											
	$\overline{\text{MCAS}}$																					9	8	7	6	5	4	3	2	1	0	
12 x 9 x 2	$\overline{\text{MRAS}}$								11	10	9	8	7	6	5	4	3	2	1	0												
	MBA																				1	0										
	$\overline{\text{MCAS}}$																						8	7	6	5	4	3	2	1	0	
12 x 8 x 2	$\overline{\text{MRAS}}$									11	10	9	8	7	6	5	4	3	2	1	0											
	MBA																					1	0									
	$\overline{\text{MCAS}}$																							7	6	5	4	3	2	1	0	

Table 9-41. DDR1 Address Multiplexing for 16-Bit Data Bus

Row x Col		msb	Address from Core Master																													lsb
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
15 x 11 x 2	$\overline{\text{MRAS}}$				14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																			1	0												
	$\overline{\text{MCAS}}$																					11	9	8	7	6	5	4	3	2	1	0	
15 x 10 x 2	$\overline{\text{MRAS}}$					14	13	12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																				1	0											
	$\overline{\text{MCAS}}$																						9	8	7	6	5	4	3	2	1	0	
14 x 11 x 2	$\overline{\text{MRAS}}$					13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																			1	0												
	$\overline{\text{MCAS}}$																					11	9	8	7	6	5	4	3	2	1	0	
14 x 10 x 2	$\overline{\text{MRAS}}$						13	12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																				1	0											
	$\overline{\text{MCAS}}$																						9	8	7	6	5	4	3	2	1	0	
13 x 11 x 2	$\overline{\text{MRAS}}$						12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																			1	0												
	$\overline{\text{MCAS}}$																					11	9	8	7	6	5	4	3	2	1	0	
13 x 10 x 2	$\overline{\text{MRAS}}$							12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																				1	0											
	$\overline{\text{MCAS}}$																						9	8	7	6	5	4	3	2	1	0	

Table 9-41. DDR1 Address Multiplexing for 16-Bit Data Bus (continued)

Row x Col		msb	Address from Core Master																														lsb
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
13 x 9 x 2	$\overline{\text{MRAS}}$								12	11	10	9	8	7	6	5	4	3	2	1	0												
	MBA																					1	0										
	$\overline{\text{MCAS}}$																							8	7	6	5	4	3	2	1	0	
12 x 10 x 2	$\overline{\text{MRAS}}$								11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																				1	0											
	$\overline{\text{MCAS}}$																						9	8	7	6	5	4	3	2	1	0	
12 x 9 x 2	$\overline{\text{MRAS}}$									11	10	9	8	7	6	5	4	3	2	1	0												
	MBA																					1	0										
	$\overline{\text{MCAS}}$																							8	7	6	5	4	3	2	1	0	
12 x 8 x 2	$\overline{\text{MRAS}}$										11	10	9	8	7	6	5	4	3	2	1	0											
	MBA																						1	0									
	$\overline{\text{MCAS}}$																								7	6	5	4	3	2	1	0	

Table 9-42. DDR2 Address Multiplexing for 64-Bit Data Bus with Interleaving Disabled

Row x Col		msb	Address from Core Master																												lsb
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29–31**
15 x 10 x 3	$\overline{\text{MRAS}}$		14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																	2	1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
14 x 10 x 3	$\overline{\text{MRAS}}$			13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																	2	1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
14 x 10 x 2	$\overline{\text{MRAS}}$				13	12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																		1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
13 x 10 x 3	$\overline{\text{MRAS}}$				12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																	2	1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	

Table 9-42. DDR2 Address Multiplexing for 64-Bit Data Bus with Interleaving Disabled (continued)

Row x Col		msb	Address from Core Master																												lsb
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29–31
13 x 10 x 2	$\overline{\text{MRAS}}$					12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																		1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
13 x 9 x 2	$\overline{\text{MRAS}}$						12	11	10	9	8	7	6	5	4	3	2	1	0												
	MBA																			1	0										
	$\overline{\text{MCAS}}$																					8	7	6	5	4	3	2	1	0	

Table 9-43. DDR2 Address Multiplexing for 32-Bit Data Bus with Interleaving Disabled

Row x Col		msb	Address from Core Master																													lsb
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30–31
15 x 11 x 3	$\overline{\text{MRAS}}$		14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
	MBA																	2	1	0												
	$\overline{\text{MCAS}}$																				11	9	8	7	6	5	4	3	2	1	0	
15 x 10 x 3	$\overline{\text{MRAS}}$			14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																		2	1	0											
	$\overline{\text{MCAS}}$																					9	8	7	6	5	4	3	2	1	0	
14 x 10 x 3	$\overline{\text{MRAS}}$				13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																		2	1	0											
	$\overline{\text{MCAS}}$																					9	8	7	6	5	4	3	2	1	0	
14 x 10 x 2	$\overline{\text{MRAS}}$					13	12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																			1	0											
	$\overline{\text{MCAS}}$																					9	8	7	6	5	4	3	2	1	0	
13 x 10 x 3	$\overline{\text{MRAS}}$					12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																		2	1	0											
	$\overline{\text{MCAS}}$																					9	8	7	6	5	4	3	2	1	0	
13 x 10 x 2	$\overline{\text{MRAS}}$						12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																			1	0											
	$\overline{\text{MCAS}}$																					9	8	7	6	5	4	3	2	1	0	
13 x 9 x 2	$\overline{\text{MRAS}}$							12	11	10	9	8	7	6	5	4	3	2	1	0												
	MBA																				1	0										
	$\overline{\text{MCAS}}$																						8	7	6	5	4	3	2	1	0	

Section, Page No. **Changes**

Table 9-44. DDR2 Address Multiplexing for 16-Bit Data Bus

Row x Col		msb	Address from Core Master																														lsb
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
15 x 10 x 3	$\overline{\text{MRAS}}$				14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																			2	1	0											
	$\overline{\text{MCAS}}$																						9	8	7	6	5	4	3	2	1	0	
14 x 10 x 3	$\overline{\text{MRAS}}$					13	12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																			2	1	0											
	$\overline{\text{MCAS}}$																						9	8	7	6	5	4	3	2	1	0	
14 x 10 x 2	$\overline{\text{MRAS}}$						13	12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																				1	0											
	$\overline{\text{MCAS}}$																						9	8	7	6	5	4	3	2	1	0	
13 x 10 x 3	$\overline{\text{MRAS}}$						12	11	10	9	8	7	6	5	4	3	2	1	0														
	MBA																			2	1	0											
	$\overline{\text{MCAS}}$																						9	8	7	6	5	4	3	2	1	0	
13 x 10 x 2	$\overline{\text{MRAS}}$							12	11	10	9	8	7	6	5	4	3	2	1	0													
	MBA																				1	0											
	$\overline{\text{MCAS}}$																						9	8	7	6	5	4	3	2	1	0	
13 x 9 x 2	$\overline{\text{MRAS}}$								12	11	10	9	8	7	6	5	4	3	2	1	0												
	MBA																					1	0										
	$\overline{\text{MCAS}}$																							8	7	6	5	4	3	2	1	0	

Chip select interleaving is supported for the memory controller, and is programmed in DDR_SDRAM_CFG[BA_INTLV_CTL]. Interleaving is supported between chip selects 0 and 1 or chip selects 2 and 3. In addition, interleaving between all four chip selects can be enabled. When interleaving is enabled, the chip selects being interleaved must use the same size of memory. If two chip selects are interleaved, then 1 extra bit in the address decode is used for the interleaving to determine which chip select to access. If four chip selects are interleaved, then two extra bits are required in the address decode.

The following tables show examples of interleaving between chip selects.

Table 9-45. Example of Address Multiplexing for 64-Bit Data Bus Interleaving between Two Banks

Row x Col		msb	Address from Core Master																												lsb
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29–31
14 x 10 x 3	$\overline{\text{MRAS}}$		13	12	11	10	9	8	7	6	5	4	3	2	1	0	CS SEL														
	MBA																	2	1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
14 x 10 x 2	$\overline{\text{MRAS}}$			13	12	11	10	9	8	7	6	5	4	3	2	1	0	CS SEL													
	MBA																		1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
13 x 10 x 3	$\overline{\text{MRAS}}$			12	11	10	9	8	7	6	5	4	3	2	1	0	CS SEL														
	MBA																	2	1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
13 x 10 x 2	$\overline{\text{MRAS}}$				12	11	10	9	8	7	6	5	4	3	2	1	0	CS SEL													
	MBA																		1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	

Table 9-46. Example of Address Multiplexing for 32-Bit Data Bus Interleaving between Two Banks

Row x Col		msb	Address from Core Master																												lsb
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30–31
14 x 10 x 3	$\overline{\text{MRAS}}$		13	12	11	10	9	8	7	6	5	4	3	2	1	0	CS SEL														
	MBA																	2	1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
14 x 10 x 2	$\overline{\text{MRAS}}$			13	12	11	10	9	8	7	6	5	4	3	2	1	0	CS SEL													
	MBA																		1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
13 x 10 x 3	$\overline{\text{MRAS}}$			12	11	10	9	8	7	6	5	4	3	2	1	0	CS SEL														
	MBA																	2	1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
13 x 10 x 2	$\overline{\text{MRAS}}$				12	11	10	9	8	7	6	5	4	3	2	1	0	CS SEL													
	MBA																		1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	

Table 9-47. Example of Address Multiplexing for 64-Bit Data Bus Interleaving between Four Banks

Row x Col		msb	Address from Core Master																												lsb
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29–31
14 x 10 x 3	$\overline{\text{MRAS}}$	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CS SEL															
	MBA																	2	1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
14 x 10 x 2	$\overline{\text{MRAS}}$		13	12	11	10	9	8	7	6	5	4	3	2	1	0	CS SEL														
	MBA																		1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
13 x 10 x 3	$\overline{\text{MRAS}}$		12	11	10	9	8	7	6	5	4	3	2	1	0	CS SEL															
	MBA																	2	1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	
13 x 10 x 2	$\overline{\text{MRAS}}$			12	11	10	9	8	7	6	5	4	3	2	1	0	CS SEL														
	MBA																		1	0											
	$\overline{\text{MCAS}}$																				9	8	7	6	5	4	3	2	1	0	

9.5.3 JEDEC Standard DDR SDRAM Interface Commands

The following section describes the commands and timings the controller uses when operating in DDR2 or DDR modes.

All read or write accesses to DDR SDRAM are performed by the DDR memory controller using JEDEC standard DDR SDRAM interface commands. The SDRAM device samples command and address inputs on rising edges of the memory clock; data is sampled using both the rising and falling edges of DQS. Data read from the DDR SDRAM is also sampled on both edges of DQS.

The following DDR SDRAM interface commands (summarized in Table 9-48) are provided by the DDR controller. All actions for these commands are described from the perspective of the SDRAM device.

- Row activate—Latches row address and initiates memory read of that row. Row data is latched in SDRAM sense amplifiers and must be restored by a precharge command before another row activate occurs.
- Precharge—Restores data from the sense amplifiers to the appropriate row. Also initializes the sense amplifiers in preparation for reading another row in the memory array (performing another activate command). Precharge must occur after read or write, if the row address changes on the next open page mode access.
- Read—Latches column address and transfers data from the selected sense amplifier to the output buffer as determined by the column address. During each succeeding clock edge, additional data is driven without additional read commands. The amount of data transferred is determined by the burst size which defaults to 4.

- Write—Latches column address and transfers data from the data pins to the selected sense amplifier as determined by the column address. During each succeeding clock edge, additional data is transferred to the sense amplifiers from the data pins without additional write commands. The amount of data transferred is determined by the data masks and the burst size, which is set to four by the DDR memory controller.
- Refresh (similar to $\overline{\text{MCAS}}$ before $\overline{\text{MRAS}}$)—Causes a row to be read in all logical banks (JEDEC SDRAM) as determined by the refresh row address counter. This refresh row address counter is internal to the SDRAM. After being read, the row is automatically rewritten in the memory array. All logical banks must be in a precharged state before executing a refresh. The memory controller also supports posted refreshes, where several refreshes may be executed at once, and the refresh interval may be extended.
- Mode register set (for configuration)—Allows setting of DDR SDRAM options. These options are: $\overline{\text{MCAS}}$ latency, additive latency (for DDR2), write recovery (for DDR2), burst type, and burst length. $\overline{\text{MCAS}}$ latency may be chosen as provided by the preferred SDRAM (some SDRAMs provide $\overline{\text{MCAS}}$ latency {1,2,3}, some provide $\overline{\text{MCAS}}$ latency {1,2,3,4,5}, and so on). Burst type is always sequential. Although some SDRAMs provide burst lengths of 1, 2, 4, 8, and page size, this memory controller supports a burst length of 4. A burst length of 8 is supported for DDR1 memory only. For DDR2 in 32-bit bus mode, all 32-byte burst accesses from the platform are split into two 16-byte (that is, 4-beat) accesses to the SDRAMs in the memory controller. The mode register set command is performed by the DDR memory controller during system initialization. Parameters such as mode register data, $\overline{\text{MCAS}}$ latency, burst length, and burst type, are set by software in DDR_SDRAM_MODE[SDMODE] and transferred to the SDRAM array by the DDR memory controller after DDR_SDRAM_CFG[MEM_EN] is set. If DDR_SDRAM_CFG[BI] is set to bypass the automatic initialization, then the MODE registers can be configured through software through use of the DDR_SDRAM_MD_CNTL register.
- Self refresh (for long periods of standby)—Used when the device is in standby for very long periods of time. Automatically generates internal refresh cycles to keep the data in all memory banks refreshed. Before execution of this command, the DDR controller places all logical banks in a precharged state.

Table 9-48. DDR SDRAM Command Table

Operation	CKE Prev.	CKE Current	$\overline{\text{MCS}}$	$\overline{\text{MRAS}}$	$\overline{\text{MCAS}}$	$\overline{\text{MWE}}$	MBA	MA10	MA
Activate	H	H	L	L	H	H	Logical bank select	Row	Row
Precharge select logical bank	H	H	L	L	H	L	Logical bank select	L	X
Precharge all logical banks	H	H	L	L	H	L	X	H	X
Read	H	H	L	H	L	H	Logical bank select	L	Column
Read with auto-precharge	H	H	L	H	L	H	Logical bank select	H	Column
Write	H	H	L	H	L	L	Logical bank select	L	Column

Table 9-48. DDR SDRAM Command Table (continued)

Operation	**CKE Prev.**	**CKE Current**	**$\overline{\text{MCS}}$**	**$\overline{\text{MRAS}}$**	**$\overline{\text{MCAS}}$**	**$\overline{\text{MWE}}$**	**MBA**	**MA10**	**MA**
Write with auto-precharge	H	H	L	H	L	L	Logical bank select	H	Column
Mode register set	H	H	L	L	L	L	Opcode	Opcode	Opcode and mode
Auto refresh	H	H	L	L	L	H	X	X	X
Self refresh	H	L	L	L	L	H	X	X	X

9.5.4 DDR SDRAM Interface Timing

The DDR memory controller supports four-beat bursts to SDRAM. For single-beat reads, the DDR memory controller performs a four- (or eight-) beat burst read, but ignores the last three (or seven) beats. Single-beat writes are performed by masking the last three (or seven) beats of the four- (or eight-) beat burst using the data mask MDM[0:8]. If ECC is disabled, writes smaller than double words are performed by appropriately activating the data mask. If ECC is enabled, the controller performs a read-modify write.

NOTE

If a second read or write is pending, reads shorter than four beats are not terminated early even if some data is irrelevant.

To accommodate available memory technologies across a wide spectrum of operating frequencies, the DDR memory controller allows the setting of the intervals defined in Table 9-49 with granularity of one memory clock cycle, except for CASLAT, which can be programmed with ½ clock granularity.

Table 9-49. DDR SDRAM Interface Timing Intervals

Timing Intervals	**Definition**
ACTTOACT	The number of clock cycles from a bank-activate command until another bank-activate command within a physical bank. This interval is listed in the AC specifications of the SDRAM as t_{RRD}.
ACTTOPRE	The number of clock cycles from an activate command until a precharge command is allowed. This interval is listed in the AC specifications of the SDRAM as t_{RAS}.
ACTTORW	The number of clock cycles from an activate command until a read or write command is allowed. This interval is listed in the AC specifications of the SDRAM as t_{RCD}.
BSTOPRE	The number of clock cycles to maintain a page open after an access. The page open duration counter is reloaded with BSTOPRE each time the page is accessed (including page hits). When the counter expires, the open page is closed with an SDRAM precharge bank command as soon as possible.
CASLAT	Used in conjunction with additive latency to obtain the READ latency. The number of clock cycles between the registration of a READ command by the SDRAM and the availability of the first piece of output data. If a READ command is registered at clock edge n, and the read latency is m clocks, the data is available nominally coincident with clock edge $n + m$.
PRETOACT	The number of clock cycles from a precharge command until an activate or a refresh command is allowed. This interval is listed in the AC specifications of the SDRAM as t_{RP}.

Table 9-49. DDR SDRAM Interface Timing Intervals (continued)

Timing Intervals	Definition
REFINT	Refresh interval. Represents the number of memory bus clock cycles between refresh cycles. Depending on DDR_SDRAM_CFG_2[NUM_PR], some number of rows are refreshed in each SDRAM bank during each refresh cycle. The value of REFINT depends on the specific SDRAMs used and the frequency of the interface as t_{RP}.
REFREC	The number of clock cycles from the refresh command until an activate command is allowed. This can be calculated by referring to the AC specification of the SDRAM device. The AC specification indicates a maximum refresh-to-activate interval in nanoseconds.
WR_DATA_DELAY	Provides different options for the timing between a write command and the write data strobe. This allows write data to be sent later than the nominal time to meet the SDRAM timing requirement between the registration of a write command and the reception of a data strobe associated with the write command. The specification dictates that the data strobe may not be received earlier than 75% of a cycle, or later than 125% of a cycle, from the registration of a write command. This parameter is not defined in the SDRAM specification. It is implementation-specific, defined for the DDR memory controller in TIMING_CFG_2.
WRREC	The number of clock cycles from the last beat of a write until a precharge command is allowed. This interval, write recovery time, is listed in the AC specifications of the SDRAM as t_{WR}.
WRTORD	Last write pair to read command. Controls the number of clock cycles from the last write data pair to the subsequent read command to the same bank as t_{WTR}.

The value of the above parameters (in whole clock cycles) must be set by boot code at system start-up (in the TIMING_CFG_0, TIMING_CFG_1, TIMING_CFG_2, and TIMING_CFG_3 registers as described in Section 9.4.1.4, “DDR SDRAM Timing Configuration 0 (TIMING_CFG_0),” Section 9.4.1.5, “DDR SDRAM Timing Configuration 1 (TIMING_CFG_1),” Section 9.4.1.6, “DDR SDRAM Timing Configuration 2 (TIMING_CFG_2),” and Section 9.4.1.3, “DDR SDRAM Timing Configuration 3 (TIMING_CFG_3)”) and be kept in the DDR memory controller configuration register space.

The following figures show SDRAM timing for various types of accesses. System software is responsible (at reset) for optimally configuring SDRAM timing parameters. The programmable timing parameters apply to both read and write timing configuration. The configuration process must be completed and the DDR SDRAM initialized before any accesses to SDRAM are attempted.

Figure 9-35 through Figure 9-37 show DDR SDRAM timing for various types of accesses; see Figure 9-35 for a single-beat read operation, Figure 9-36 for a single-beat write operation, and Figure 9-37 for a double word write operation. Note that all signal transitions occur on the rising edge of the memory bus clock and that single-beat read operations are identical to burst-reads. These figures assume the CLK_ADJUST is set to 1/2 DRAM cycle, an additive latency of 0 DRAM cycles is used, and the write latency is 1 DRAM cycle (for DDR1).

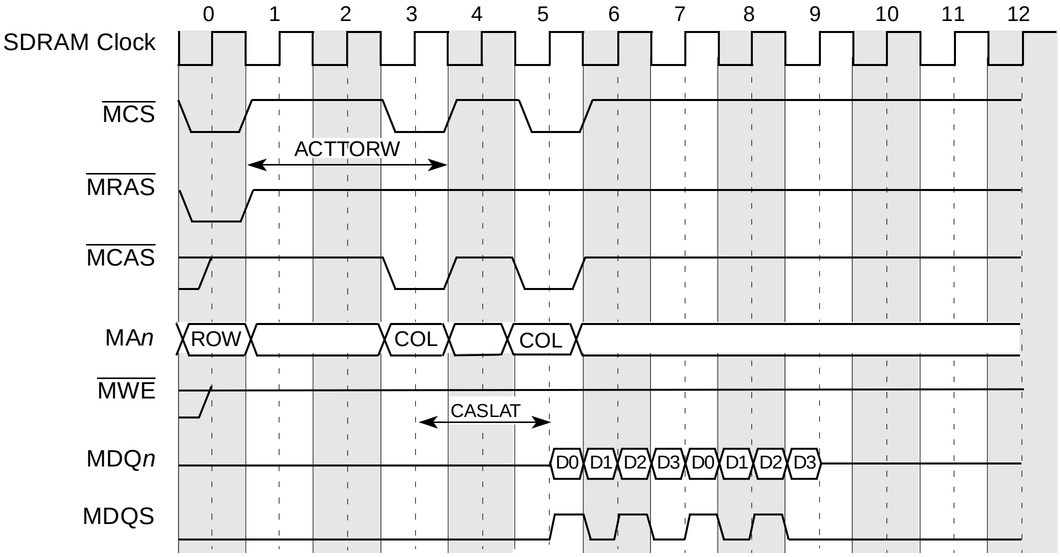


Figure 9-35. DDR SDRAM Burst Read Timing—ACTTORW = 3, $\overline{\text{MCAS}}$ Latency = 2

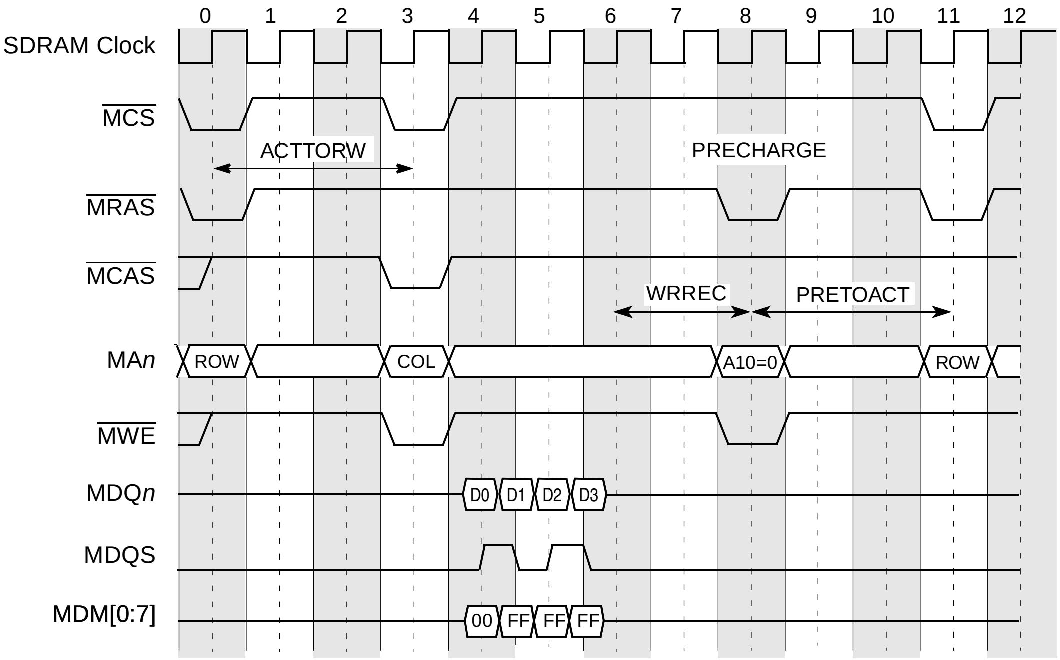


Figure 9-36. DDR SDRAM Single-Beat (Double Word) Write Timing—ACTTORW = 3

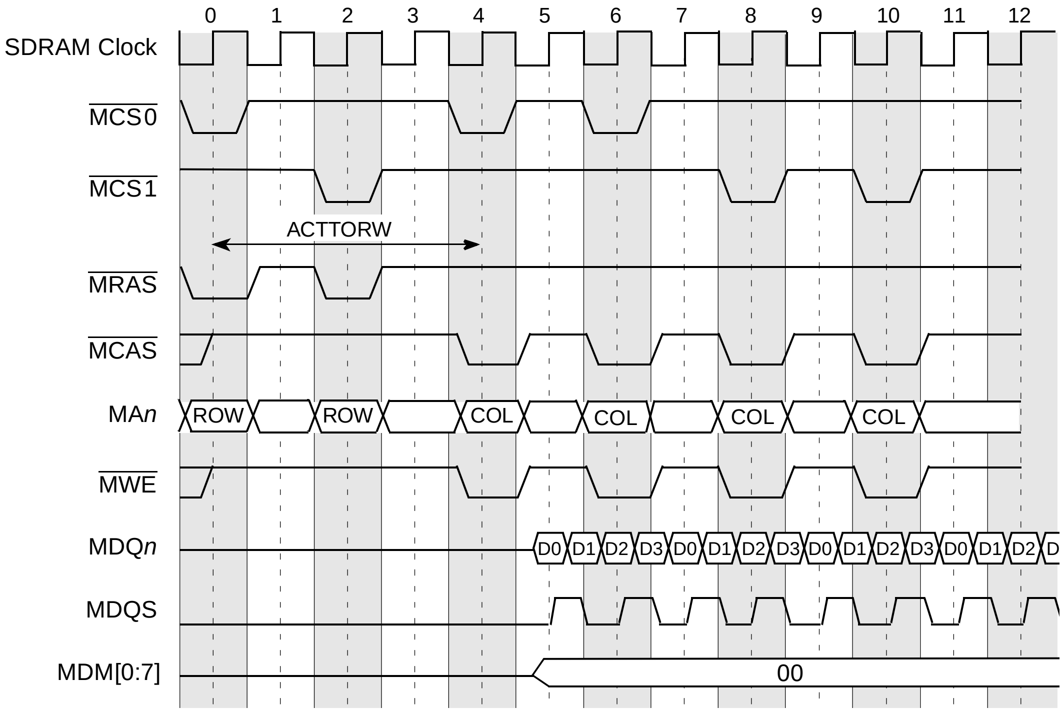


Figure 9-37. DDR SDRAM 4-Beat Burst Write Timing—ACTTORW = 4

9.5.5 DDR SDRAM Registered DIMM Mode

To reduce loading, registered DIMMs latch the DDR SDRAM control signals internally before using them to access the array. Setting DDR_SDRAM_CFG[RD_EN] compensates for this delay on the DIMMs' control bus by delaying the data and data mask writes (on SDRAM buses) by an extra SDRAM clock cycle.

NOTE

Application system board must assert the reset signal on DDR memory devices until software is able to program the DDR memory controller configuration registers, and must deassert the reset signal on DDR memory devices before DDR_SDRAM_CFG[MEM_EN] is set. This ensures that the DDR memory devices are held in reset until a stable clock is provided and, further, that a stable clock is provided before memory devices are released from reset.

Figure 9-38 shows the registered DDR SDRAM DIMM burst write timing.

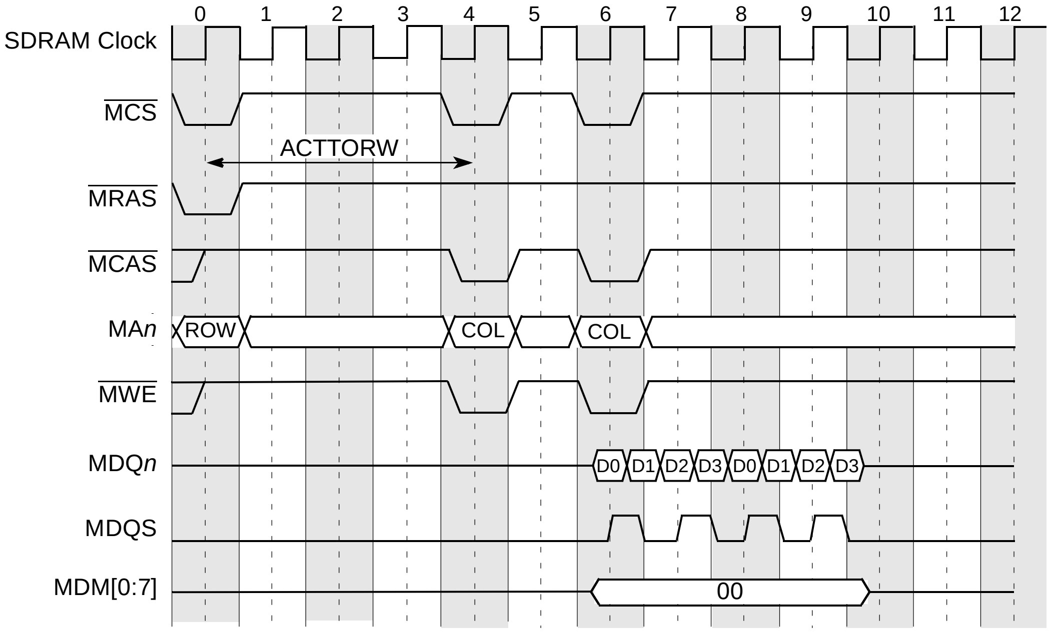


Figure 9-38. Registered DDR SDRAM DIMM Burst Write Timing

9.5.6 DDR SDRAM Write Timing Adjustments

The DDR memory controller facilitates system design flexibility by providing a write timing adjustment parameter, write data delay, (TIMING_CFG_2[WR_DATA_DELAY]) for data and DQS. The DDR SDRAM specification requires DQS be received no sooner than 75% of an SDRAM clock period—and no later than 125% of a clock period—from the capturing clock edge of the command/address at the SDRAM. The WR_DATA_DELAY parameter may be used to meet this timing requirement for a variety of system configurations, ranging from a system with one DIMM to a fully populated system with two DIMMs. TIMING_CFG_2[WR_DATA_DELAY] specifies how much to delay the launching of DQS and data from the first clock edge occurring one SDRAM clock cycle after the command is launched. The delay increment step sizes are in 1/4 SDRAM clock periods starting with the default value of 0.

Figure 9-39 shows the use of the WR_DATA_DELAY parameter.

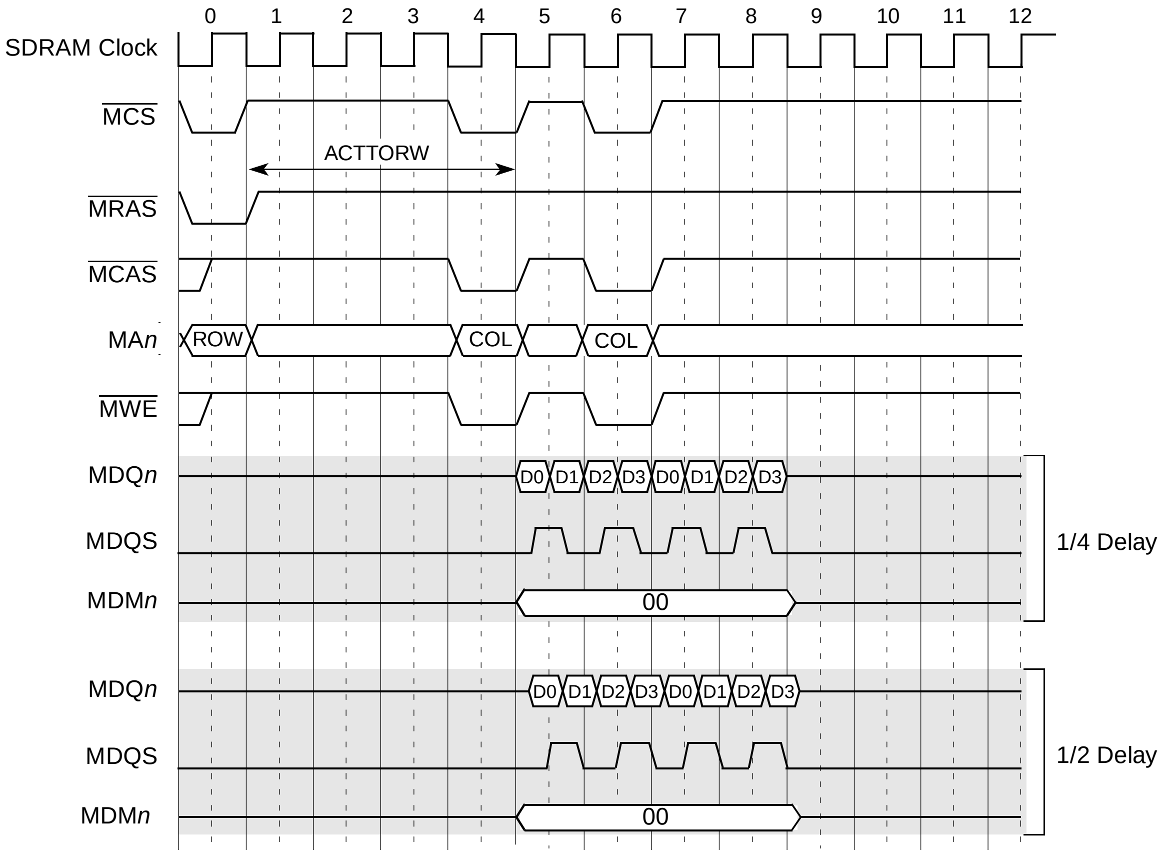


Figure 9-39. Write Timing Adjustments Example for Write Latency = 1

9.5.7 DDR SDRAM Refresh

The DDR memory controller supports auto-refresh and self-refresh. Auto-refresh is used during normal operation and is controlled by the DDR_SDRAM_INTERVAL[REFINT] value; self-refresh is used only when the DDR memory controller is set to enter a sleep power management state. The REFINT value, which represents the number of memory bus clock cycles between refresh cycles, must allow for possible outstanding transactions to complete before a refresh request is sent to the memory after the REFINT value is reached. If a memory transaction is in progress when the refresh interval is reached, the refresh cycle waits for the transaction to complete. In the worst case, the refresh cycle must wait the number of bus clock cycles required by the longest programmed access. To ensure that the latency caused by a memory transaction does not violate the device refresh period, it is recommended that the programmed value of REFINT be less than that required by the SDRAM.

When a refresh cycle is required, the DDR memory controller does the following:

1. Completes all current memory requests.
2. Closes all open pages with a PRECHARGE-ALL command to each DDR SDRAM bank with an open page (as indicated by the row open table).
3. Issues one or more auto-refresh commands to each DDR SDRAM bank (as identified by its chip select) to refresh one row in each logical bank of the selected physical bank.

The auto-refresh commands are staggered across the four possible banks to reduce the system's instantaneous power requirements. Three sets of auto refresh commands are issued on consecutive cycles when the memory is fully populated with two DIMMs. The initial PRECHARGE-ALL commands are also staggered in three groups for convenience. It is important to note that when entering self-refresh mode, only one refresh command is issued simultaneously to all physical banks. For this entire refresh sequence, no cycle optimization occurs for the usual case where fewer than four banks are installed. After the refresh sequence completes, any pending memory request is initiated after an inactive period specified by TIMING_CFG_1 [REFREC] and TIMING_CFG_3[EXT_REFREC]. In addition, posted refreshes are supported to allow the refresh interval to be set to a larger value.

9.5.7.1 DDR SDRAM Refresh Timing

Refresh timing for the DDR SDRAM is controlled by the programmable timing parameter TIMING_CFG_1 [REFREC], which specifies the number of memory bus clock cycles from the refresh command until a logical bank activate command is allowed. The DDR memory controller implements bank staggering for refreshes, as shown in Figure 9-40 (TIMING_CFG_1 [REFREC] = 10 in this example).

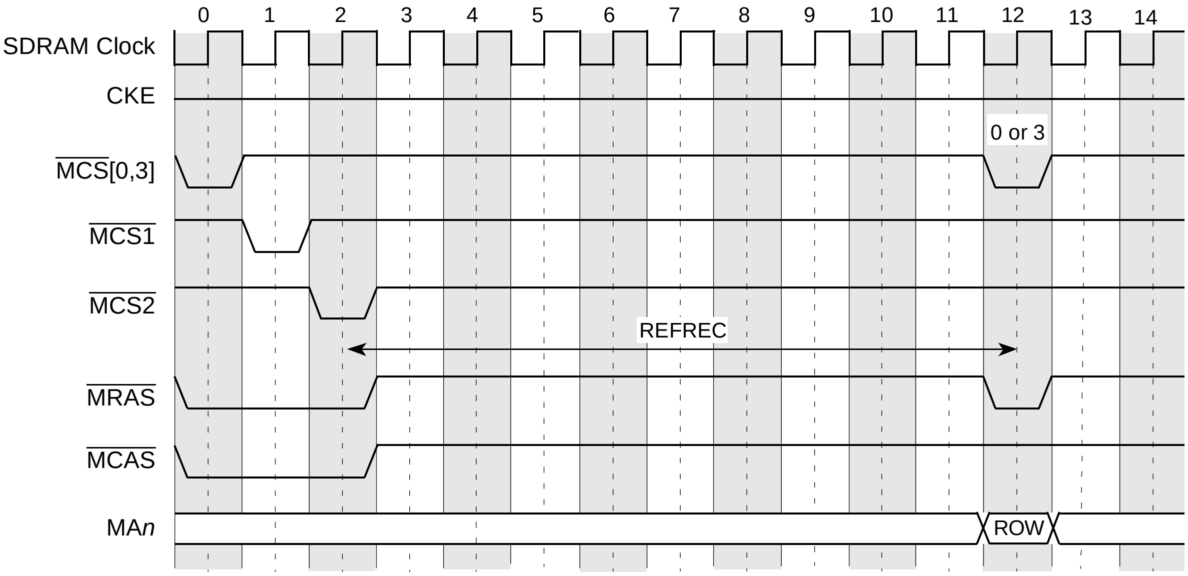


Figure 9-40. DDR SDRAM Bank Staggered Auto Refresh Timing

System software is responsible for optimal configuration of TIMING_CFG_1 [REFREC] and TIMING_CFG_3[EXT_REFREC] at reset. Configuration must be completed before DDR SDRAM accesses are attempted.

9.5.7.2 DDR SDRAM Refresh and Power-Saving Modes

In full-on mode, the DDR memory controller supplies the normal auto refresh to SDRAM. In sleep mode, the DDR memory controller can be configured to take advantage of self-refreshing SDRAMs or to provide no refresh support. Self-refresh support is enabled with the SREN memory control parameter.

Table 9-50 summarizes the refresh types available in each power-saving mode.

Table 9-50. DDR SDRAM Power-Saving Modes Refresh Configuration

Power Saving Mode	Refresh Type	SREN
Sleep	Self	1
	None	—

Note that in the absence of refresh support, system software must preserve DDR SDRAM data (such as by copying the data to disk) before entering the power-saving mode.

The dynamic power-saving mode uses the CKE DDR SDRAM pin to dynamically power down when there is no system memory activity. The CKE pin is negated when both of the following conditions are met:

- No memory refreshes are scheduled
- No memory accesses are scheduled

CKE is reasserted when a new access or refresh is scheduled or the dynamic power mode is disabled. This mode is controlled with DDR_SDRAM_CFG[DYN_PWR_MGMT].

Dynamic power management mode offers tight control of the memory system's power consumption by trading power for performance through the use of CKE. Powering up the DDR SDRAM when a new memory reference is scheduled causes an access latency penalty, depending on whether active or precharge powerdown is used, along with the settings of TIMING_CFG_0[ACT_PD_EXIT] and TIMING_CFG_0[PRE_PD_EXIT]. For example, a penalty of 1 cycle is shown in Figure 9-41.

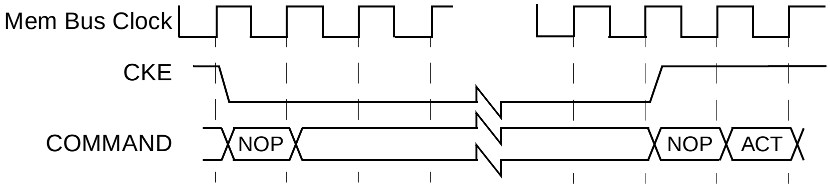


Figure 9-41. DDR SDRAM Power-Down Mode

9.5.7.2.1 Self-Refresh in Sleep Mode

The entry and exit timing for self-refreshing SDRAMs is shown in the following figures.

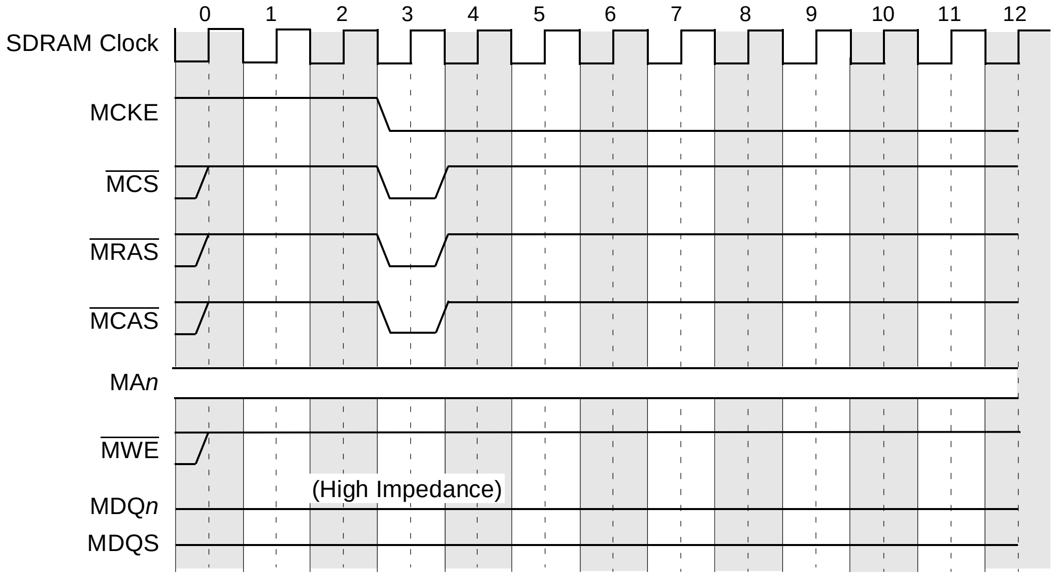


Figure 9-42. DDR SDRAM Self-Refresh Entry Timing

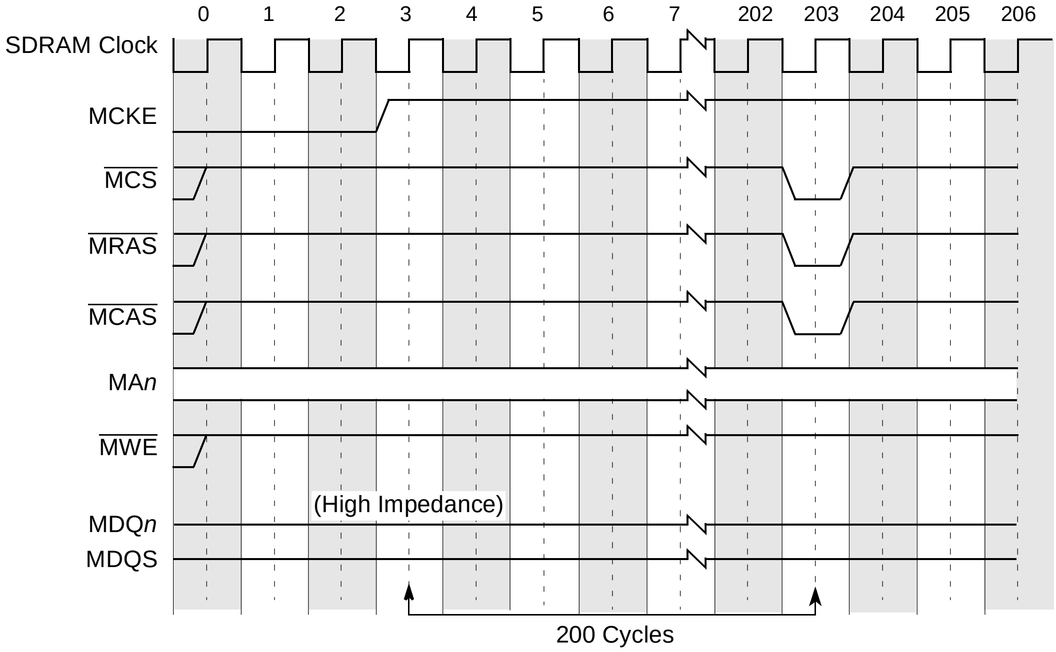


Figure 9-43. DDR SDRAM Self-Refresh Exit Timing

9.5.8 DDR Data Beat Ordering

Transfers to and from memory are always performed in four- or eight-beat bursts (four beats = 32 bytes when a 64-bit bus is used). For transfer sizes other than four or eight beats, the data transfers are still operated as four- or eight-beat bursts. If ECC is enabled and either the access is not double-word aligned or the size is not a multiple of a double word, a full read-modify-write is performed for a write to SDRAM. If ECC is disabled or both the access is double-word aligned with a size that is a multiple of a double word, the data masks MDM[0:8] (MDM[0:] for 32-bit bus) can be used to prevent the writing of unwanted data to SDRAM. The DDR memory controller also uses data masks to prevent all unintended full double words from writing to SDRAM. For example, if a write transaction is desired with a size of one double word (8 bytes), then the second, third, and fourth beats of data are not written to DRAM, as the width of the data bus is 64 bits.

Table 9-51 lists the data beat sequencing to and from the DDR SDRAM and the data queues for each of the possible transfer sizes with each of the possible starting double-word offsets. All underlined double-word offsets are valid for the transaction.

Table 9-51. Memory Controller—Data Beat Ordering

Transfer Size	Starting Double-Word Offset	Double-Word Sequence[1] to/from DRAM and Queues
1 double word	0 1 2 3	**0** - 1 - 2 - 3 **1** - 2 - 3 - 0 **2** - 3 - 0 - 1 **3** - 0 - 1 - 2
2 double words	0 1 2	**0 - 1** - 2 - 3 **1 - 2** - 3 - 0 **2 - 3** - 0 - 1
3 double words	0 1	**0 - 1 - 2** - 3 **1 - 2 - 3** - 0

[1] All underlined **Double**-word offsets are valid for the transaction.

9.5.9 Page Mode and Logical Bank Retention

The DDR memory controller supports an open/closed page mode with an allowable open page for each logical bank of DRAM used. In closed page mode for DDR SDRAMs, the DDR memory controller uses the SDRAM auto-precharge feature, which allows the controller to indicate that the page must be automatically closed by the DDR SDRAM after the READ or WRITE access. This is performed using MA[10] of the address during the COMMAND phase of the access to enable auto-precharge. Auto-precharge is non-persistent in that it is either enabled or disabled for each individual READ or WRITE command. It can, however, be enabled or disabled separately for each chip select.

When the DDR memory controller operates in open page mode, it retains the currently active SDRAM page by not issuing a precharge command. The page remains open until one of the following conditions occurs:

- Refresh interval is met.
- The user-programmable DDR_SDRAM_INTERVAL[BSTOPRE] value is exceeded.

- There is a logical bank row collision with another transaction that must be issued.

Page mode can dramatically reduce access latencies for page hits. Depending on the memory system design and timing parameters, using page mode can save two to three clock cycles for subsequent burst accesses that hit in an active page. Also, better performance can be obtained using more banks, especially in systems which use many different channels. Page mode is disabled by clearing DDR_SDRAM_INTERVAL[BSTOPRE] or setting CS*n*_CONFIG[AP_*n*EN].

9.5.10 Error Checking and Correcting (ECC)

The DDR memory controller supports error checking and correcting (ECC) for the data path between the core master and system memory. The memory detects all double-bit errors, detects all multi-bit errors within a nibble, and corrects all single-bit errors. Other errors may be detected, but are not guaranteed to be corrected or detected. Multi-bit errors are always reported when error reporting is enabled. When a single-bit error occurs, the single-bit error counter register is incremented, and its value compared to the single-bit error trigger register. An error is reported when these values are equal. The single-bit error registers can be programmed such that minor memory faults are corrected and ignored, but a catastrophic memory failure generates an interrupt.

For writes that are smaller than 64 bits, the DDR memory controller performs a double-word read from system memory of the address for the write (checking for errors), and merges the write data with the data read from memory. Then, a new ECC code is generated for the merged double word. The data and ECC code is then written to memory. If a multi-bit error is detected on the read, the transaction completes the read-modify-write to keep the DDR memory controller from hanging. However, the corrupt data is masked on the write, so the original contents in SDRAM remain unchanged.

The syndrome encodings for the ECC code are shown in Table 9-52 and Table 9-53.

In 32-bit mode, Table 9-52 is split into 2 halves. The first half, consisting of rows 0–31, is used to calculate the ECC bits for the first 32 data bits of any 64-bit granule of data. This always applies to the odd data beats on the DDR data bus. The second half of the table, consisting of rows 32–63, is used to calculate the ECC bits for the second 32 bits of any 64-bit granule of data. This always applies to the even data beats on the DDR data bus.

Table 9-52. DDR SDRAM ECC Syndrome Encoding

Data Bit	Syndrome Bit							
	0	1	2	3	4	5	6	7
0	•	•						•
1	•		•					•
2	•			•				•
3	•				•			•
4	•	•				•		
5	•		•			•		
6	•			•		•		
7	•				•	•		

Data Bit	Syndrome Bit							
	0	1	2	3	4	5	6	7
32			•	•				•
33			•		•			•
34	•		•		•			
35		•	•		•			
36			•	•		•		
37			•		•	•		
38	•		•		•	•		•
39		•	•		•	•		•

Table 9-52. DDR SDRAM ECC Syndrome Encoding (continued)

Data Bit	Syndrome Bit							
	0	1	2	3	4	5	6	7
8	•	•					•	
9	•		•				•	
10	•			•			•	
11	•				•		•	
12	•	•				•	•	•
13	•		•			•	•	•
14	•			•		•	•	•
15	•				•	•	•	•
16		•	•					•
17		•		•				•
18		•			•			•
19	•	•			•			
20		•	•			•		
21		•		•		•		
22		•			•	•		
23	•	•			•	•		•
24		•	•				•	
25		•		•			•	
26		•			•		•	
27	•	•			•		•	•
28		•	•			•	•	•
29		•		•		•	•	•
30		•			•	•	•	•
31	•	•			•	•	•	

Data Bit	Syndrome Bit							
	0	1	2	3	4	5	6	7
40			•	•			•	
41			•		•		•	
42	•		•		•		•	•
43		•	•		•		•	•
44			•	•		•	•	•
45			•		•	•	•	•
46	•		•		•	•	•	
47		•	•		•	•	•	
48		•				•	•	
49			•			•	•	
50				•		•	•	
51	•					•	•	
52		•				•		•
53			•			•		•
54				•		•		•
55	•					•		•
56		•					•	•
57			•				•	•
58				•			•	•
59	•						•	•
60				•	•		•	
61	•			•	•		•	•
62		•		•	•		•	•
63			•	•	•		•	•

Table 9-53. DDR SDRAM ECC Syndrome Encoding (Check Bits)

Check Bit	Syndrome Bit							
	0	1	2	3	4	5	6	7
0	•							
1		•						
2			•					
3				•				
4					•			
5						•		

Table 9-53. DDR SDRAM ECC Syndrome Encoding (Check Bits) (continued)

Check Bit	Syndrome Bit							
	0	1	2	3	4	5	6	7
6							•	
7								•

9.5.11 Error Management

The DDR memory controller detects four different kinds of errors: training, single-bit, multi-bit, and memory select errors. The following discussion assumes all the relevant error detection, correction, and reporting functions are enabled as described in Section 9.4.1.26, "Memory Error Interrupt Enable (ERR_INT_EN)," Section 9.4.1.25, "Memory Error Disable (ERR_DISABLE)," and Section 9.4.1.24, "Memory Error Detect (ERR_DETECT)."

Single-bit errors are counted and reported based on the ERR_SBE value. When a single-bit error is detected, the DDR memory controller does the following:

- Corrects the data
- Increments the single-bit error counter ERR_SBE[SBEC]
- Generates an interrupt if the counter value ERR_SBE[SBEC] equals the programmable threshold ERR_SBE[SBET]
- Completes the transaction normally

If a multi-bit error is detected for a read, the DDR memory controller logs the error and generates the interrupt, and transfer error acknowledge (TEA) is asserted internally on the CSB bus (if enabled, as described in Section 9.4.1.25, "Memory Error Disable (ERR_DISABLE)"). Another error the DDR memory controller detects is a memory select error, which causes the DDR memory controller to log the error and generate a interrupt (if enabled, as described in Section 9.4.1.24, "Memory Error Detect (ERR_DETECT)"). This error is detected if the address from the memory request does not fall into any of the enabled, programmed chip select address ranges. For all memory select errors, the DDR memory controller does not issue any transactions onto the pins after the first read has returned data strobes. If the DDR memory controller is not using sample points, then a dummy transaction is issued to DDR SDRAM with the first enabled chip select. In this case, the source port on the pins is forced to 0x1F to show the transaction is not real. Table 9-54 shows the errors with their descriptions. The final error the memory controller detects is the automatic calibration error. This error is set if the memory controller detects an error during its training sequence.

Table 9-54. Memory Controller Errors

Category	Error	Descriptions	Action	Detect Register
Notification	Single-bit ECC threshold	The number of ECC errors has reached the threshold specified in the ERR_SBE.	The error is reported through interrupt if enabled.	The error control register only logs read versus write, not full type
Access Error	Multi-bit ECC error	A multi-bit ECC error is detected during a read, or read-modify-write memory operation.		
	Memory select error	Read, or write, address does not fall within the address range of any of the memory banks.		

9.6 Initialization/Application Information

System software must configure the DDR memory controller, using a memory polling algorithm at system start-up, to correctly map the size of each bank in memory. Then, the DDR memory controller uses its bank map to assert the appropriate $\overline{\mathrm{MCS}n}$ signal for memory accesses according to the provided bank depths. System software must also configure the DDR memory controller at system start-up to appropriately multiplex the row and column address bits for each bank. Refer to row-address configuration in Section 9.4.1.2, “Chip Select Configuration (CSn_CONFIG).” Address multiplexing occurs according to these configuration bits.

At system reset, initialization software (boot code) must set up the programmable parameters in the memory interface configuration registers. See Section 9.4.1, “Register Descriptions,” for more detailed descriptions of the configuration registers. These parameters are shown in Table 9-55.

Table 9-55. Memory Interface Configuration Register Initialization Parameters

Name	Description	Parameter		Section/page
CS*n*_BNDS	Chip select memory bounds	SA*n* EA*n*		9.4.1.1/9-18
CS*n*_CONFIG	Chip select configuration	CS_*n*_EN AP_*n*_EN ODT_RD_CFG ODT_WR_CFG	BA_BITS_CS_*n* ROW_BITS_CS_*n* COL_BITS_CS_*n*	9.4.1.2/9-18
TIMING_CFG_3	Extended timing parameters for fields in TIMING_CFG_1	EXT_REFREC		9.4.1.3/9-20
TIMING_CFG_0	Timing configuration	RWT WRT RRT WWT	ACT_PD_EXIT PRE_PD_EXIT ODT_PD_EXIT MRS_CYC	9.4.1.4/9-21
TIMING_CFG_1	Timing configuration	PRETOACT ACTTOPRE ACTTORW CASLAT	REFREC WRREC ACTTOACT WRTORD	9.4.1.5/9-22
TIMING_CFG_2	Timing configuration	ADD_LAT CPO WR_LAT RD_TO_PRE	WR_DATA_DELAY CKE_PLS FOUR_ACT	9.4.1.6/9-24

Table 9-55. Memory Interface Configuration Register Initialization Parameters (continued)

Name	Description	Parameter	Section/page
DDR_SDRAM_CFG	Control configuration	SREN ECC_EN RD_EN SDRAM_TYPE DYN_PWR 32_BE 8_BE DBW NCAP 2T_EN BA_INTLV_CTL x32_EN HSE BI	9.4.1.7/9-26
DDR_SDRAM_CFG_2	Control configuration	SR_IE DLL_RST_DIS DQS_CFG ODT_CFG NUM_PR D_INIT	9.4.1.8/9-29
DDR_SDRAM_MODE	Mode configuration	ESDMODE SDMODE	9.4.1.9/9-31
DDR_SDRAM_MODE_2	Mode configuration	ESDMODE2 ESDMODE3	9.4.1.10/9-32
DDR_SDRAM_INTERVAL	Interval configuration	REFINT BSTOPRE	9.4.1.12/9-35
DDR_DATA_INIT	Data initialization configuration register	INIT_VALUE	9.4.1.13/9-35
DDR_SDRAM_CLK_CNTL	Clock adjust	CLK_ADJUST	9.4.1.14/9-36
DDR_INIT_ADDR	Initialization address	INIT_ADDR	9.4.1.15/9-36

9.6.1 Programming Differences between Memory Types

Depending on the memory type used, certain fields must be programmed differently. Table 8-123 illustrates the differences in certain fields for different memory types. Note that this table does not list all fields that must be programmed.

Table 9-56. Programming Differences between Memory Types

Parameter	Description	Differences		Section/page
AP*n*_EN	Chip Select *n* Auto Precharge Enable	DDR1DDR2	Can be used to place chip select *n* in auto precharge mode	9.4.1.2/9-18
ODT_RD_CFG	Chip Select ODT Read Configuration	DDR1	Should always be set to 000	9.4.1.2/9-18
		DDR2	Can be enabled to assert ODT if desired. This could be set differently depending on system topology. However, systems with only 1 chip select typically not uses ODT when issuing reads to the memory.	

Table 9-56. Programming Differences between Memory Types (continued)

Parameter	Description	Differences		Section/page
ODT_WR_CFG	Chip Select ODT Write Configuration	DDR1	Should always be set to 000	9.4.1.2/9-18
		DDR2	Can be enabled to assert ODT if desired. This could be set differently depending on system topology. However, ODT typically is set to assert for the chip select that is getting written to (value would be set to 001).	
ODT_PD_EXIT	ODT Powerdown Exit	DDR1	Should be set to 0001	9.4.1.4/9-21
		DDR2	Should be set according to the DDR2 specifications for the memory used. The JEDEC parameter this applies to is t_{AXPD}.	
PRETOACT	Precharge to Activate Timing	DDR1	Should be set according to the specifications for the memory used (t_{RP})	9.4.1.5/9-22
		DDR2	Should be set according to the specifications for the memory used (t_{RP})	
ACTTOPRE	Activate to Precharge Timing	DDR1	Should be set, along with the Extended Activate to Precharge Timing, according to the specifications for the memory used (t_{RAS})	9.4.1.5/9-22
		DDR2	Should be set, along with the Extended Activate to Precharge Timing, according to the specifications for the memory used (t_{RAS})	
ACTTORW	Activate to Read/Write Timing	DDR1	Should be set according to the specifications for the memory used (t_{RCD})	9.4.1.5/9-22
		DDR2	Should be set according to the specifications for the memory used (t_{RCD})	
CASLAT	CAS Latency	DDR1	Should be set, along with the Extended CAS Latency, to the desired CAS latency	9.4.1.5/9-22
		DDR2	Should be set, along with the Extended CAS Latency, to the desired CAS latency	
REFREC	Refresh Recovery	DDR1	Should be set, along with the Extended Refresh Recovery, to the specifications for the memory used (t_{RFC})	9.4.1.5/9-22
		DDR2	Should be set, along with the Extended Refresh Recovery, to the specifications for the memory used (T_{RFC})	
WRREC	Write Recovery	DDR1	Should be set according to the specifications for the memory used (t_{WR})	9.4.1.5/9-22
		DDR2	Should be set according to the specifications for the memory used (t_{WR})	
ACTTOACT	Activate *A* to Activate *B*	DDR1	Should be set according to the specifications for the memory used (t_{RRD})	9.4.1.5/9-22
		DDR2	Should be set according to the specifications for the memory used (t_{RRD})	

Table 9-56. Programming Differences between Memory Types (continued)

Parameter	Description	Differences		Section/page
WRTORD	Write to Read Timing	DDR1	Should be set according to the specifications for the memory used (t_{WTR})	9.4.1.5/9-22
		DDR2	Should be set according to the specifications for the memory used (t_{WTR})	
ADD_LAT	Additive Latency	DDR1	Should be set to 000	9.4.1.6/9-24
		DDR2	Should be set to the desired additive latency. This must be set to a value less than TIMING_CFG_1[ACTTORW]	
WR_LAT	Write Latency	DDR1	Should be set to 001	9.4.1.6/9-24
		DDR2	Should be set to CAS latency – 1 cycle. For example, if the CAS latency if 5 cycles, then this field should be set to 100 (4 cycles).	
RD_TO_PRE	Read to Precharge Timing	DDR1	Should be set to 010 if burst length is 4 and 100 if burst length is 8	9.4.1.6/9-24
		DDR2	Should be set according to the specifications for the memory used (t_{RTP}). Time between read and precharge for non-zero value of additive latency (AL) is a minimum of AL + t_{RTP} cycles.	
CKE_PLS	Minimum CKE Pulse Width	DDR1	Can be set to 001	9.4.1.6/9-24
		DDR2	Should be set according to the specifications for the memory used (t_{CKE})	
FOUR_ACT	Four Activate Window	DDR1	Should be set to 00001	9.4.1.6/9-24
		DDR2	Should be set according to the specifications for the memory used (t_{FAW}). Only applies to eight logical banks.	
RD_EN	Registered DIMM Enable	DDR1	If registered DIMMs are used, then this field should be set to 1	9.4.1.7/9-26
		DDR2	If registered DIMMs are used, then this field should be set to 1	
8_BE	8-beat burst enable	DDR1	If a 32-bit bus is used, and 8-beat bursts are desired, then this field should be set to 1.	9.4.1.7/9-26
		DDR2	Should be set to 0	
2T_EN	2T Timing Enable	DDR1	In heavily loaded systems, this can be set to 1 to gain extra timing margin on the interface at the cost of address/command bandwidth.	9.4.1.7/9-26
		DDR2	In heavily loaded systems, this can be set to 1 to gain extra timing margin on the interface at the cost of address/command bandwidth.	
DLL_RST_DIS	DLL Reset Disable	DDR1	Should typically be set to 0, unless it is desired to bypass the DLL reset when exiting self refresh.	9.4.1.8/9-29
		DDR2	Should typically be set to 0, unless it is desired to bypass the DLL reset when exiting self refresh.	

Table 9-56. Programming Differences between Memory Types (continued)

Parameter	Description	Differences		Section/page
DQS_CFG	DQS Configuration	DDR1	Should be set to 00	9.4.1.8/9-29
		DDR2	Should be set to 00	
ODT_CFG	ODT Configuration	DDR1	Should be set to 00	9.4.1.8/9-29
		DDR2	Can be set for termination at the IOs according to system topology. Typically, if ODT is enabled, then the internal IOs should be set up for termination only during reads to DRAM.	
BSTOPR	Burst To Precharge Interval	DDR1	Can be set to any value, depending on the application. Auto precharge can be enabled by setting this field to all 0s.	9.4.1.12/9-35
		DDR2	Can be set to any value, depending on the application. Auto precharge can be enabled by setting this field to all 0s.	

9.6.2 DDR SDRAM Initialization Sequence

After configuration of all parameters is complete, system software must set DDR_SDRAM_CFG[MEM_EN] to enable the memory interface. Note that 200 μs must elapse after DRAM clocks are stable (DDR_SDRAM_CLK_CNTL[CLK_ADJUST] is set and any chip select is enabled) before MEM_EN can be set, so a delay loop in the initialization code may be necessary if software is enabling the memory controller. If DDR_SDRAM_CFG[BI] is not set, the DDR memory controller conducts an automatic initialization sequence to the memory, which follows the memory specifications. If the bypass initialization mode is used, then software can initialize the memory through the DDR_SDRAM_MD_CNTL register.

9.6.3 Using Forced Self-Refresh Mode to Implement a Battery-Backed RAM System

This section describes the options offered by this device to support battery-backed main memory.

9.6.3.1 Hardware Based Self-Refresh

An external voltage sense device can be connected to this device through the IRQ1/DDR_SR_REQ signal. The assertion of this signal, when a voltage drop has been identified, is then be sensed by the DDR controller. The DDR controller immediately responds by sending a self refresh command to main memory, telling it to enter self-refresh mode.The DDR controller to the assertion of DDR_SR_REQ is controlled by DDR_SDRAM_CFG_2[SR_IE]; see Section 9.4.1.8, "DDR SDRAM Control Configuration 2 (DDR_SDRAM_CFG_2)," for further information on this bit. Note that if IRQ1/DDR_SR_REQ is defined to be used as DDR_SR_REQ, it precludes any other usage of this pin as IRQ1; thus IRQ1 source should be disabled in the interrupt controller. Section 8.5.13, "System External Interrupt Mask Register (SEMSR)," contains a description of the register used to accomplish this task.

9.6.3.2 Software Based Self-Refresh

The DDR controller also has a software-programmable bit, DDR_SDRAM_CFG_2[FRC_SR], that immediately puts main memory into self-refresh mode. See Section 9.4.1.8, “DDR SDRAM Control Configuration 2 (DDR_SDRAM_CFG_2),” for a description of this register.

It is expected that a critical interrupt routine triggered by an external voltage sensing device has time to set this bit.

9.6.3.3 Bypassing Re-initialization During Battery-Backed Operation

The DDR controller offers an initialization bypass feature (DDR_SDRAM_CFG[BI]), which system designers may use to prevent re-initialization of main memory during system power-on following an abnormal shutdown. See Section 9.4.1.7, “DDR SDRAM Control Configuration (DDR_SDRAM_CFG),” for information on this bit and Section 9.4.1.15, “DDR Initialization Address (DDR_INIT_ADDR),” for a discussion of avoiding possible ECC errors in this mode.

Note that the DDR controller automatically waits 200 DRAM cycles before issuing any command after the assertion of MCKE[0:1] when this mode is used.

Section, Page No. **Changes**

10.1, 10-1 Replace Figure 10-1, “Enhanced Local Bus Controller Block Diagram,” with the following:

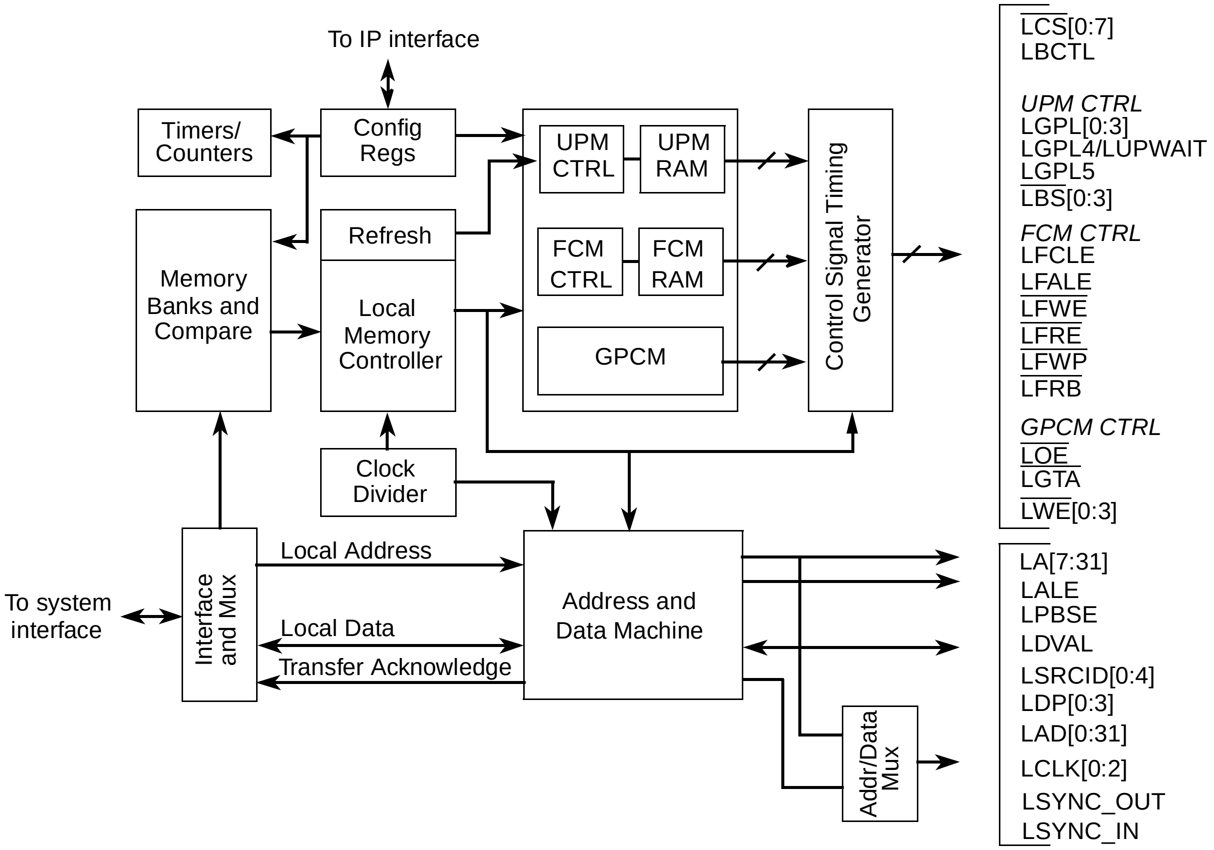


Figure 10-1 Enhanced Local Bus Controller Block Diagram

10.1.2, 10-2 In Section 10.1.2, “Features,” modify the bullet “Odd/even parity checking including read-modify-write (RMW) parity for single accesses” to say “Odd/even parity checking for single access.”

10.1.2, 10-3 Add the following list item:

- Different machines (FCM/GPCM/UPM) share the address, data, and control signals. While the eLBC is servicing a transaction, subsequent transactions are queued until the current transaction has completed.

10.3.1.1, 10-11 In Figure 10-2, “Base Registers (BRn),” change reset value of reserved bit 30 from “1” to “0.”

10.3.1.9, 10-26 In Section 10.3.1.9, “Transfer Error Status Register (LTESR),” add the following sentence to the end of the paragraph:

“Note that error statuses are only reflected in LTESR if they have been enabled in LTEDR.”

Section, Page No.	Changes
10.3.1.10, 10-27	In Section 10.3.1.10, “Transfer Error Check Disable Register (LTEDR),” replace the first sentence with the following: “The transfer error check disable register (LTEDR), shown in Figure 13-26, is used to disable error/event checking, which are reported in LTESR.”
10.3.1.12, 10-30	Add the following Note prior to Figure 10-16, “Transfer Error Attributes Register (LTEATR)”:

NOTE

LTEATR may not capture accurate information for errors that occur when an FCM special operation is in progress.

Section, Page No.	Changes
10.3.1.12, 10-30	Update the bit field description of SRCID (bits 11–15) in Table 10-19, “LTEATR Field Descriptions” as follows: Captures the source of the transaction when this information is provided on the internal interface to the eLBC. The coding of the source ID debug information is the same as the coding of AEATR[MSTR_ID] (see Section 6.2.6, “Arbiter Event Attributes Register (AEATR).”)
10.3.1.13, 10-31	Add the following Note prior to Figure 10-17, “Transfer Error Address Register (LTEAR)”:

NOTE

LTEAR may not capture accurate information for errors that occur when an FCM special operation is in progress.

Section, Page No.	Changes
10.3.1.14, 10-32	In Table 10-21, “LBCR Field Descriptions,” replace the description for bits 16–23 with the following:

Table 10-21. LBCR Field Descriptions

Bits	Name	Description
16–23	BMT	Bus monitor timing. Defines the bus monitor time-out period. The number of LCLK clock cycles to count down before a time-out error is generated is given by: if BMT = 0, then bus cycles = 256 × PS if BMT ≠ 0, then bus cycles = BMT × PS where PS is set according to LBCR[BMTPS]. The value of bus cycles must not be less than 40 bus cycles for reliable operation. When BMT = 0, bus cycles = 256 × PS.

Section, Page No. **Changes**

10.3.1.21, 10-40 In Figure 10-26, “Flash Byte Count Register,” add row for reset values. The modified figure is shown below:

Offset 0x0F4 Access: Read/Write

	0 … 19	20 … 31
R	—	BC
W		
Reset	All zeros	

Figure 10-26 Flash Byte Count Register

10.4, 10-41 Add the following Note at the end of the section:

NOTE

Different machines (FCM/GPCM/UPM) share the address, data, and control signals. While the eLBC is servicing a transaction, subsequent transactions are queued until the current transaction has completed.

10.4.1.2, 10-42 Add the following two paragraphs at the end of the section (after Figure 10-28, “Example of 8-Bit GPCM Writing 32 Bytes to Address 0x5420 (LCRR[PBYP] = 0)”):

“If the RCW is loaded by the eLBC, LALE may remain at an unknown value for up to 8 cycles after PORESET negation. Thus, LALE should be ignored for 8 CLKIN/PCI_SYNC_IN cycles after PORESET negation. In general, it is recommended that a latch be implemented for this adjustment and not a state machine triggered by LALE.

If the RCW is not loaded by the eLBC (for example, I2C or hard-coded options are used), then LALE is at an unknown value until the PLL is locked and should be ignored until the negation of HRESET.”

10.4.1.6, 10-45 Add the following Note at the end of the section:

NOTE

When the FCM is in the middle of a long transaction (such as NAND erase, write, and so on), another transaction on the GPCM or UPM will trigger the bus monitor to start, even though the GPCM or UPM is waiting for the FCM to finish. If the bus monitor times out, it could corrupt the current NAND flash operation as well as terminate the GPCM or UPM operation. To avoid such cases, it is recommended that the bus monitor timeout be programmed to its maximum setting of LBCR[BMT] = 0 and LBCR[BMTPS] = 0xF.

10.4.2.4, 10-57 Add an introductory sentence to state that Figure 10-43, “External Termination of GPCM Access,” is for PLL-enabled mode as follows:

In PLL-enabled mode, the timing of $\overline{\text{LGTA}}$ is illustrated by the example in Figure 10-43.

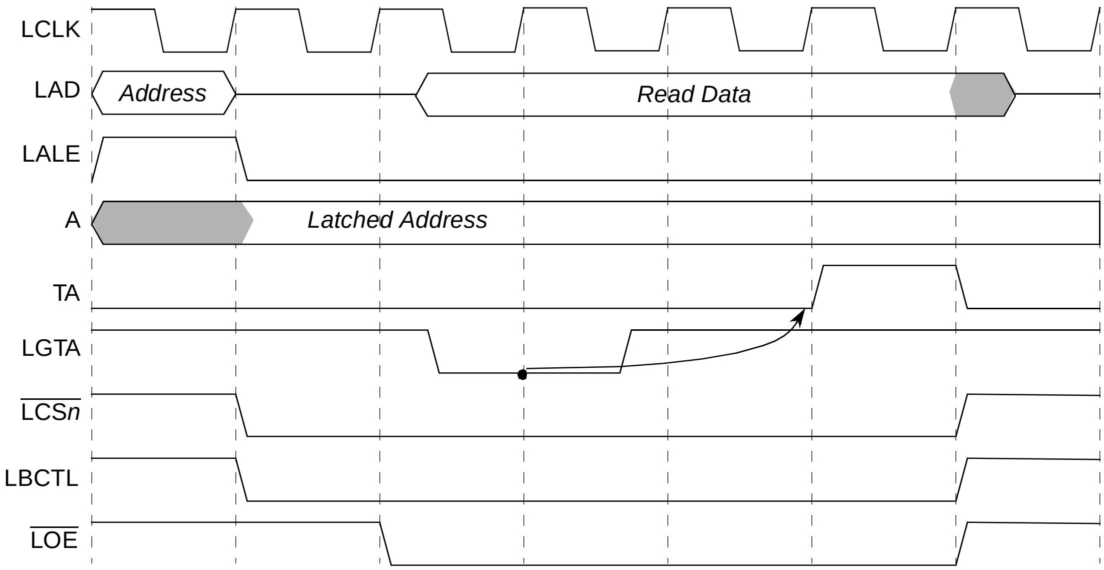


Figure 10-43. External Termination of GPCM Access (PLL Enabled Mode)

In addition, add a new figure for PLL bypass mode after Figure 10-43 (preceded by an introductory sentence) as follows:

In PLL bypass mode, the timing of $\overline{\text{LGTA}}$ is illustrated by the example in Figure 10-44.

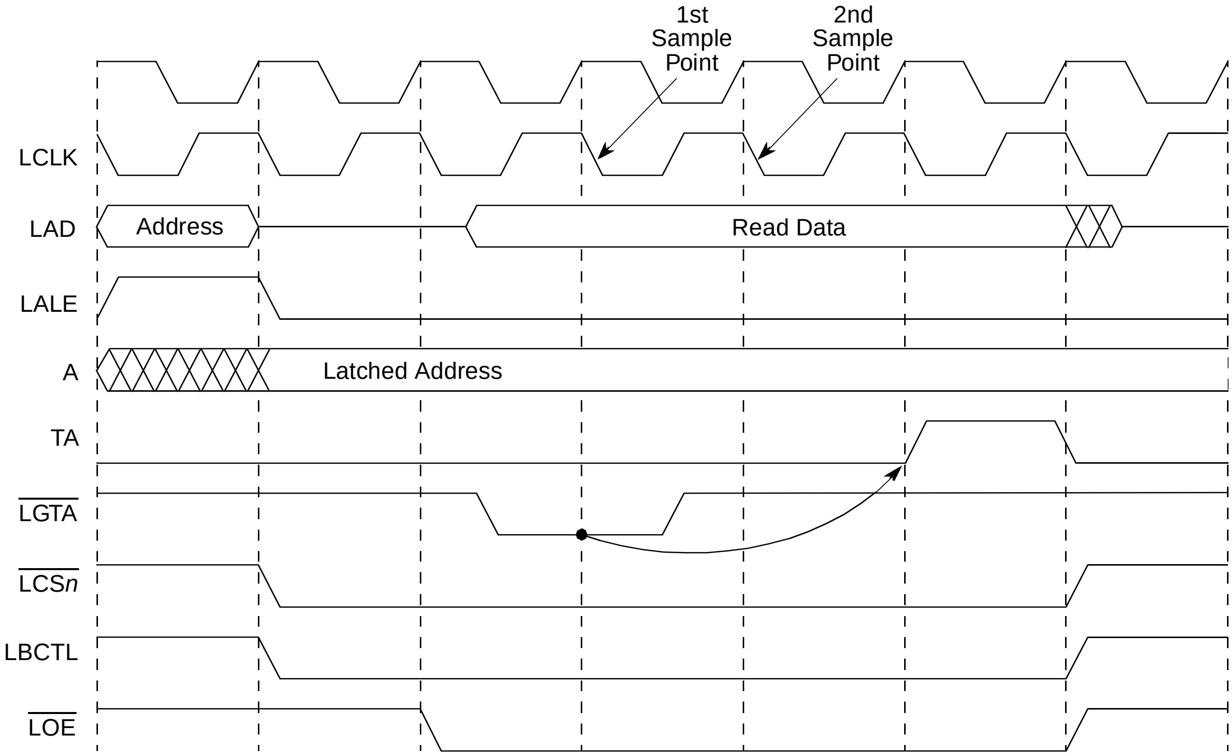


Figure 10-44. External Termination of GPCM Access (PLL Bypass Mode)

Section, Page No. **Changes**

10.4.4.4.7, 10-87 Add the following note prior to Table 10-40, “UPM Address Multiplexing”:

NOTE

Multiple-bank DRAM and SDRAM devices require that the bank address be driven during both RAS and CAS cycles. The UPM does not support a persistent bank address on both RAS and CAS cycles. Therefore, external logic must be used to supply a bank address to these devices.

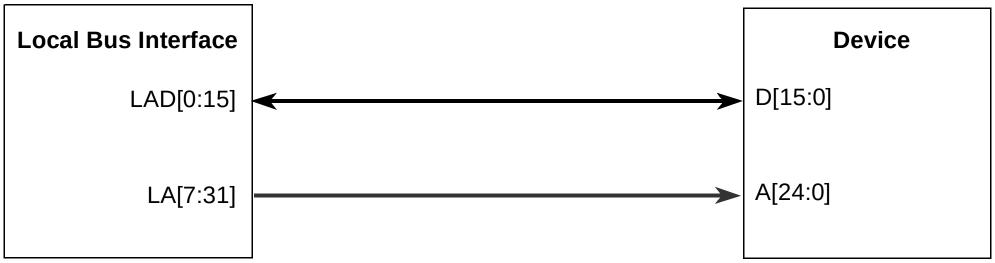
10.4.4.4.10, 10-89 Modify the first sentence of the second paragraph from:

“Synchronization of LUPWAIT starts at the rising edge of the bus clock and takes at least 1 bus cycle to complete.”

to:

“Synchronization of LUPWAIT starts at the falling edge of the bus clock and takes at least 1 bus cycle to complete.”

10.5.1.2, 10-92 Replace Figure 10-71, “Non-Multiplexed Address and Data Buses,” with the following:



Notes:
1. The LALE pin is used as LA[10].
2. To select this mode at POR, see CFG_LBMUX signal description in Table 4-1.

Figure 10-71. Non-Multiplexed Address and Data Buses

10.5.1.3, 10-92 Remove the section entitled, “Multiplexed Address and Data To Save Maximum Pins In 8- to 16-Bit Addressing.

Section, Page No.	**Changes**
Chapter 11, throughout	Remove references to “read wait”, “suspend”, “resume”, “read-wait control”, “interrupt detect”, and “SD combo cards” throughout the chapter.
11.1, 11-2	Modify Figure 11-2, “eSDHC Block Diagram,” as follows:

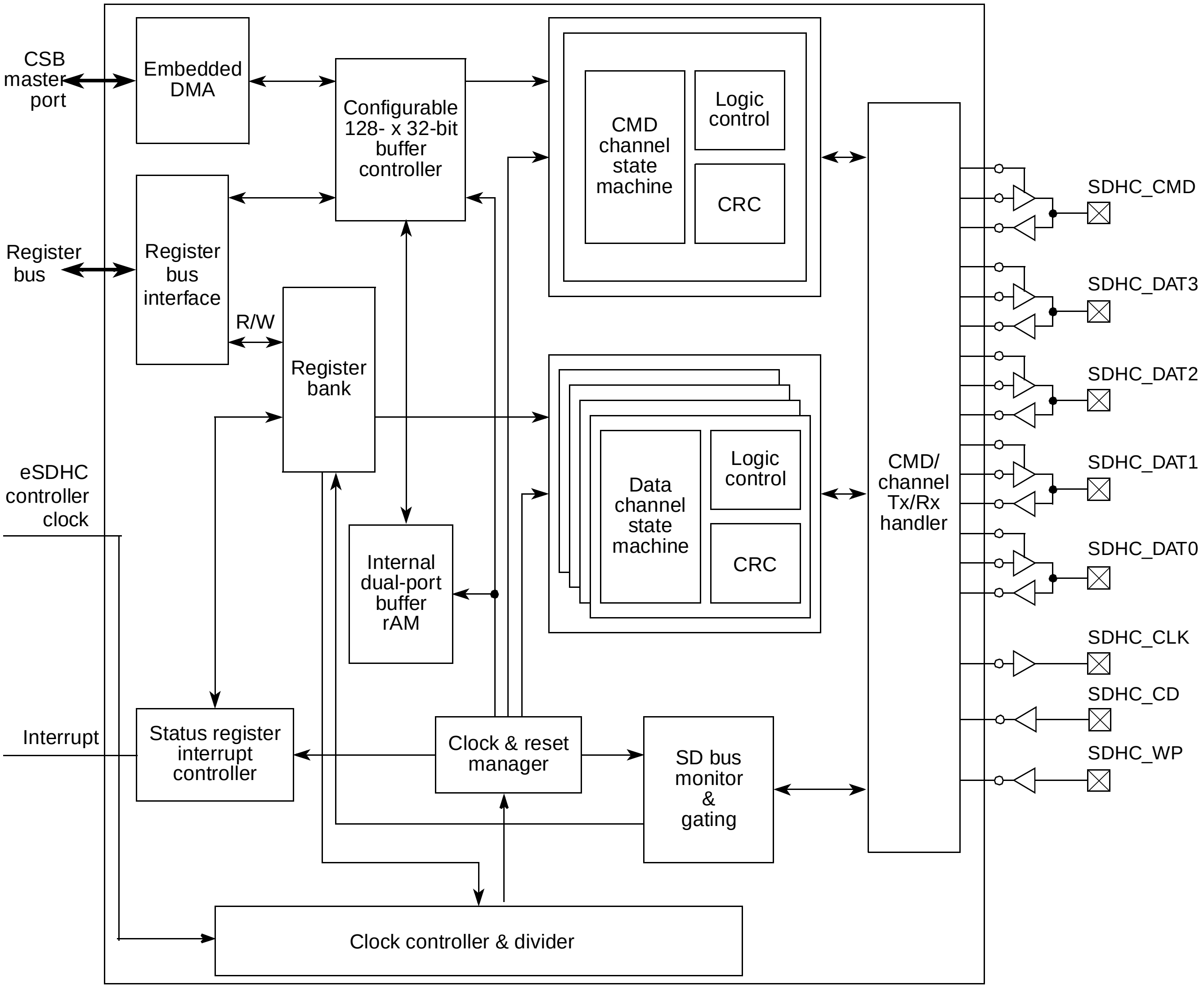


Figure 11-2 eSDHC Block Diagram

11.2, 11-3	Change the feature list item from: •Synchronous and asynchronous abort to: •Synchronous abort

Section, Page No.	**Changes**
11.3, 11-3	In Table 11-1, "Signal Properties," update the "Pull-up/Pull down required" column for the SDHC_DAT3 signal and update the "function" column for SDHC_DAT0 signal as follows:

Table 11-1. Signal Properties

Name	Port	Function	Reset State	Pull up/Pull down Required
SDHC_DAT3	I/O	**4-bit mode:** DAT3 line or configured as card detection pin **1-bit mode:** May be configured as card detection pin	High impedance	10 kΩ–100 kΩ to OV_{DD} is required during the normal operating conditions. Board should have a 100 KΩ pull-down resistor if DAT3 is used for an SD card detection. In this case, the pull-up resistor has to be disconnected during the identification phase. The pull-up resistor has to be connected and the pull-down resistor should be switched out after the identification phase.
SDHC_DAT0	I/O	**4-bit mode:** Dat 0 line	High impedance	Pull up

11.4, 11-4	In Table 11-2, "eSDHC Memory Map," update the reset value of Present State Register (PRSSTAT) as "0xFF8*n*_00n0."
11.4, 11-5	In Table 11-2, "eSDHC Memory Map," change the access information for the Interrupt Status (IRQSTAT) register from "w1c" to "Mixed."
11.4.1, 11-5	In Table 11-3, "DSADDR Field Description," divide the "Field" column to "Bits" column with value "0–31" and "Name" column with value "DS_ADDR" The updated table is shown below":

Table 11-3 DSADDR Field Description

Bits	Name	Description
0–31	DS_ADDR	DMA system address. When the eSDHC stops a DMA transfer, this register points to the system address of the next contiguous data position. **Note:** The DS_ADDR must be aligned to a four-byte boundary; the two least-significant bits must be cleared.

11.4.3, 11-7

In Table 11-5, "CMDARG Field Descriptions," replace the following sentence:

"If PRSSTAT[CMD] is set, this register is write-protected."

with:

"If PRSSTAT[CIHB] is set, this register is write-protected."

Section, Page No. **Changes**

11.4.6, 11-11

Modify the existing note prior to Figure 11-8, “Buffer Data Port Register (DATPORT),” from:

“When the internal DMA is not enabled and a write transaction is in operation, DATPORT must not be read.”

to:

“When the internal DMA is not enabled and a write transaction is in operation, DATPORT must not be read. DATPORT also must not be used to read (or write) data by the CPU or external DMA if the data will be written (or read) by the eSDHC internal DMA.”

11.4.7, 11-12

Update the reset values of Figure 11-9, “Present State Register (PRSSTAT),” to show as follows:

Offset: 0x024 (PRSSTAT) Access: Read

	0			3	4			7	8	9		11	12	13	14	15
R	—				DLSL				CLSL	—			WPSPL	CDPL	—	CINS
W																
Reset	*n*	*n*	*n*	*n*	*n*	*n*	*n*	*n*	1	0	0	0	*n*	*n*	0	*n*

	16			19	20	21	22	23	24	25	26	27	28	29	30	31
R	—				BREN	BWEN	RTA	WTA	SD OFF	PER OFF	HCK OFF	IPG OFF	—	DLA	CDIHB	CIHB
W																
Reset	0	0	0	0	0	0	0	0	*n*	*n*	*n*	*n*	0	0	0	0

Figure 11-9. Present State Register (PRSSTAT)

In addition, update the bit field descriptions of SDOFF (bit 24), PEROFF (bit 25), HCKOFF (bit 26), and IPGOFF (bit 27) in Table 11-11, “PRSSTAT Field Descriptions,” by adding the following note:

“This status bit resets to 0, but reflects the value of the automatic clock gating and may transition to 1 if the eSDHC is idle.”

Section, Page No. **Changes**

11.4.8, 11-16 In Figure 11-10, “Protocol Control Register (PROCTL),” make bit 7 reserved to show as follows:

Offset: 0x028 (PROCTL) Access: Read/Write

	0				4	5	6	7	8			11	12	13	14	15
R W	—					WE CRM	WE CINS	—						RW CTL	CREQ	SABG REQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16							23	24	25	26	27	28	29	30	31
R W	—								CDSS	CDTL	EMODE		D3CD	DTW		—
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Figure 11-10. Protocol Control Register (PROCTL)

In addition, remove bit description of WECINT (bit 7) from Table 11-12, “PROCTL Field Descriptions” and make bit 7 as reserved.

11.4.9, 11-21 In Table 11-13, “SYSCTL Field Descriptions,” change the last bulleted item of IPGEN (bit 31) to the following text:

“The internal bus clock is not gated off”

In addition, change “SD Physical Specification version 1.1” to “SD Physical Specification version 2.0” in the sentence “According to the SD Physical Specification version 1.1....” of the “16–23 SDCLKFS” field. The updated sentence now reads:

“According to the SD Physical Specification version 2.0, the maximum SD clock frequency is 50 MHz, and should never exceed this limit.”

11.4.10, 11-21 Modify the second sentence as follows:

“For all bits (except BGE), writing one to a bit clears it, while writing zero keeps the bit unchanged.”

In Figure 11-12, “Interrupt Status Register (IRQSTAT),” change the access information for the BGE bit (bit 29) from “w1c” to “R/W.” Also, change the register access from “w1c” to “Mixed.” The updated figure is shown below:

Offset: 0x030 (IRQSTAT) Access: Mixed

	0 2	3	4 6	7	8	9	10	11	12	13	14	15
R	—	DMAE	—	AC12E	—	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE
W		w1c		w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	All zeros											

	16 23	24	25	26	27	28	29	30	31
R	—	CRM	CINS	BRR	BWR	DINT	BGE	TC	CC
W		w1c	w1c	w1c	w1c	w1c		w1c	w1c
Reset	All zeros								

Figure 11-12 Interrupt Status Register (IRQSTAT)

11.4.10, 11-21, 11-23

In Figure 11-12, “Interrupt Status Register (IRQSTAT),” make bit 23 reserved to show as follows:

Offset: 0x030 (IRQSTAT) Access: w1c

	0 2	3	4 6	7	8	9	10	11	12	13	14	15
R	—	DMAE	—	AC12E	—	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE
W		w1c		w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	All zeros											

	16 22 23	24	25	26	27	28	29	30	31
R	—	CRM	CINS	BRR	BWR	DINT	BGE	TC	CC
W		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	All zeros								

Figure 11-12. Interrupt Status Register (IRQSTAT)

In addition, remove bit field description of CINT (bit 23) from Table 11-14, “IRQSTAT Field Descriptions,” and make it reserved.

Also, in Table 11-14, update bit field description of CIE (bit 12) from “1 Timeout” to “1 Error”.

11.4.11, 11-25, 11-26 In Figure 11-13, “Interrupt Status Enable Register (IRQSTATEN),” make bit 23 reserved to show as follows:

Offset: 0x034 (IRQSTATEN) Access: Read/Write

	0		2	3	4		6	7	8	9	10	11	12	13	14	15
R W	—			DMAE SEN	—			AC12E SEN	—	DEBE SEN	DCE SEN	DTOE SEN	CIE SEN	CEBE SEN	CCE SEN	CTOE SEN
Reset	0	0	0	1	0	0	0	1	0	1	1	1	1	1	1	1

	16						22	23	24	25	26	27	28	29	30	31
R W	—								CRM SEN	CINS SEN	BRR SEN	BWR SEN	DINT SEN	BGE SEN	TC SEN	CC SEN
Reset	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1

Figure 11-13. Interrupt Status Enable Register (IRQSTATEN)

In addition, remove bit field description of CINTSEN (bit 23) from Table 11-18, “IRQSTATEN Field Descriptions” and make it reserved.

11.4.12, 11-27,11-28 In Figure 11-14, “Interrupt Signal Enable Register (IRQSIGEN),” make bit 23 reserved to show as follows:

Offset: 0x038 (IRQSIGEN) Access: Read/Write

	0		2	3	4		6	7	8	9	10	11	12	13	14	15
R W	—			DMAE IEN	—			AC12E IEN	—	DEBE IEN	DCE IEN	DTOE IEN	CIE IEN	CEBE IEN	CCE IEN	CTOE IEN
Reset	All zeros															

	16							23	24	25	26	27	28	29	30	31
R W	—								CRM IEN	CINS IEN	BRR IEN	BWR IEN	DINT IEN	BGE IEN	TC IEN	CC IEN
Reset	All zeros															

Figure 11-14. Interrupt Signal Enable Register (IRQSIGEN)

In addition, remove bit field description of CINTIEN from Table 11-19, “IRQSIGEN Field Descriptions” and make it reserved.

Section, Page No.	**Changes**
11.4.14, 11-31	In Figure 11-16, “Host Capabilities Register (HOSTCAPBLT),” make bit 8 reserved to show as follows:

Offset: 0x040 (HOSTCAPBLT) Access: Read

	0				4	5	6	7	8	9	10	11	12	13	14	15
R	—					VS18	VS30	VS33	—	DMAS	HSS	—		MBL		
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1	1

	16															31
R	—															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 11-16. Host Capabilities Register (HOSTCAPBLT)

In addition, remove the bit field description of SRS (bit 8) from Table 11-22, “HOSTCAPBLT Field Descriptions,” and make it reserved.

11.4.16, 11-33	In Figure 11-18, “Force Event Register (FEVT),” make bit 0 reserved to show as follows:

Offset: 0x050 (FEVT) Access: Write

	0	1	2	3	4		6	7	8	9	10	11	12	13	14	15
R	—			0	—			0	—	0	0	0	0	0	0	0
W				FEVTDMAE				FEVTAC12E		FEVTDEBE	FEVTDCE	FEVTDTOE	FEVTCIE	FEVTCEBE	FEVTCCE	FEVTCTOE
Reset	All zeros															

	16							23	24	25	26	27	28	29	30	31
R	—								0	—		0	0	0	0	0
W									FEVTCNIBAC12E			FEVTAC12IE	FEVTAC12EBE	FEVTAC12CE	FEVTAC12TOE	FEVTAC12NE
Reset	All zeros															

Figure 11-18. Force Event Register (FEVT)

In addition, remove the bit field description of FEVTCINT (bit 0) from Table 11-24, “FEVT Field Descriptions,” and make it reserved.

Section, Page No. **Changes**

11.4.16, 11-34 In Table 11-24, “FEVT Field Descriptions,” change the value of field 15 from “FEVTCCE” to “FEVTCTOE.” The updated row is shown below:

Table 11-24. FEVT Field Descriptions

Bits	Name	Description
15	FEVTCTOE	Force event command time out error. Forces IRQSTAT[CTOE] to set.

11.5.2, 11-38 Add the following sentence at the end of the paragraph and prior to Figure 11-22, “DMA CSB Interface Block”:

“The internal DMA must not be used to read (or write) data if the data will be written (or read) by the CPU or an external DMA through the DATPORT register.”

11.6.2.1, 11-44 Replace Figure 11-25, “Flow Diagram for Card Detection,” and the three bullets after it with the following diagram and steps:

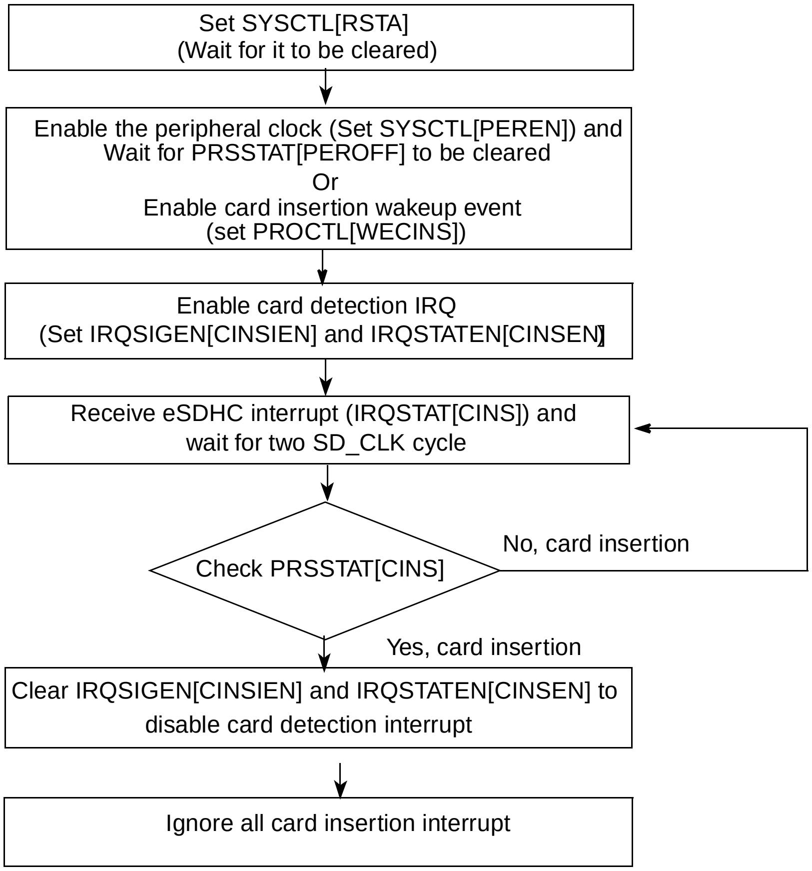


Figure 11-19 Flow Diagram for Card Detection

1. Set SYSCTL[RSTA] and wait for it to be cleared.
2. Set SYSCTL[PEREN] to enable the clocks and wait for PRSSTAT[PEROFF] to be cleared or enable wakeup event (set PROCTL[WECINS]).
3. Set IRQSIGEN[CINSIEN] and IRQSTATEN[CINSEN] to enable card detection interrupt.

4. When an IRQSTAT[CINS] interrupt from eSDHC is received, wait for two SD_CLK cycle and then check PRSSTAT[CINS] to see if the interrupt is caused by card insertion.
5. Clear the IRQSIGEN[CINSIEN] and IRQSTATEN[CINSEN] to disable card detection interrupt and ignore all card insertion interrupt afterwards.

11.6.2.2, 11-45 Add introductory text for the pseudocode and update the pseudocode as follows:

The following pseudocode shows how to initialize the eSDHC host controller and the memory card.

```
software_reset()
{
Configure the I/O muxes to select SD signals;
set_bit(SYSCTL, RSTA);
// software reset the host
set SYSCTL[DTOCV and SDCLKFS];
// get the SDHC_CLK of frequency around 400 kHz
poll PRSSTAT[CIHB and CDIHB];
// wait until both bits are cleared
set_bit(SYSCTRL, INTIA);
// send 80 clock ticks for card to power-up
If the card is SD/MMC
      send_command(CMD_GO_IDLE_STATE, <other parameters>);
// reset the card with CMD0
If the card is SDIO
      send_command(CMD_IO_RW_DIRECT, <other parameters>);
//reset the card with CMD52
```

11.6.5, 11-54 In Table 11-27, “Commands for MMC/SD,” change “version 1.1” to “version 2.0” in the “Description” column for CMD Index “CMD6.” The updated description should read:

“Checks switch ability (mode 0) and switch card function (mode 1). Refer to SD Physical Specification version 2.0 for details.”

11.6.5, 11-57 In Table 11-27, “Commands for MMC/SD,” add the following commands. Note that only newly added commands are shown.

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
ACMD18	adtc	[31:0] stuff bits	R1	SECURE_READ_MULTI_BLOCK	Protected Area Access Command: Reads continuously transfer data blocks from Protected Area of SD Memory Card. Refer to Security Specification Version 2.00 for more details.
ACMD25	adtc	[31:0] stuff bits	R1	SECURE_WRITE_MULTI_BLOCK	Protected Area Access Command: Writes continuously transfer data blocks to Protected Area of SD memory card. Refer to Security Specification Version 2.00 for more details.

ACMD26	adtc	[31:0] stuff bits	R1	SECURE_WRITE_MKB	System Area Access Command: Overwrite the existing Media Key Block (MKB) on System Area of SD Memory Card with new MKB. This command is used in dynamic update MKB scheme. Refer to Security Specification Version 2.00 for more details.
ACMD38	ac	[31:0] stuff bits	R1b	SECURE_ERASE	Protected Area Access Command: Erase a specified region of the Protected Area of SD Memory Card. Refer to Security Specification Version 2.00 for more details.
ACMD43	adtc	[31:24]Unit_Count: [23:16] MKB_ID: [15:0]Unit_Offset:	R1	GET_MKB	Reads Media Key Block from System Area of SD Memory Card. • Unit_Count specifies the Number of units to read. (Here, a unit = 512 bytes (fixed).) • MKB_ID specifies the application unique number. • Unit_Offset specifies the start address(offset) to read. Refer to Security Specification Version 2.00 for more details.
ACMD44	adtc	[31:0] stuff bits	R1	GET_MID	Reads Media ID from the System Area of SD Memory Card. Refer to Security Specification Version 2.00 for more details.
ACMD45	adtc	[31:0] stuff bits	R1	SET_CER_RN1	AKE Command: Writes random number RN1 as challenge1 in AKE process. Refer to Security Specification Version 2.00 for more details.
ACMD46	adtc	[31:0] stuff bits	R1	GET_CER_RN2	AKE Command: Reads random number RN2 as challenge2 in AKE process. Refer to Security Specification Version 2.00 for more details.
ACMD47	adtc	[31:0] stuff bits	R1	SET_CER_RES2	AKE Command: Writes RES2 as response2 to RN2 in AKE process. Refer to Security Specification Version 2.00 for more details.
ACMD48	adtc	[31:0] stuff bits	R1	GET_CER_RES1	AKE Command: Reads RES1 as response1 to RN1 in AKE process. Refer to Security Specification Version 2.00 for more details.
ACMD49	ac	[31:0] stuff bits	R1b	CHANGE_SECURE_AREA	Protected Area Access Command: Change size of Protected Area. Refer to Security Specification Version 2.00 for more details.

Section, Page No.	Changes
13.5.3, 13-18	Update the paragraph beginning “Accesses to CSB memory depend...” to read as follows: “Accesses to CSB memory depend on the alignment of the source and destination addresses and the size of the transfer. The DMA controller transfers a full cache line whenever possible. On misaligned addresses, full cache line transfers (32 byte bursting) occur if the source and destination offsets end in the same byte offset. For example, if the destination address is 0x4000_1050 and the source address is 0x9000_2050, the transfer bursts because both end in 0x50. However, if the destination address is 0x4000_1050 and the source is 0x9000_2000, the transfer does not burst because the last byte offset is not the same. On misaligned destination addresses, subtransfers of less than a cache line occur on the initial and final beats of the transfer while full cache lines occur on intermediate beats.”
14.4.7, 14-61	Add the following section after Section 14.4.7, “CompactPCI Hot Swap Specification Support”. Note that the subsequent figures and tables are re-numbered.

14.4.8 Byte Ordering

Whenever data must cross a bridge between two busses, the byte ordering of data on the source and destination buses must be considered. The internal platform bus of this device is inherently big endian and the PCI bus interface is inherently little endian.

There are two methods to handle ordering of data as it crosses a bridge—address invariance and data invariance. Address invariance preserves the addressing of bytes within a scalar data element, but not the relative significance of the bytes within that scalar. Conversely, data invariance preserves the relative significance of bytes within a scalar, but not the addressing of the individual bytes that make up a scalar.

This device uses address invariance as its default byte ordering policy, but it also may be configured to use a data invariance policy. The byte ordering policy is controlled by the ATMUs using the data invariance enable parameter in the inbound and outbound window attributes registers (PIWAR*n*[DIEN] and POWAR*n*[DIEN]). Note that by using the ATMUs to control the byte ordering policy, it is possible to allow data invariance for certain devices or regions, while mapping other devices or regions using address invariance. However, in such cases, it is important that software keep track of which devices and regions use which byte ordering policy and access them accordingly.

As stated above, address invariance preserves the byte address of each byte on an I/O interface as it is placed in memory or moved into a register. This policy can have the effect of reversing the significance order of bytes (most significant to least significant and vice versa), but it has the benefit of preserving the format of general data structures. Provided that software is aware of the endianness and format of the data structure, it can correctly interpret the data on either side of the bridge.

Figure 14-55 shows the transfer of a 4-byte scalar, 0x4142_4344, from a big endian source across an address invariant bridge to a little endian destination.

	Big endian source bus					Little endian destination bus			
Byte lane	0	1	2	3		3	2	1	0
Address lsbs	000	001	010	011		011	010	001	000
Data	41	42	43	44	⇨	44	43	42	41
Significance	MSB			LSB		MSB			LSB

Figure 14-55. Address Invariant Byte Ordering—4 bytes Outbound

Note that although the significance of the bytes within the scalar have changed, the address of the individual bytes that make up the scalar have not changed. As long as software is aware that the source of the data used a big endian format, the data can be interpreted correctly.

Figure 14-56 shows data flowing the other way, from a little endian source to a big endian destination.

	Little endian source bus					Big endian destination bus			
Byte lane	3	2	1	0		0	1	2	3
Address lsbs	011	010	001	000		000	001	010	011
Data	41	42	43	44	⇨	44	43	42	41
Significance	MSB			LSB		MSB			LSB

Figure 14-56. Address Invariant Byte Ordering—4 bytes Inbound

Figure 14-57 shows an outbound transfer of an 8-byte scalar, 0x5455_1617_CDCE_2728, using address invariance.

	Big endian source bus									Little endian destination bus							
Byte lane	0	1	2	3	4	5	6	7		7	6	5	4	3	2	1	0
Address lsbs	000	001	010	011	100	101	110	111		111	110	101	100	011	010	001	000
Data	54	55	16	17	CD	CE	27	28	⇨	28	27	CE	CD	17	16	55	54
Significance	MSB							LSB		MSB							LSB

Figure 14-57. Address Invariant Byte Ordering—8 bytes Outbound

Figure 14-58 shows an inbound transfer of a 2-byte scalar, 0x5837, using address invariance.

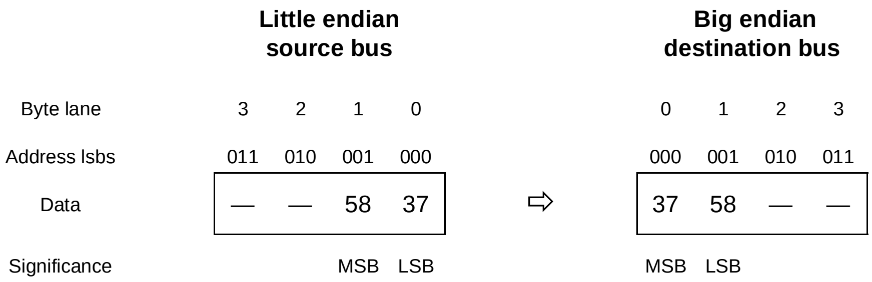


Figure 14-58. Address Invariant Byte Ordering—2 bytes Inbound

Note that in all of these examples, the original addresses of the individual bytes within the scalars (as created by the source) have been preserved.

14.4.8.1 Byte Order for Configuration Transactions

All internal memory-mapped registers in the CCSR space use big endian byte ordering. However, the PCI specification defines PCI configuration registers as little endian. All accesses to the PCI configuration port, CFG_DATA, including the those targeting the internal PCI configuration registers, use the address invariance policy as shown in Figure 14-59. Therefore, software must access CFG_DATA with little-endian formatted data—either using the **lwbrx**/**stwbrx** instructions or by manipulating the data before writing to and after reading from CFG_DATA.

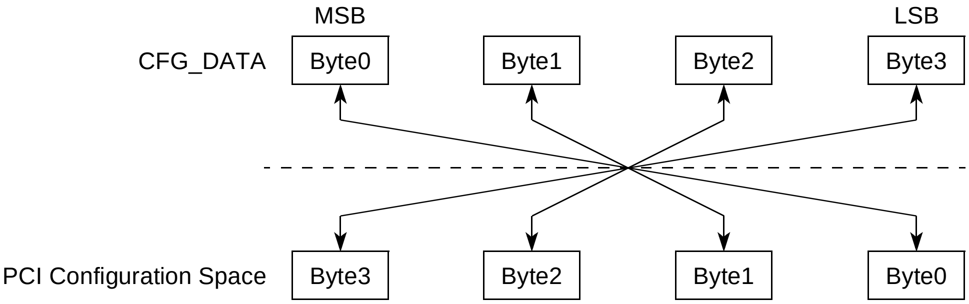


Figure 14-59. CFG_DATA Byte Ordering

15.3.1, 15-10 Update Table 15-3, “PCI Express Memory Map,” for the following registers. Note that only affected registers are shown:

Offset	Register	Access	Reset	Section/Page
0x4D4	PCI Express BAR Enable Register (PEX_BAR_ENABLE)	R/W	0x0000_000F	15.4.7.1/15-73
0x4E0	PCI Express BAR Select Configuration Register (PEX_BAR_SEL)	R/W	0x0000_0000	15.4.7.2/15-74
0x504	PCI Express BAR Prefetch Configuration Register (PEX_BAR_PF)	R/W	0x0000_0000	15.4.7.3/15-74
0x590	PCI Express PME_To_Ack Timeout Register (PEX_PME_TO_ACK_TOR)	R/W	0x0262_5A00	15.4.8.1/15-75
0x5A0	PCI Express PCI Interrupt Mask Register (PEX_SS_INTR_MASK)	R/W	0x0000_003F	15.4.8.3/14-77

15.3.1, 15-10 Update the “Section/Page” column in Table 15-3, “PCI Express Memory Map,” for PEX_SS_INTR_MASK register as follows:

Table 15-3. PCI Express Memory Map

Offset	Register	Access	Reset	Section/Page
0x5A0	PCI Express PCI Interrupt Mask Register (PEX_SS_INTR_MASK)	R/W	0x0000_003F	15.4.8.3/15-77

15.4.2.1, 15-22 Add the following paragraph immediately after the second paragraph:

“Note that, all the EP BARs are enabled by default. To disable a particular unused EP BAR and make it invisible to Root Complex's enumeration software, the corresponding BAR_ENABLE bit of the PEX_BAR_ENABLE register must be cleared. The EP local host must ensure that only those EP BARs in use are enabled. For further details, see Section 15.4.7.1 “PCI Express BAR Enable Register (PEX_BAR_ENABLE).””

15.4.7, 15-73 Add the following section for PCI Express BAR Enable Register (PEX_BAR_ENABLE). This section appears prior to 15.4.7.1, “PCI Express BAR Size Low Configuration Register (PEX_BAR_SIZEL).” The subsequent sections, figures, and tables are re-numbered accordingly.

15.4.7.1 PCI Express BAR Enable Register (PEX_BAR_ENABLE)

PEX_BAR_ENABLE, shown in Figure 15-91, is used to enable BARs. It supports a maximum of four BARs for the controller. To enable a given BAR, the bit corresponding to the BAR has to be set.

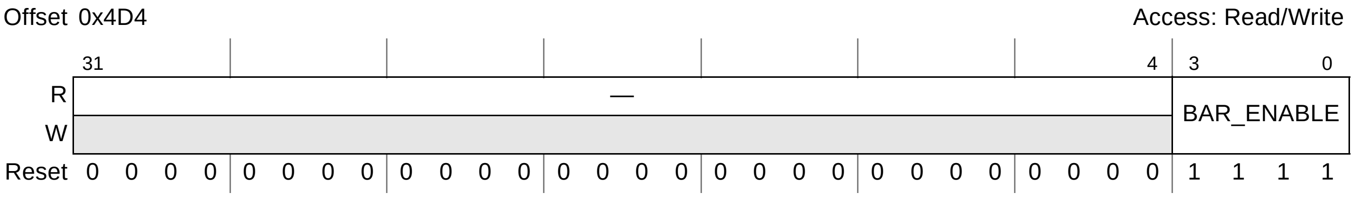


Figure 15-91. PCI Express BAR Enable Register (PEX_BAR_ENABLE)

The fields of the PEX_BAR_ENABLE register are described in Table 15-89.

Table 15-89. PEX_BAR_ENABLE Field Descriptions

Bits	Name	Description
31–4	—	Reserved. Must be zeros
3–0	BAR_ENABLE	BAR enable signal for all the BARs in PEX. A given BAR will be enabled only if the corresponding bit in this register is enabled. Bit 0 corresponds to BAR0, bit 1 to BAR1 and so on. For example, if this field is 0000001111, all four BARs are enabled.

15.4.7.1, 15-73 In Figure 15-91, “PCI Express BAR Size Low Configuration Register (PEX_BAR_SIZEL),” change the access from “Mixed” to “Read/Write.”

15.4.7.1, 15-74 In Table 15-89, “PEX_BAR_SIZEL Fields Description,” add a note “(for 64 bit BAR only)” to the bit field description of MASK (bits 31–12) as follows:

31–12	MASK	Mask. Sets the mask for the BAR, and any bit with a value of zero is masked. When the RC does a configuration write to the BAR during the enumeration sequence, bits that are masked cannot be modified and remain zeros. All ones and zeros in this register must be consecutive. The actual size is according to the location of the least significant bit in the MASK[31:12] field, which is set. If MASK[31:m] is all ones, the size is 2^m bytes. If MASK[31:12] is all zeros, the window size is 4 Gigabytes. For example: 1111...1111 - 2^{12}, 4 Kilobytes window. 1111...1110 - 2^{13}, 8 Kilobytes window. ... 1100...0000 - 2^{30}, 1 Gigabytes window. 1000...0000 - 2^{31}, 2Gigabytes window. 0000...0000 - 4 Gigabytes window (for 64 bit BAR only).

15.4.7.1, 15-73 Update the reset value in Figure 15-91, "PCI Express BAR Size Low Configuration Register (PEX_BAR_SIZEL)" as shown in the figure below:

Offset 0x4D8 Access: Read/Write

	31																			12	11											0
R	MASK																				—											
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-91 PCI Express BAR Size Low Configuration Register (PEX_BAR_SIZEL)

Also in Table 15-89, "PEX_BAR_SIZEL Fields Description," modify the "Description" column value for the "MASK" bit as follows:

Table 15-89. PEX_BAR_SIZEL Fields Description

Bits	Name	Description
31–12	MASK	Mask. Sets the mask for the BAR, and any bit with a value of zero is masked. When the RC does a configuration write to the BAR during the enumeration sequence, bits that are masked cannot be modified and remain zeros. All ones and zeros in this register must be consecutive. The actual size is according to the location of the least significant bit in the MASK[31:12] field, which is set. If MASK[31:m] is all ones, the size is 2^m bytes. If MASK[31:12] is all zeros, the window size is 4 Gigabytes. For example: 1111...1111 - 2^{12}, 4 Kilobytes window. 1111...1110 - 2^{13}, 8 Kilobytes window. 1111...1100 - 2^{14}, 16 Kilobytes window (default). ... 1100...0000 - 2^{30}, 1 Gigabytes window. 1000...0000 - 2^{31}, 2 Gigabytes window. 0000...0000 - 4 Gigabytes window (for 64 bit BAR only).

15.4.7.2, 15-74 In Figure 15-92, "PCI Express BAR Select Configuration Register (PEX_BAR_SEL)," change the access from "Mixed" to "Read/Write" and change the reset values to "all zeros."

15.4.7.3, 15-75 In Figure 15-93, "PCI Express BAR Prefetch Configuration Register (PEX_BAR_PF)," change reset values to "all zeros."

Section, Page No.	Changes
15.4.8.1, 15-75	In Figure 15-94, “PCI Express PME_To_Ack Timeout Register (PEX_PME_TO_ACK_TOR),” change the access from “Mixed” to “Read/Write.”
15.4.8.3, 15-77	In Figure 15-96, “PCI Express PCI Interrupt Mask Register (PEX_SS_INTR_MASK),” change the access from “Mixed” to “Read/Write.”
15.5.8.4, 15-99	Change bit field description of RSTIE (bit 1) in Table 15-122, “PEX_CSMIER Register Fields Description,” from: “PCI Express reset interrupt enable. If set, enables the generation of an interrupt when the PCI Express is reset.” to: “PCI Express reset interrupt enable. If set, enables the generation of an interrupt when the PCI Express is reset, for example, in the case of link down.”
15.5.8.8, 15-103	Change bit field description of RST (bit 1) in Table 15-126, “PEX_CSMISR Register Fields Description,” from: “PCI Express reset. Indicates that a PCI Express reset occurred.” to: “PCI Express reset. Indicates that a PCI Express reset or link down occurred. The PCI Express controller will not accept any further inbound or outbound transactions. To unlock the PCI Express controller: 1. Write one to clear PEX_CSMISR[RST]. 2. Perform a soft reset to the PCI Express CSB bridge by clearing and re-setting PECRn[CBRST]. 3. Reprogram the PCI Express CSB bridge registers.”
15.5.9, 15-104	Add the following section for PCI Express Slot Control Misc. Register after Section 15.5.9.1, “PCI Express PM Control Register (PEX_PM_CTRL).” Note that the subsequent figures and tables are re-numbered.

15.5.9.2 PCI Express Slot Control Misc Register

The PCI Express slot control misc register, shown in Figure 15-130, is used to control the PCI Express slot capability function.

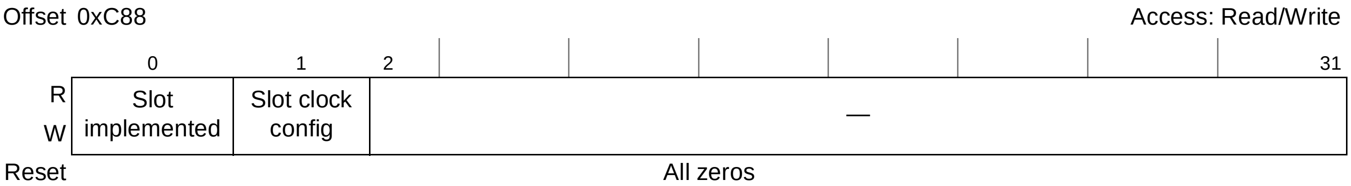


Figure 15-130. PCI Express Slot Control Misc Register

Table 15-128 describes the PCI Express slot control misc register fields.

Table 15-128. PCI Express Slot Control Misc Register Field Descriptions

Bits	Name	Description
0	Slot implemented	When set, this bit indicates that the PCI Express link associated with this port is connected to a slot.
1	Slot clock config	This bit indicates that the component uses the same physical reference clock that the platform provides on the connector. If the device uses an independent clock regardless of the presence of a reference on the connector, this bit must be cleared.
2–31	—	Reserved

15.5.11.1, 15-108 Modify the last row of Table 15-133, “PEX_EPIWTARn Register Fields Description” as follows:

Table 15-133. PEX_EPIWTAR*n* Register Fields Description

Bits	Name	Description
0	EN	Enable. If set, indicates that the address translation of the EP inbound window is enabled. **Note:** This bit only controls the enabling of the respective EP BAR's address translation. To entirely disable an EP BAR, refer to the description in the PEX_BAR_ENABLE register.

15.6.1.3, 15-114 Remove Section 15.6.1.3, “Byte Swapping”, Section 15.6.1.4, “Outbound Byte Swapping,” and Section 15.6.1.5, “Inbound Byte Swapping,” and replace with the following section:

15.6.1.3 Byte Ordering

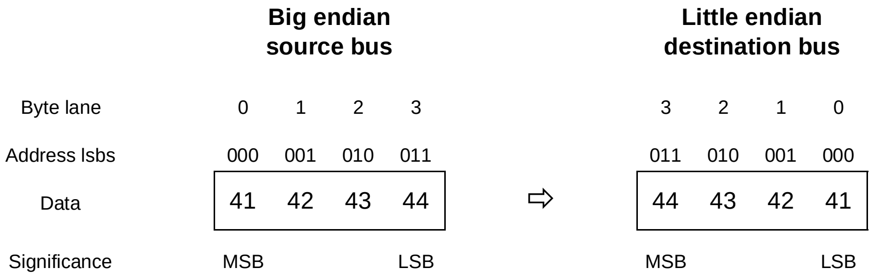
Whenever data must cross a bridge between two busses, the byte ordering of data on the source and destination buses must be considered. The internal platform bus of this device is inherently big endian and the PCI Express bus interface is inherently little endian.

There are two methods to handle ordering of data as it crosses a bridge—address invariance and data invariance. Address invariance preserves the addressing of bytes within a scalar data element, but not the relative significance of the bytes within that scalar. Conversely, data invariance preserves the relative significance of bytes within a scalar, but not the addressing of the individual bytes that make up a scalar.

This device uses address invariance as its default byte ordering policy, but it also may be configured to use a data invariance policy. The byte ordering policy is controlled by the ATMUs using the data invariance enable parameter in the inbound and outbound window attributes registers (PEXIWAR*n*[DIEN] and PEXOWAR*n*[DIEN]). Note that by using the ATMUs to control the byte ordering policy, it is possible to allow data invariance for certain devices or regions, while mapping other devices or regions using address invariance. However, in such cases, it is important that software keep track of which devices and regions use which byte ordering policy and access them accordingly.

As stated above, address invariance preserves the byte address of each byte on an I/O interface as it is placed in memory or moved into a register. This policy can have the effect of reversing the significance order of bytes (most significant to least significant and vice versa), but it has the benefit of preserving the format of general data structures. Provided that software is aware of the endianness and format of the data structure, it can correctly interpret the data on either side of the bridge.

Figure 15-142 shows the transfer of a 4-byte scalar, 0x4142_4344, from a big endian source across an address invariant bridge to a little endian destination.

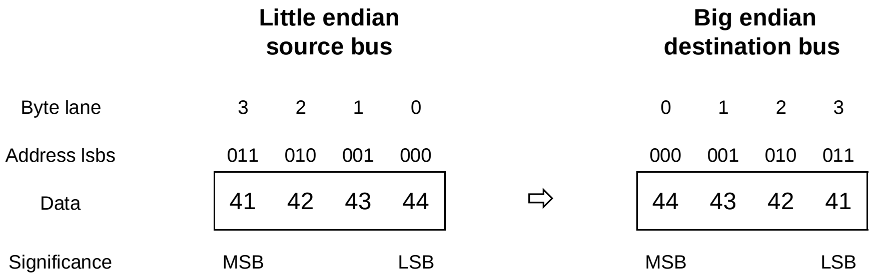


	Big endian source bus					Little endian destination bus			
Byte lane	0	1	2	3		3	2	1	0
Address lsbs	000	001	010	011		011	010	001	000
Data	41	42	43	44	⇨	44	43	42	41
Significance	MSB			LSB		MSB			LSB

Figure 15-142. Address Invariant Byte Ordering—4 bytes Outbound

Note that although the significance of the bytes within the scalar have changed, the address of the individual bytes that make up the scalar have not changed. As long as software is aware that the source of the data used a big endian format, the data can be interpreted correctly.

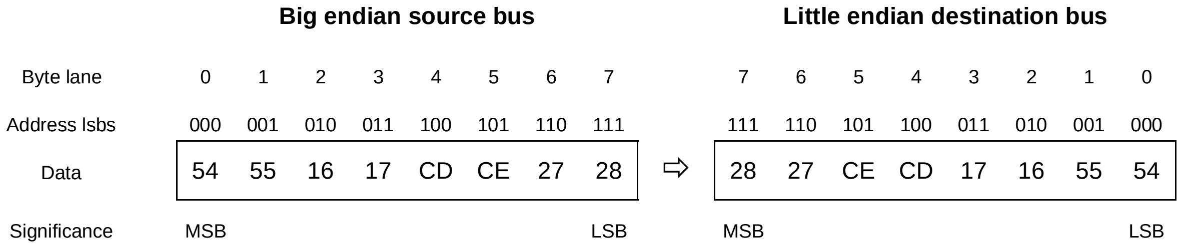
Figure 15-143 shows data flowing the other way, from a little endian source to a big endian destination.



	Little endian source bus					Big endian destination bus			
Byte lane	3	2	1	0		0	1	2	3
Address lsbs	011	010	001	000		000	001	010	011
Data	41	42	43	44	⇨	44	43	42	41
Significance	MSB			LSB		MSB			LSB

Figure 15-143. Address Invariant Byte Ordering—4 bytes Inbound

Figure 15-144 shows an outbound transfer of an 8-byte scalar, 0x5455_1617_CDCE_2728, using address invariance.



	Big endian source bus									Little endian destination bus							
Byte lane	0	1	2	3	4	5	6	7		7	6	5	4	3	2	1	0
Address lsbs	000	001	010	011	100	101	110	111		111	110	101	100	011	010	001	000
Data	54	55	16	17	CD	CE	27	28	⇨	28	27	CE	CD	17	16	55	54
Significance	MSB							LSB		MSB							LSB

Figure 15-144. Address Invariant Byte Ordering—8 bytes Outbound

Figure 15-145 shows an inbound transfer of a 2-byte scalar, 0x5837, using address invariance.

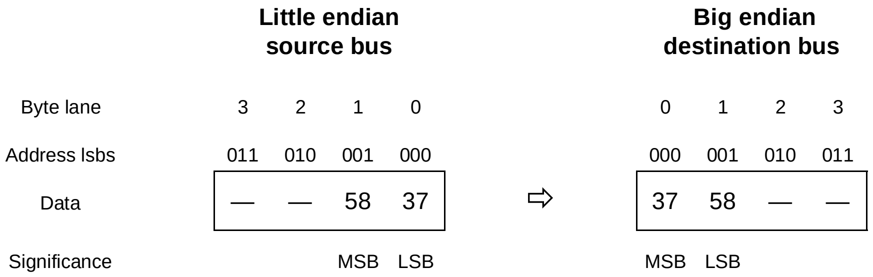


Figure 15-145. Address Invariant Byte Ordering—2 bytes Inbound

Note that in all of these examples, the original addresses of the individual bytes within the scalars (as created by the source) have been preserved.

15.6.1.3.1 Byte Order for Configuration Transactions

All internal memory-mapped registers in the CCSR space use big endian byte ordering. However, the PCI Express specification defines PCI Express configuration registers as little endian. All accesses to the PCI Express configuration port, PEX_CONFIG_DATA, including the those targeting the internal PCI Express configuration registers, use the address invariance policy as shown in Figure 15-146. Therefore, software must access PEX_CONFIG_DATA with little-endian formatted data—either using the **lwbrx**/**stwbrx** instructions or by manipulating the data before writing to and after reading from PEX_CONFIG_DATA.

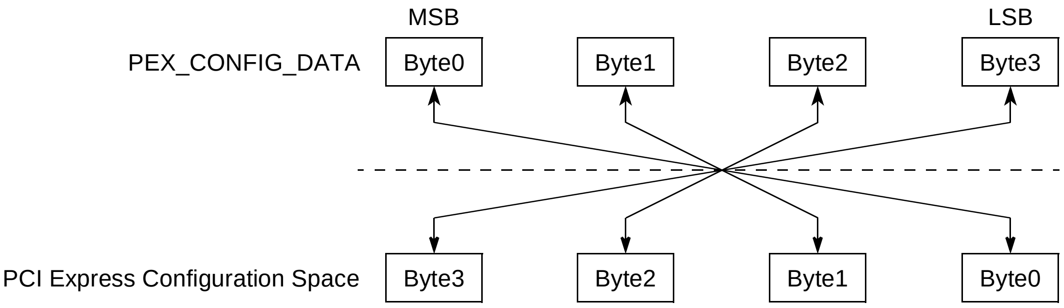


Figure 15-146. PEX_CONFIG_DATA Byte Ordering

15.8.1, 15-130

In Table 15-145, “DMA Descriptor Bit Fields Description,” update the description of bit 3 as follows:

“No snoop for CSB transactions.

0 The memory transaction is broadcast on the CSB as non-global (that is, not snooped).

1 The memory transaction is broadcast on the CSB as global (that is, snooped).”

15.8.4.4, 15-144

Replace the third sentence of the first paragraph, by removing “or directly in the descriptor register.” The modified sentence reads as follows:

“The descriptors programmed by software in host memory provide information to the hardware about the impending data transfer.”

Section, Page No. **Changes**

16.3.4, 16-20 In Table 16-17, “TransCfg Field Descriptions,” add the following note to the “Description” column for the bit “DFIS_SIZE.” The updated row is shown below:

Table 16-17. TransCfg Field Descriptions

Bits	Name	Description
31–16	DFIS_SIZE	Data FIS framing length words. Determines the maximum length each data FIS should be. **Note:** Maximum data FIS size is 8K, it may vary depending on data size to be transferred.

16.3.5.1, 16-29 In Table 16-26, “SYSPR Field Descriptions,” add the following row at the end of the table:

Table 16-26. SYSPR Field Descriptions

Bits	Name	Description
30–31	—	Reserved

Chapter 17, throughout Update the Interrupt Status Registers of all the execution units for their access type. Currently they are specified as R/W, which needs to be changed to RO.

17.3.2, 17-22 In Table 17-4, “Header Dword Bit Definitions,” add the following row at the end of the table:

Table 17-4. Header Dword Bit Definitions

Bits	Name	Description
32–63	—	Reserved

17.4.4.1, 17-38 In Figure 17-11, “Channel Configuration Register (CCR),” modify the offsets as shown in the figure below:

Offset Channel 1: 0x3_1108 Channel 1: 0x3_110C
Channel 2: 0x3_1208 Channel 2: 0x3_120C
Channel 3: 0x3_1308 Channel 3: 0x3_130C
Channel 4: 0x3_1408 Channel 4: 0x3_140C

Access: Read/Write

	0–28	29	30	31
R W	—	NPR	CON	R
Reset	All zeros			

	32–50	51	52–54	55	56	57	58	59	60	61	62	63
R W	—	PBS	—	BS	IWSE	—	EAE	CDWE	AWSE	NT	CDIE	—
Reset	All zeros											

Figure 17-11 Channel Configuration Register (CCR)

17.4.4.2, 17-41 In Figure 17-12, "Channel Status Register (CSR)," modify the offsets as shown in the figure below:

Offset Channel_1 0x3_1110 Channel 1: 0x3_1114
Channel_2 0x3_1210 Channel 2: 0x3_1214
Channel_3 0x3_1310 Channel 3: 0x3_1314
Channel_4 0x3_1410 Channel 4: 0x3_1414

Access: Read/Write

	0	1–7	8	9–15	16–22	23–31
R W	—	GET_STATE	—	PUT_STATE	—	MAIN_STATE
Reset	All zeros					

	32–34	35–39	40–43	44	45	46	47	48–59	60–63
R W	—	FF_LEVEL	—	PRD	SRD	PD	SD	Error	—
Reset	All zeros								

Figure 17-12 Channel Status Register (CSR)

Section, Page No. **Changes**

17.4.4.3, 17-44 In Table 17-16, “Current Descriptor Pointer Register Fields,” remove the first row and add the following two rows:

Table 17-16. Current Descriptor Pointer Register Fields

Bits	Name	Description
0–15	ID_TAG	Identification Tag. This value is copied from the ID_TAG field written by the host into the fetch FIFO (see Section 21.4.4.4, “Fetch FIFO Enqueue Register (FFER)”). **Note:** It should be noted that ID_TAG is optional. The Security Engine does not do anything with it other than copy it through to the descriptor header upon completion.
16–27	—	Reserved, must be cleared.

17.4.4.4, 17-44 In Figure 17-14, “Fetch FIFO Enqueue Register (FFER), replace “CUR_DES_PTR_ADRS” with “FETCH_ADR” and add “ID_TAG” for bits “0–15.” The updated figure is shown below:

Offset Channel 1: 0x3_1148, Channel 2: 0x3_1248,
Channel 3: 0x3_1348, Channel 4: 0x3_1448

Access: Write only

	0–15	16–27	28–31	32–63
R				
W	ID_TAG	—	EPTR	FETCH_ADR
Reset	All zeros			

Figure 17-14 Fetch FIFO Enqueue Register (FFER)

Also, replace the first row of Table 17-17, “Fetch FIFO Enqueue Register Field Descriptions” with the following two rows:

Table 17-17. Fetch FIFO Enqueue Register Field Descriptions

Bits	Name	Description
0–15	ID_TAG	Identification Tag. This value is copied from the ID_TAG field written by the host into the fetch FIFO (see Section 21.4.4.4, “Fetch FIFO Enqueue Register (FFER)”). **Note:** It should be noted that ID_TAG is optional. The Security Engine does not do anything with it other than copy it through to the descriptor header upon completion.
16–27	—	Reserved, must be cleared.

17.5.4.2, 17-51 In Table 17-19, “Field Names in Interrupt Enable, Interrupt Status, and Interrupt Clear Registers,” change the bit name for bit “24–31” from “Err and Dn bits for channels (CHN_1 to CHN_4)” to “Err and Dn bits for channels (CHN_4 to CHN_1).” The updated row is shown below:

Table 17-19. Field Names in Interrupt Enable, Interrupt Status, and Interrupt Clear Registers

Bits	Name	Description
24–31	Err and Dn bits for channels (CHN_4 to CHN_1)	Err 0 No error detected. 1 Error detected. Indicates that channel status register must be read to determine exact cause of the error. Dn 0 Not DONE. 1 DONE bit indicates that the corresponding channel has completed a descriptor.

Also, in the last row of the table, modify the bits column as shown below:

Table 17-19. Field Names in Interrupt Enable, Interrupt Status, and Interrupt Clear Registers

Bits	Name	Description
16–19, 32–35, 40–41, 44–45, 48–49, 52–53, 56–57, 60–61	—	Reserved, must be cleared.

17.5.4.4, 17-54 In Figure 17-19, “ID Register,” change the “Reset” row value for bit “31” from “0” to “1.” The updated figure is shown below:

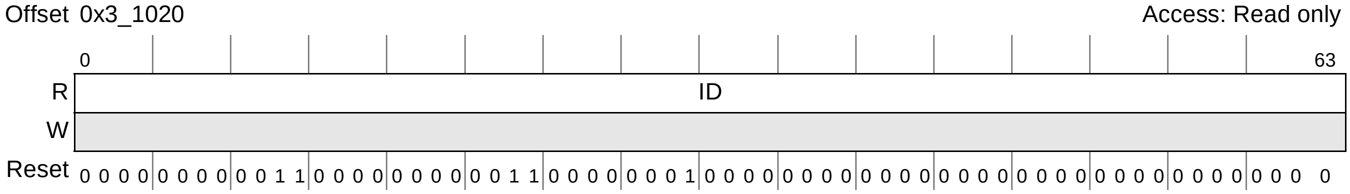


Figure 17-19 ID Register

17.7.1.2, 17-60 In Table 17-22, “AESU Mode Register Field Descriptions,” update bit 59 description for CMAC Cipher Mode as follows:

“CMAC Cipher Mode (ECM = 01, CM = 10): Load Keys—Do not compute E(K, 0128) to derive K1 and K2, but instead use the value loaded in Context Registers 3–4. This is useful after a context switch. Deriving K1 and K2 does not incur any timing penalty.

0	Compute E(K, 0128) and write it to Context Registers 5–6
1	Load E(K, 0128) and preserve it in Context Registers 5–6”

Section, Page No.	Changes
17.7.1.11.1, 17-70	In Section 17.7.1.11.1, “Context for Confidentiality Cipher Modes,” update the entire subsection, “Context for Counter (CTR) Cipher Mode,” as follows:

Context for Counter (CTR) Cipher Mode

In counter cipher mode, a random 128-bit initial counter value is incremented modulo 2^M with each block processed. The running counter is encrypted and XORed either with the plaintext to derive the ciphertext or with the ciphertext to recover the plaintext. The modulus exponent M can be set between 8 and 128, in multiples of 8.

There are two options for loading CTR mode context. When using descriptor type 0001_0, AES-CTR context is loaded as shown in Table 17-33. The Context In length must be set to 56B, and the context itself must be 32B of zeroes (context registers 1–4), followed by the initial counter value (context registers 5–6), and finally the modulus exponent M in context register 7. When using descriptor type 0000_0, 24B of context can be loaded as shown for SRT, dispensing with the need for the initial zeros.

17.7.1.11.1, 17-71	In Section 17.7.1.11.1, “Context for Confidentiality Cipher Modes,” update the entire subsection “Context for SRT Cipher Mode,” as follows:

Context for SRT Cipher Mode

As mentioned in the footnote to Table 17-23, SRT is not a new AES cipher mode but rather an AESU method of performing AES-CTR cipher mode with reduced context loading overhead. This mode was originally developed for SRTP, but is also applicable to the use of AES-CTR for LTE EEA2. SRT cipher mode can be used with descriptor type 0000_0 (aes_ctr_nosnoop) and type 0010_1 (SRTP). As with CTR cipher mode, a random 128-bit initial counter value is incremented modulo 2^M with each block processed. The running counter is encrypted and XORed with the plaintext to derive the ciphertext, or with the ciphertext to recover the plaintext. The modulus exponent M can be set between 8 and 128 in multiples of 8.

As shown in Table 17-33, in SRT mode context registers 1–2 hold the initial counter value, and context register 3 holds the modulus exponent M.

NOTE

There are two methods of performing AES-CTR with reduced context loading, as follows:

- Use descriptor type 0000_0 with the AES mode register set to CTR.
- Use descriptor type 0001_0 with the AES mode register set to SRT.

These methods are completely equivalent for cipher-only operations (no snooping for ICV generation). When performing AES-CTR in conjunction with HMAC-SHA-1, reduced context loading can only be achieved by using SRT.

Section, Page No. **Changes**

17.7.1.11.2, 17-71 Update Table 17-30, “AESU Context Registers for Integrity Modes,” as follows:

Table 17-30. AESU Context Registers for Integrity Modes

Context Register # (byte address)	Cipher Mode providing only Data Integrity		
	XCBC-MAC	GCM-GHASH	CMAC (OMAC1)
1 (0x34100)	Computed MAC	Computed MAC	Computed MAC
2 (0x34108)			
3 (0x34110)	Received MAC*		Received MAC*
4 (0x34118)			
5 (0x34120)	Key 3		E(K, 0^{128})
6 (0x34128)			
7 (0x34130)	Key 2	len(AAD)^{T}	
8 (0x34138)			
9 (0x34140)	Key 1**	H	
10 (0x34148)			
11 (0x34150)		len(AAD)^{C}	

Notes:

Context register 12 is unused for these modes

* Used only in ICV mode—must be written at start of new message for ICV checking

^{C} Length of data processed with current descriptor (in bits)

^{T} Length of total data (in bits)

17.7.1.11.2, 17-72 In Section 17.7.1.11.2, “Context for Data Integrity Cipher Modes,” update the entire subsection, “Context and Operation for XCBC-MAC Cipher Mode,” as follows:

Context and Operation for XCBC-MAC Cipher Mode

XCBC-MAC cipher mode is an authentication-only mode of AES. Normal CBC-MAC runs AES in CBC cipher mode and assigns the final ciphertext result as the MAC. XCBC-MAC supports only 16-byte keys.

The AESU supports three mode options while operating in XCBC-MAC cipher mode. These options are controlled by the AUX bits in the AESU mode register, as defined in the list below. The encrypt/decrypt bit has no meaning in an XCBC-MAC operation and is ignored by the AESU.

AUX0	Controls whether the XCBC-MAC is completed with this descriptor, or whether this descriptor is only doing a portion of the total MAC generation, and context needs to be output so that it can be reloaded for subsequent operations.
AUX1	Controls whether K1, K2, and K3 are initialized at the start of the descriptor or whether previously initialized keys are reloaded.
AUX2	Controls whether the AESU is to perform automatic checking of a received MAC against the newly calculated MAC.

For a descriptor that generates an XCBC-MAC over a full message, set AUX0 = 0, AUX1 = 0, AUX2 = 0. The descriptor's key length is loaded into the AESU Key Size Register, and the key itself is loaded into the AESU Key Registers 1U and 1L. The generated MAC is held in AESU context registers 1–2 and output according to the ICV Out length and pointer in the descriptor.

For a descriptor that generates an XCBC-MAC over a full message and compares the calculated MAC with the MAC received with the message, set AUX0 = 0, AUX1 = 0, AUX2 = 1. The descriptor's key length is loaded into the AESU key size register, and the key itself is loaded into the AESU key registers 1U and 1L. The generated MAC is held in AESU context registers 1–2 and compared to the received MAC, which the channel loads into AESU context registers 3–4 using the Extent field in the descriptor. Success or failure of the MAC comparison can be reported by an interrupt or through the ICCR1 bits in the modified descriptor header if header writeback is enabled.

Sometimes a message cannot be processed in a single descriptor. All data may not be present or the message may be larger than a single XCBC-MAC descriptor can process (> 64KB-1). In either case, this situation can be handled through combinations of AUX mode bits.

For the first descriptor, which processes the initial portion of the message, set AUX0 = 1, AUX1 = 0, AUX2 = 0. The descriptor's key length is loaded into the AESU Key Size Register, and the key itself is loaded into the AESU key registers 1U and 1L. The AESU generates K3, K2, K1 and stores them respectively in AESU context registers 5–6, 7–8, and 9–10. When the first descriptor has consumed all of its message data, it outputs the contents of context registers 1–10 using the Context Out length (80B) and pointer.

There can be an unlimited number of middle descriptors, which are neither the first nor last descriptor. Middle descriptors set AUX0 = 1, AUX1 = 1, AUX2 = 0. The descriptor's key length is set to 16B, and the key pointer is set to the address of key 1 (context registers 9–10) from the previous descriptor. This loads key 1 into AESU key size registers 1U and 1L. The descriptor’s Context In length is set to 64B, and the pointer is set to the address of context registers 1–8 from the previous descriptor.

When the middle descriptor has consumed all of its message data, it outputs the contents of context registers 1–10 using the Context Out length (80B) and pointer.

For the last descriptor, which processes the final portion of the message, set AUX0 = 0, AUX1 = 1, AUX2 = 0. The descriptor's key length is set to 16B, and the key pointer is set to the address of key 1 (context registers 9–10) from the previous descriptor. This loads key 1 into AESU key size registers 1U and 1L. The descriptor’s Context In length is set to 64B, and the pointer is set to the address of context registers 1–8 from the previous descriptor.

When the last descriptor has consumed all its message data, the generated MAC is held in AESU context registers 1–2 and output according to the ICV Out length and pointer in the descriptor. To compare the calculated MAC with the MAC received with the message, set AUX2 = 1 on the final descriptor. The generated MAC is held in AESU context registers 1–2 and compared to the received MAC, which is input using the Extent field in the descriptor. The channel overwrites the context reload values in context registers 3–4 with the real received MAC.

Section, Page No.	**Changes**
17.7.1.11.2, 17-73	Update the entire subsection, “Context and Operation for CMAC (OMAC1) Cipher Mode,” as follows:

Context and Operation for CMAC (OMAC1) Cipher Mode

CMAC cipher mode is an authentication-only mode of AES. CMAC may be specified using the following notation:

- E(K,L) denotes the AES-encrypt function;
- xtime(L) is defined as follows, where L is a 128-bit vector with L[127] as most significant bit:
  - If L[127] = 0, then xtime(L) = L<<1 (where ‘<<‘ denotes bitwise left shift)
  - Else xtime(L) = (L<<1) XOR 0x87.

The AESU supports three mode options while operating in CMAC mode. These options are controlled by the AUX bits in the AESU Mode Register, as defined in the list below. The encrypt/decrypt bit has no meaning in a CMAC operation and is ignored by the AESU.

AUX0	Controls whether the CMAC is completed with this descriptor or whether this descriptor is only doing a portion of the total MAC generation and context needs to be output so that it can be reloaded for subsequent operations.
AUX1	Controls whether K1, K2 are initialized at the start of the descriptor or whether previously initialized keys are reloaded.
AUX2	Controls whether the AESU is to perform automatic checking of a received MAC against the newly calculated MAC.

For a descriptor that generates a CMAC over a full message, set AUX0 = 0, AUX1 = 0, AUX2 = 0. The descriptor's key length is loaded into the AESU key size register, and the key itself is loaded into the AESU key registers. The generated MAC is held in AESU context registers 1–2 and output according to the ICV Out length and pointer in the descriptor.

Note that for some uses of CMAC, the message data itself may be modified to include packet-specific context in the CMAC generation process. For instance, when using AES-CMAC for the LTE EIA2 algorithm, a 64-bit, IV-like value is prepended to the PDCP header prior to calculating the MAC. The 64-bit value consists of COUNT (32 bit) || BEARER (5 bit) || Direction (1 bit) || Zeroes (26 bit).

For a descriptor that generates a CMAC over a full message and compares the calculated MAC with the MAC received with the message, set AUX0 = 0, AUX1 = 0, AUX2 = 1. The descriptor's key length is loaded into the AESU key size register, and the key itself is loaded into the AESU key registers. The generated CMAC is held in AESU context registers 1–2 and compared to the received CMAC, which the channel loads into AESU context registers 3–4 using the Extent field in the descriptor. Success or failure of the MAC comparison can be reported by an interrupt or through the ICCR1 bits in the modified descriptor header if header writeback is enabled.

Sometimes a message cannot be processed in a single descriptor. All data may not be present or the message may be larger than a single CMAC descriptor can process (> 64 KByte-1). In either case, this situation can be handled through combinations of AUX mode bits.

For the first descriptor, which processes the initial portion of the message, set AUX0 = 1, AUX1 = 0, AUX2 = 0. The descriptor's key length is loaded into the AESU key size register, and the key itself is loaded into the AESU key registers. The AESU internally generates CMAC context and stores it respectively in AESU context registers 5–6. When the first descriptor has consumed all its message data, it outputs the contents of context registers 1–6 using the Context Out length (48B) and pointer.

There can be an unlimited number of middle descriptors, which are neither the first nor last descriptor. Middle descriptors set AUX0 = 1, AUX1 = 1, AUX2 = 0. The descriptor's key length is set to 0. The descriptor’s Context In length is set to 48B, and the pointer is set to the address of context registers 1–6 from the previous descriptor.

When the middle descriptor has consumed all its message data, it outputs the contents of context registers 1–6, using the Context Out length (48B) and pointer.

For the last descriptor, which processes the final portion of the message, set AUX0 = 0, AUX1 = 1, AUX2 = 0. The descriptor's key length is set to 0. The descriptor’s Context In length is set to 48B, and the pointer is set to the address of context registers 1–6 from the previous descriptor.

When the last descriptor has consumed all its message data, the generated CMAC is held in AESU context registers 1–2, and output according to the ICV Out length and pointer in the descriptor. To compare the calculated MAC with the MAC received with the message, set AUX2 = 1 on the final descriptor. The generated CMAC is held in AESU context registers 1–2 and compared to the received CMAC, which is input using the Extent field in the descriptor. The channel overwrites the context reload values in context registers 3–4 with the real received MAC.

17.7.4.7, 17-114 Replace Figure 17-60, “DEU Interrupt Mask Register,” with the following figure:

Offset 0x3_2038 Access: Read/Write

	0–49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
R W	—	KPE	IE	ERE	CE	KSE	DSE	ME	AE	OFE	IFE	IFU	IFO	OFU	OFO
Reset	1000														

Figure 17-60 DEU Interrupt Mask Register

17.7.5.14, 17-129 In Figure 17-73, “KEU Key Data Register_1 (CK-high),” change the access from “Write-only” to “Read/Write”. In addition, remove orphan figure title, Figure 17-74, “KEU Key Data Register_1 (CK-high).”

17.7.5.14, 17-130 In Figure 17-75, “KEU Key Data Register_2 (CK-Low),” change the access from “Write-only to “Read/Write.” In addition, remove the orphan figure title, Figure 17-76, “KEU Key Data Register_2 (CK-Low).”

17.7.5.15, 17-130 In Figure 17-77, “KEU Key Data Register_3 (IK-high),” change the access from “Write-only” to “Read/Write.” In addition, remove the orphan figure titles—Figure 17-78, “KEU Key Data Register_3 (IK-high),” and Figure 17-80, “KEU Key Data Register_4 (IK-low).”

17.7.7.2, 17-146 Move Table 17-68, “ROUTINE Field Description,” from Section 17.7.7.2, “PKEU Key Size Register,” to Section 17.7.7.1, “PKEU Mode Register.”

17.7.7.7, 17-151 In Figure 17-98, “PKEU Interrupt Status Register,” modify the name of bit “51” from “EI” to “IE.” The updated figure is shown below:

Offset 0x3_C030 Access: Read

	0–48	49	50	51	52	53	54	55	56	57	58–63
R W	—	EVM	INV	IE	BE	CE	KSE	DSE	ME	AE	—
Reset	All zeros										

Figure 17-98 PKEU Interrupt Status Register

18.2, 18-3 Add the following information to the third and fifth sub-bullet of the second bullet:

“(carrier extend symbols in full duplex mode are not supported)”

The modified bullet items now reads as:

- — 1000 Mbps full-duplex RGMII and RTBI (carrier extend symbols in full duplex mode are not supported)
- — 1000 Mbps full-duplex SGMII (carrier extend symbols in full duplex mode are not supported)

18.4, 18-7 In Table 18-1, “eTSECn Network Interface Signal Properties,” change statement in the signal description for TSECn_TXD[3:0] from:

“RMII—TXD[3:2] unused; output driven zero”

to:

“RMII—TXD[3:2] unused”

18.5, 18-11 Remove the following sentence from the third paragraph:

“Writes to reserved register bits must always store 0, as writing 1 to reserved bits may have unintended side-effects.”

18.5.1, 18-12 Update Table 18-3, “Module Memory Map Summary,” to read as follows. Add a row for “Lossless Flow Control Registers” after “DMA system registers”.

Address Offset	Function
000–0FF	eTSEC general control/status registers
100–2FF	eTSEC transmit control/status registers
300–4FF	eTSEC receive control/status registers
500–5FF	MAC registers
600–7FF	RMON MIB registers
800–8FF	Hash table registers
900–AFF	—
B00–BFF	DMA system registers
C00–C3F	Lossless Flow Control registers
C40–DFF	—
E00–EFF	1588 Hardware Assist

Section, Page No.	Changes
18.5.2, 18-19	In Table 18-4, “Module Memory Map,” change the access designation for CAR1 and CAR2 registers to say as “w1c”. In addition, add the following eTSEC Lossless Flow Control Registers, after eTSEC DMA Attribute Registers in Table 18-4.

eTSEC1 Offset	Name	Access	Reset	Section/Page
eTSEC Lossless Flow Control Registers				
0x2_4C00	RQPRM0*—Receive Queue Parameters register 0	R/W	All zeros	18.5.3.9.1/18-108
0x2_4C04	RQPRM1*—Receive Queue Parameters register 1	R/W	All zeros	
0x2_4C08	RQPRM2*—Receive Queue Parameters register 2	R/W	All zeros	
0x2_4C0C	RQPRM3*—Receive Queue Parameters register 3	R/W	All zeros	
0x2_4C10	RQPRM4*—Receive Queue Parameters register 4	R/W	All zeros	
0x2_4C14	RQPRM5*—Receive Queue Parameters register 5	R/W	All zeros	
0x2_4C18	RQPRM6*—Receive Queue Parameters register 6	R/W	All zeros	
0x2_4C1C	RQPRM7*—Receive Queue Parameters register 7	R/W	All zeros	
0x2_4C20–0x2_4C40	Reserved	—	—	—
0x2_4C44	RFBPTR0*—Last Free RxBD pointer for ring 0	R/W	All zeros	18.5.3.9.2/18-109
0x2_4C48	Reserved	—	—	—
0x2_4C4C	RFBPTR1*—Last Free RxBD pointer for ring 1	R/W	All zeros	18.5.3.9.2/18-109
0x2_4C50	Reserved	—	—	—
0x2_4C54	RFBPTR2*—Last Free RxBD pointer for ring 2	R/W	All zeros	18.5.3.9.2/18-109
0x2_4C58	Reserved	—	—	—
0x2_4C5C	RFBPTR3*—Last Free RxBD pointer for ring 3	R/W	All zeros	18.5.3.9.2/18-109
0x2_4C60	Reserved	—	—	—
0x2_4C64	RFBPTR4*—Last Free RxBD pointer for ring 4	R/W	All zeros	18.5.3.9.2/18-109
0x2_4C68	Reserved	—	—	—
0x2_4C6C	RFBPTR5*—Last Free RxBD pointer for ring 5	R/W	All zeros	18.5.3.9.2/18-109
0x2_4C70	Reserved	—	—	—
0x2_4C74	RFBPTR6*—Last Free RxBD pointer for ring 6	R/W	All zeros	18.5.3.9.2/18-109
0x2_4C78	Reserved	—	—	—
0x2_4C7C	RFBPTR7*—Last Free RxBD pointer for ring 7	R/W	All zeros	18.5.3.9.2/18-109

Section, Page No.	**Changes**
18.5.3.1.3, 18-26	In Table 18-8, "IEVENT Field Descriptions," change the second sentence of the IEVENT[CRL] (bit 14) field description from: "The frame is discarded without being transmitted and transmission of the next frame commences." to: "The frame is discarded without being transmitted and the queue halts (TSTAT[THLTn] set to 1)."
18.5.3.1.6, 18-31	In Table 18-11, "ECNTRL Field Descriptions," update ECNTRL[CLRCNT] (bit 17) field description to read as follows: "Clear all statistics counters and carry registers. 0 Allow MIB counters to continue to increment and keep any overflow indicators. 1 Reset all MIB counters and CAR1 and CAR2. This bit is self-resetting."
18.5.3.1.6, 18-31	In Table 18-11, "ECNTRL Field Descriptions," update ECNTRL[AUTOZ] (bit 18) field description to read as follows: "Automatically zero MIB counter values and carry registers. 0 The user must write the addressed counter zero after a host read. 1 The addressed counter value is automatically cleared to zero after a host read. This is a steady state signal and must be set prior to enabling the Ethernet controller and must not be changed without proper care."
18.5.3.1.6, 18-32	In Table 18-11, "ECNTRL Field Descriptions," update ECNTRL[GMIIM] bit definition to read as follows: "GMII interface mode. Not supported."
18.5.3.1.6, 18-33	Update Table 18-12, "eTSEC Interface Configurations," to read as follows:

Table 18-12. eTSEC Interface Configurations

Interface Mode	ECNTRL Field							MACCFG2 Field
	FIFM[1]	GMIIM[2]	TBIM	RPM	R100M	RMM	SGMIIM	I/F Mode
MII 10/100 Mbps	0	0	0	0	0	0	0	01
RMII 100 Mbps	0	0	0	0	1	1	0	01
RMII 10 Mbps	0	0	0	0	0	1	0	01
RGMII 1Gbps	0	0	0	1	0	0	0	10
RGMII 100 Mbps	0	0	0	1	1	0	0	01
RGMII 10 Mbps	0	0	0	1	0	0	0	01
RTBI 1Gbps	0	0	1	1	0	0	0	10
SGMII 1 Gbps	0	0	1	0	0	0	1	10

Table 18-12. eTSEC Interface Configurations (continued)

Interface Mode	ECNTRL Field							MACCFG2 Field
	FIFM[1]	GMIIM[2]	TBIM	RPM	R100M	RMM	SGMIIM	I/F Mode
SGMII 100 Mbps	0	0	1	0	1	0	1	01
SGMII 10 Mbps	0	0	1	0	0	0	1	01

[1] FIFM bit is not supported.

[2] GMIIM bit is not supported.

18.5.3.1.8, 18-35	In Table 18-14, “DMACTRL Field Descriptions,” replace the DMACTRL[TOD] (bit 29) field description for “1” definition with the following: “1 eTSEC immediately fetches a new TxBD from ring 0.”
18.5.3.2.1, 18-38	In Table 18-16, “TCTRL Field Descriptions,” change TCTRL[TXSCHED] (bits 29–30) field description for “01” state to read as follows: “01 Priority scheduling mode. Frames from enabled TxBD rings are serviced in ascending ring index order.”
18.5.3.3.1, 18-48	In Figure 18-22, “RCTRL Register Definition,” change bit 17 from “reserved” to “LFC” as follows:

Offset eTSEC1:0x2_4300; eTSEC2:0x2_5300; Access: Read/Write

	0	7	8 10	11 15
R W	L2OFF	TS	—	PAL
Reset	All zeros			

	16	17	18	19	20	21	22	23	24 25	26	27	28	29	30	31
R W	—	LFC	VLEX	FILREN	FSQEN	GHTX	IPCSEN	TUCSEN	PRSDEP	—	BC_REJ	PROM	RSF	EMEN	—
Reset	All zeros														

Figure 18-22. RCTRL Register Definition

In addition, add the description for bit 17 to Table 18-27, “RCTRL Field Descriptions,” as follows:

Bits	Name	Description
17	LFC	Lossless flow control. When set, the eTSEC determines the number of free BDs (through RQPARM*n*[LEN] and RBTPTR*n*) in each active ring. Should the free BD count in an active ring drop below its setting for RQPARM*n*[FBTHR], the eTSEC asserts link layer flow control. For full-duplex ethernet connections, the eTSEC emits a pause frame as if TCTRL[TFC_PAUSE] was set. For FIFO packet interface connections, the RFC signal is asserted. 0 Disabled. This is the default 1 Enabled, calculate the free BDs in each active ring and assert link layer flow control if required.

18.5.3.3.8, 18-60

In Table 18-34, “RQFPR Field Descriptions,” append the following information with “TOS” field description (PID=1010, bits 24-31):

“(Software should acknowledge the PIC=1 IP6 bit to distinguish proper alignment of the TOS field.)” The updated row is shown below:

Table 18-34. RQFPR Field Descriptions

PID	Bit	Name	Description
1010	0–23	—	Reserved, should be written with zero.
	24–31	TOS	IPv4 header Type Of Service field or IPv6 Traffic Class field. This value defaults to 0x00 (default RFC 2474 best-effort behavior) if no IP header appeared. Note that for IPv6 the Traffic Class field is extracted using the IP header definition in RFC 2460. IPv6 headers formed using the earlier RFC 1883 have a different format and must be handled with software. (Software should acknowledge the PIC=1 IP6 bit to distinguish proper alignment of the TOS field.)

18.5.3.5.1, 18-67

In Table 18-39, “MACCFG1 Field Descriptions,” add the following note to Rx_Flow (bit 26) and Tx_Flow (bit 27):

“Note: Should not be set when operating in Half-Duplex mode.”

18.5.3.5.2, 18-69

In Table 18-40, “MACCFG2 Field Descriptions,” for MACCFG2[Huge Frame] (bit 26) field description, replace the right-hand “Buffer descriptor updated” column as follows:

Buffer descriptor updated
yes
no
yes
no

18.5.3.5.5, 18-72

In Table 18-43, “MAXFRM Descriptions,” add minimum MAXFRM value requirement for bits 16–31. First sentence of field description now reads as follows:

“This field is set to 0x0600 (1536 bytes) by default and always must be set to a value greater than or equal to 0x0040 (64 bytes), but not greater than 0x2580 (9600 bytes).”

18.5.3.5.6, 18-73

In Table 18-44, “MIIMCFG Field Descriptions,” remove references to CCB clock.

In addition, update the bit field description of MgmtClk (bits 29–31) as follows:

Bits	Name	Description
29–31	MgmtClk	This field determines the clock frequency of the MII management clock (EC_MDC). Its default value is 111. **Note:** The eTSEC system clock is selected by the SCCR register. (See Chapter 4, “Reset, Clocking, and Initialization.”) 00x 1/4 of the eTSEC system clock divided by 8 010 1/6 of the eTSEC system clock divided by 8 011 1/8 of the eTSEC system clock divided by 8 100 1/10 of the eTSEC system clock divided by 8 101 1/14 of the eTSEC system clock divided by 8 110 1/20 of the eTSEC system clock divided by 8 111 1/28 of the eTSEC system clock divided by 8

18.5.3.6, 18-79 Add the following note to the end of this section, following an existing note, and prior to Section 18.5.3.6.1, “Transmit and Receive 64-Byte Frame Counter (TR64)”

NOTE

The transmit and receive frame counters (TR64, TR127, TR 255, TR511, TR1K, TRMAX, and TRMGV) do not increment for aborted frames (collision retry limit exceeded, late collision, underrun, EBERR, TxFIFO data error, frame truncated due to exceeding MAXFRM, or excessive deferral).”

18.5.3.6.17, 18-87 In Table 18-71, “RFLR Field Descriptions,” update the bit description of RFLR (bits 16–31) to read as follows:

“Receive frame length error counter. Increments for each frame received in which the 802.3 length field did not match the number of data bytes actually received (46–1500 bytes). The counter does not increment if the length field is not a valid 802.3 length, such as an Ethertype value.

Frames tagged with a single VLAN tag are checked for valid length based on bytes 17-18 (rather than 13-14). Frames tagged (stacked) with multiple VLAN tags are not checked for valid length.”

18.5.3.6.41, 18-99 In Table 18-95, “TOVR Field Descriptions,” update TOVR[TOVR] (bits 20-31) field description to read as follows:

“Transmit oversize frame counter. Increments each time a frame is transmitted which exceeds 1518 (non VLAN) or 1522 (VLAN) with a correct FCS value.”

18.5.3.6.44, 18-100 In Figure 18-94, “Carry Register 1 (CAR1) Register Definition,” change the access designation for CAR1 register to show as “w1c”.

18.5.3.6.45, 18-101 In Figure 18-95, “Carry Register 2 (CAR2) Register Definition,” change the access designation for CAR2 register to show as “w1c”.

Section, Page No.	Changes
18.5.3.8.1, 18-108	Add the section for Lossless Flow Control Configuration Registers, after Section 18.5.3.8.1, “Attribute Register (ATTR).” Note that the subsequent sections, figures and tables are re-numbered.

18.5.3.9 Lossless Flow Control Configuration Registers

When enabled through RCTRL[LFC], the eTSEC tracks location of the last free BD in each Rx BD ring through the value of RFBPTR*n*. Using this pointer and the ring length stored in RQPRM*n*[LEN], the eTSEC continuously calculates the number of free BDs in the ring. Whenever the calculated number of free BDs in the ring drops below the pause threshold specified in RQPRM*n*[FBTHR], the eTSEC issues link layer flow control. It continues to assert flow control until the free BD count for each active ring reaches or exceeds RQPRM*n*[FBTHR]. See section Section 18.6.5.1, “Back Pressure Determination through Free Buffers,” for the theory of operation of these registers.

18.5.3.9.1 Receive Queue Parameters 0–7 (RQPRM0–PQPRM7)

The RQPRM*n* registers specify the minimum number of BDs required to prevent flow control being asserted and the total number of Rx BDs in their respective ring. Whenever the free BD count calculated by the eTSEC for any active ring drops below the value of RQPRM*n*[FBTHR] for that ring, link level flow control is asserted. Software must not write to RQPRM*n* while LFC is enabled and the eTSEC is actively receiving frames. However, software may modify these registers after disabling LFC by clearing RCTRL[LFC]. Note that packets may be lost due to lack of RxBDs while RCTRL[LFC] is clear. Software can prevent packet loss by manually generating pause frames (through TCTRL[TFC_PAUSE]) to cover the time when RCTRL[LFC] is clear. Figure 18-102 describes the definition for the RQPRM*n* register.

Offset eTSEC1:0x2_4C00+4×*n*; eTSEC2:0x2_5C00+4×*n*; Access: Read/Write

	0–7	8–31
R W	FBTHR	LEN
Reset	All zeros	

Figure 18-102. RQPRM Register Definition

Table 18-106 describes the fields of the RQPRM register.

Table 18-106. RQPRM Field Descriptions

Bits	Name	Description
0–7	FBTHR	Free BD threshold. Minimum number of BDs required for normal operation. If the eTSEC calculated number of free BDs drops below this threshold, link layer flow control is asserted.
8–31	LEN	Ring length. Total number of Rx BDs in this ring.

18.5.3.9.2 Receive Free Buffer Descriptor Pointer Registers 0–7 (RFBPTR0–RFBPTR7)

The RFBPTR*n* registers specify the location of the last free buffer descriptor in their respective ring. These registers live in the same 32b address space – and must share the same 4 most significant bits – as RBPTR*n*.

That is, RFBPTR*n* and its associated RBPTR*n* must remain in the same 256MB page. Like RBPTR*n*, whenever RBASE*n* is updated, RFBPTR*n* is initialized to the value of RBASE*n*. This indicates that the ring is completely empty. As buffers are freed and their respective BDs are returned (by setting the EMPTY bit) to the ring, software is expected to update this register. The eTSEC then performs modulo arithmetic involving RBASE*n*, RBPTR*n* and RFBPTR*n* to determine the number of free BDs remaining in the ring. If, at any stage, the value written to RFBPTR*n* matches that of the respective RBPTR*n* the eTSEC free BD calculation assumes that the ring is now completely empty. For more information on the recommended use of these registers, see Section 18.6.5.1, “Back Pressure Determination through Free Buffers.”
Figure 18-103 describes the definition for the RFBPTR*n* register.

Offset eTSEC1:0x2_4C44+8×*n*; eTSEC2:0x2_5C44+8×*n*; Access: Read/Write

	0–28	29–31
R W	RFBPTR*n*	—
Reset	All zeros	

Figure 18-103. RFBPTR0–RFBPTR7 Register Definition

Table 18-107 describes the fields of the RFBPTR*n* registers.

Table 18-107. RFBPTR0–RFBPTR7 Field Descriptions

Bits	Name	Description
0–28	RFBPTR	Pointer to the last free BD in RxBD Ring *n*. When RBASE*n* is updated, eTSEC initializes RFBPTR*n* to the value in the corresponding RBASE*n*. Software may update this register at any time to inform the eTSEC the location of the last free BD in the ring. Note that the 3 least-significant bits of this register are read only and zero.
29–31	—	Reserved.

18.5.3.9, 18-108 Change the Section title from “Hardware Assist for IEEE1588 Compliant Timestamping” to “IEEE 1588-Compatible Timestamping Registers.”

18.5.3.9.1, 18-108 Make bits 20 and 21 as reserved in Figure 18-102, “TMR_CTRL Register Definition,” to show as follows:

Offset eTSEC1:0x2_4E00 Access: Mixed

	0	1	2	3	4	5	6									15
R W	ALM1P	ALM2P	—	FS	PP1L	PP2L	TCLK_PERIOD									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

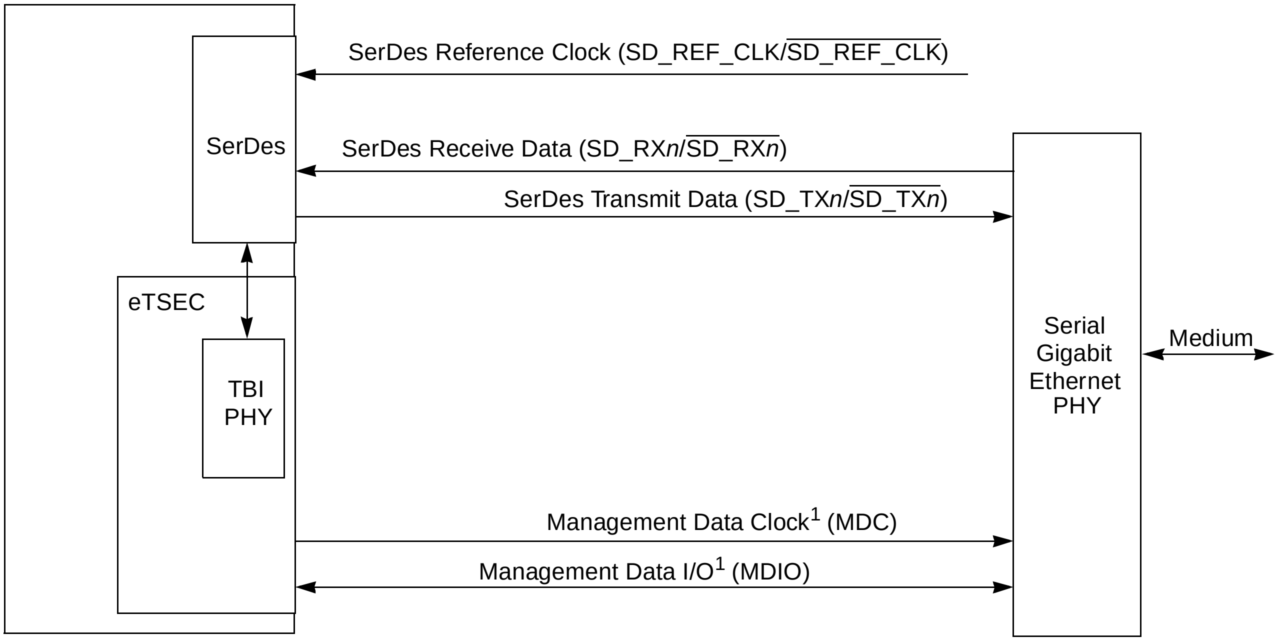
	16	17	18			21	22	23	24	25	26	27	28	29	30	31
R W	RTPE	FRD	—				ETEP2	ETEP1	COPH	CIPH	TMSR	—	BYP	TE	CKSEL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 18-102. TMR_CTRL Register Definition

Section, Page No.	Changes
	In addition, update Table 18-106, “TMR_CTRL Register Field Descriptions,” by removing the bit field descriptions of ESFDP (bit 20) and ESFDE (bit 21), and making them reserved.
18.5.3.9.1, 18-110	In Table 18-106, “TMR_CTRL Register Field Descriptions,” replace TMR_CTRL[CIPH] (bit 25) field description with the following: “Oscillator input clock phase. 0 non-inverted timer input clock 1 inverted timer input clock (NOTE: this setting is reserved if CKSEL=01.)”
18.5.4.3.10, 18-131	In Table 18-133, “TBICON Field Descriptions,” replace the bit field description of Clock select (bit 10) with the following: “Clock select. This bit selects how the on-chip TBI PHY is clocked. This bit is cleared by default. 0 The TBI PHY is clocked by dual split-phase 62.5 MHz receive clocks. These external signals must be provided via TBI receive clock 0 (TSEC*n*_RX_CLK) and TBI receive clock 1 (TSEC*n*_TX_CLK). If operating in SGMII mode, clearing this bit effectively disables the TBI PHY clock. 1 The TBI PHY is clocked by a single 125 MHz receive clock (required for SGMII operation). This single clock, if operating in a non-SGMII (parallel) mode, must be provided via the TBI receive clock 0 (TSEC*n*_RX_CLK) external signal. If operating in SGMII mode, this clock is provided on-chip by the SerDes block.”
18.6.1.4, 18-135	Add the following section after Section 18.6.1.4, “Reduced Ten-Bit Interface (RTBI).”

18.6.1.5 Serial Gigabit Media-Independent Interface (SGMII)

This section describes the serial gigabit media-independent interface (SGMII) intended to be used between a SerDes PHY and the eTSEC to implement a serial gigabit version of a media-independent interface. Figure 18-128 depicts the basic components of the SGMII including the signals required to establish eTSEC module connection with a PHY. Note that in SGMII the eTSEC utilizes the on-chip TBI PHY in addition to the SerDes interface.



[1] The management signals (MDC and MDIO) may be common to all of the Ethernet controllers' connections in the system, assuming that each PHY has a different management address.

Figure 18-128. eTSEC-SGMII Connection

18.6.1.5, 18-138 Add Table 18-137, "SGMII Signalling," to this section, just prior to the "Shared Signals" table (preceded by an introductory sentence) as follows:

Table 18-137 describes the signalling for SGMII interfaces. SGMII communication using the eTSEC is accomplished through the SerDes interface.

Table 18-137. SGMII Signalling

Signals	I/O	No. of Signals	Function
SD*n*_RX[*n*]	I	2	SGMII receive data (differential)
SD*n*_TX[*n*]	O	2	SGMII transmit data (differential)
SD*n*_REF_CLK	I	2	Reference clock (differential)
Sum		6	—

18.6.2.2, 18-140 Update the note in the third step from:

"(Note that SOFT_RESET must remain set for at least 3 TX clocks before proceeding.)"

to state as follows:

"(Note that SOFT_RESET must remain set for at least 3–10 TX clocks before proceeding.)"

Section, Page No. **Changes**

18.6.2.9, 18-150

Modify the second sentence of the third paragraph that formerly read as:

“Since the pause timer commences counting immediately upon receipt of a PAUSE frame, regardless of whether transmission is currently in progress, a sufficiently large pause time must be received to stop transmission past a frame of MTU size.”

to read as follows:

“The controller completes any frame in progress before stopping transmission and does not commence counting the pause time until transmit is idle.”

18.6.2.11, 18-153

Update the bulleted list after first paragraph to read as follows:

- The first TxBD pointer, TBPTRn, of any given frame is located at a 16-byte aligned address.
- All BDs for any multiple-BD frame reside in the same cache line.
- TCP/UDP and IP Checksum generation are disabled in each frame's TxFCB, or in TCTRL, or frames are limited to 1200 bytes in length
- Each TxBD[Data Length] is greater-than or equal to 64 bytes.

18.6.4.2.3, 18-165

Add GPI column to Table 18-150, “Special Filer Rules,” as follows:

Table 18-150. Special Filer Rules

Rule Description	RQCTRL Fields							RQPROP Word	RQCTRL Word[1]
	GPI	CLE	REJ	AND	Q	CMP	PID		
Default file—Always file frame to ring Q	0	0	0	0	Q	01	0000	All zeros	0x0000_*qq*20
Default reject—Always discard frame	0	0	1	0	000_000	01	0000	All zeros	0x0000_0120
Empty rule in AND—Always matches	0	0/1[2]	0	1	000_000	01	0000	0xFFFF_FFFF	0x0000_00A0
Empty rule in rule set—Always fails	0	0/1[3]	0	0	000_000	11	0000	0xFFFF_FFFF	0x0000_0060

[1] Hexadecimal digits *qq* denotes field Q shifted left 2 bits.

[2] Set CLE = 1 if the empty rule guards a cluster.

[3] Set CLE = 1 if the empty rule occurs at the end of a cluster.

18.6.4.2.1, 18-163

Add the following after the second paragraph:

“Note that in non-FIFO modes such as GMII running at slow platform frequency, MaxRules equation will also apply. For example, using 20 for IPG, a minimum of 190.5MHz eTSEC sysclk (381MHz platform) is needed to process all 256 entries in GMII for a 64-byte receive frame.”

18.6.4.2.6, 18-166

Add RQCTRL[GPI] column to Table 18-152, “Filer Table Example—802.1p Priority Filing,” as follows:

Table 18-152. Filer Table Example—802.1p Priority Filing

Table Entry	RQCTRL Fields							RQPROP	Comment	RQCTRL Word
	GPI	CLE	REJ	AND	Q	CMP	PID			
0	0	0	0	0	000_000	00	1001	0x0000_0007	File priority 7 to ring 0	0x0000_0009
1	0	0	0	0	000_001	00	1001	0x0000_0006	File priority 6 to ring 1	0x0000_0409
2	0	0	0	0	000_010	00	1001	0x0000_0005	File priority 5 to ring 2	0x0000_0809
3	0	0	0	0	000_011	00	1001	0x0000_0004	File priority 4 to ring 3	0x0000_0C09
4	0	0	0	0	000_100	00	1001	0x0000_0003	File priority 3 to ring 4	0x0000_1009
5	0	0	0	0	000_101	00	1001	0x0000_0002	File priority 2 to ring 5	0x0000_1409
6	0	0	0	0	000_110	00	1001	0x0000_0001	File priority 1 to ring 6	0x0000_1809
7	0	0	0	0	000_111	00	1001	All zeros	File undefined 802.1p or priority 0 to ring 7—Default always matches	0x0000_1C09

18.6.4.2.7, 18-167 Add RQCTRL[GPI] column to Table 18-153, “Filer Table Example—IP Diff-Serv Code Points Filing,” as follows:

Table 18-153. Filer Table Example—IP Diff-Serv Code Points Filing

Table Entry	RQCTRL Fields							RQPROP	Comment	RQCTRL Word
	GPI	CLE	REJ	AND	Q	CMP	PID			
0	0	0	0	0	001_000	01	1010	0x0000_00E0	File class 7 to queue 8 (TOS >= 0xE0)	0x0000_202A
1	0	0	0	0	001_001	01	1010	0x0000_00C0	File class 6 to queue 9 (TOS >= 0xC0)	0x0000_242A
2	0	0	0	0	001_010	01	1010	0x0000_00A0	File class 5 to queue 10 (TOS >= 0xA0)	0x0000_282A
3	0	0	0	0	001_011	01	1010	0x0000_0080	File class 4 to queue 11 (TOS >= 0x80)	0x0000_2C2A
4	0	0	0	0	000_100	01	1010	0x0000_0060	File class 3 to queue 4 (TOS >= 0x60)	0x0000_102A
5	0	0	0	0	001_100	01	1010	0x0000_0040	File class 2 to queue 12 (TOS >= 0x40)	0x0000_302A
6	0	0	0	0	010_100	01	1010	0x0000_0020	File class 1 to queue 20 (TOS >= 0x20)	0x0000_502A
7	0	0	0	0	011_100	01	1010	All zeros	File class 0 to queue 28 (TOS >= 0x00) or file to ring 4 by default	0x0000_702A

Section, Page No. **Changes**

18.6.4.2.8, 18-168 Add RQCTRL[GPI] column to Table 18-154, “Filer Table Example—TCP and UDP Port Filing,” to show as follows:

Table 18-154. Filer Table Example—TCP and UDP Port Filing

Table Entry	RQCTRL Fields							RQPROP	Comment	RQCTRL Word
	GPI	CLE	REJ	AND	Q	CMP	PID			
0	0	1	0	1	000_000	00	1011	0x0000_0006	Enter cluster if layer 4 is TCP	0x0000_028B
1	0	0	0	1	000_000	01	1111	0x0000_0014	AND rule—FTP from TCP ports 20 and 21: file to ring 2	0x0000_00AF
2	0	0	0	0	000_010	11	1111	0x0000_0016		0x0000_086F
3	0	0	0	0	000_011	00	1111	0x0000_0017	telnet from TCP port 23: file to ring 3	0x0000_0C0F
4	0	0	0	0	000_000	00	1111	All zeros	*empty entry reserved for future use*	0x0000_000F
5	0	0	0	0	000_000	00	1111	All zeros	*empty entry reserved for future use*	0x0000_000F
6	0	1	0	0	000_001	01	0000	All zeros	end cluster; default TCP: file to ring 1	0x0000_0620
7	0	1	0	1	000_000	00	1011	0x0000_0011	Enter cluster if layer 4 is UDP	0x0000_028B
8	0	0	0	0	000_101	00	1111	0x0000_0801	NFS from UDP port 2049	0x0000_140F
9	0	0	0	0	000_111	00	1111	0x0000_0208	Route from UDP port 520	0x0000_000F
10	0	0	0	0	000_110	00	1111	0x0000_0045	TFTP from UDP port 69	0x0000_180F
11	0	1	0	0	000_100	01	0000	All zeros	End cluster; default UDP: file to ring 4	0x0000_1220
12	0	0	0	0	000_000	01	0000	All zeros	By default, file to ring 0	0x0000_0020

18.6.4.2.9, 18-168 Add the following section for “Filer Example—Interrupt from Deep Sleep,”. This section appears after Section 18.6.4.2.8, “Filer Example—TCP and UDP Port Filing.”

18.6.4.2.9 Filer Example—Interrupt from Deep Sleep

The example in Table 18-155 shows how the filer can facilitate exit from deep sleep if any of the following packets arrive:

- ARP packet with Target IP address matching either of two IP addresses (either static or link local address)
- IPv4/UDP multicast DNS query
- IPv4/UDP SNMP broadcast query

These packets are also be stored in memory. All other packets are dropped.

Table 18-155. Filer Example—Interrupt from Deep Sleep

Table Entry	RQCTRL Fields							RQPROP	Comment
	GPI	**CLE**	**REJ**	**AND**	**Q**	**CMP**	**PID**		
0	0	0	0	0	000_000	11	0000	0x0001_0000	Check for ARP request; set mask register to mask off everything but the ARP request flag.
1	0	1	0	1	000_000	00	0001	0x0001_0000	Check to see if ARP request flag is set by doing a =1 comparison. Enter the “ARP Request Cluster” if true.
2	0	0	0	0	000_000	11	0000	0xFFFF_FFFF	ARP Cluster: Set Mask to unmask everything (Reset mask to all F’s)
3	1	0	0	0	000_001	00	1100	0xXXXX_XXXX	Compare the ARP Target IP address against “MY_IP_1”, which is indicated by the user-defined value of 0xXXXX_XXXX; if they match, accept the frame and generate an interrupt.
4	1	0	0	0	000_001	00	1100	0xYYYY_YYYY	Compare the ARP Target IP address against "MY_IP_2", which is indicated by the user-defined value of 0xYYYY_YYYY; if they match, accept the frame.
5	0	1	1	0	000_000	01	0000	0x0000_0000	Default rule that will always discard the packet; inserted here because an ARP request was received, but the Target IP address did not match either local IP addresses; therefore drop the packet and exit the cluster.
6	0	0	0	0	000_000	11	0000	0x0000_02D0	Set Mask for IP4 Packet (2D0), with IPv4 checksum checked and verified, and UDP header located.
7	0	0	0	1	000_000	00	0001	0x0000_02D0	Check to see if IP4 Packet, with IPv4 checksum checked and verified, and UDP header.
8	0	0	0	1	000_000	00	0000	0xFFFF_FFFF	Set Mask to unmask everything (Reset mask to all F's).
9	0	1	0	1	000_000	00	1011	0x0000_0011	Check against UDP protocol; if this passes, enter the cluster - all packets in the cluster are IPv4 packets with UDP protocol identified as the L4 protocol type.
10	0	0	0	1	000_000	00	0011	0x00XX_XXXX	Compare upper L2 DA bits to XX_XXXX (for multicast DNS query)
11	0	0	0	1	000_000	00	0100	0x00YY_YYYY	Compare lower L2 DA bits to YY_YYYY
12	0	0	0	1	000_000	00	1100	0xZZZZ_ZZZZ	Compare L3 Destination IP address to ZZZZ_ZZZZ
13	0	0	0	1	000_000	00	1110	0x0000_XXXX	Compare L4 destination port to XXXX
14	1	0	0	0	000_001	00	1111	0x0000_YYYY	If all of the previously consecutive ANDed conditions pass, multicast DNS Query has matched.

Table 18-155. Filer Example—Interrupt from Deep Sleep (continued)

Table Entry	RQCTRL Fields							RQPROP	Comment
	GPI	CLE	REJ	AND	Q	CMP	PID		
15	0	0	0	1	000_000	00	0011	0x00XX_XXXX	Compare upper L2 DA bits to XX_XXXX (for SNMP broadcast query).
16	0	0	0	1	000_000	00	0100	0x00YY_YYYY	Compare lower L2 DA bits to YY_YYYY.
17	1	0	0	0	000_001	00	1110	0x0000_ZZZZ	If all of the previously consecutive ANDed conditions pass, SNMP broadcast Query has matched.
18	0	1	1	0	000_000	01	0000	0x0000_0000	Cluster End: IPv4, UDP Comparison Default rule that will always discard the packet; inserted here because an IPv4 packet with L4=UDP request was received, but the profiles didn't match anything “interesting”; therefore drop the packet and exit the cluster.
19	0	0	1	0	000_000	01	0000	0x0000_0000	Default rule that will always discard the packet; inserted here no matches for any “interesting” packets were received that are used to wake up the CPU. All packets that reach this rule are discarded.

18.6.4.3.1, 18-169 Revise the section “Priority-Based Queuing (PBQ)” to read as follows:

18.6.4.3.1 Priority-Based Queuing (PBQ)

PBQ is the simplest scheduler decision policy. The enabled TxBD rings are assigned a priority value based on their index. Rings with a lower index have precedence over rings with higher indices, with priority assessed on a frame-by-frame basis. For example, frames in TxBD ring 0 have higher priority than frames in TxBD ring 1, and frames in TxBD ring 1 have higher priority than frames in TxBD ring 2, and so on.

The scheduling decision is then achieved as follows:

```
loop
     # start or S/W clear of TSATn
     ring = 0;
     while ring <= 7 loop
      if enabled(ring) and not ring_empty(ring) then
             transmit_frame(ring);
             ring = 0;
      else
             ring = ring + 1;
      endif
     endloop
endloop
```

Section, Page No.	Changes
18.6.4.3.2, 18-170	Add the following section after Section 18.6.4.3.2, “Modified Weighted Round-Robin Queuing (MWRR).” Note that subsequent sections, figures and tables are re-numbered.

18.6.5 Lossless Flow Control

The eTSEC DMA subsystem is designed to be able to support simultaneous receive and transmit traffic at gigabit line rates. If the host memory has sufficient bandwidth to support such line rates, then the principle cause of overflow on receive traffic is due to a lack of Rx BDs. Thus, the long term receive throughput is determined by the rate at which software can process receive traffic. If a user desires to prevent dropped packets, they can inform the far-end link to stop transmission while the software processing catches up with the backlog.

To avoid overflow in the latter case, back pressure must be applied to the far-end transmitter before the Rx descriptor controller encounters a non-empty BD and halts with a BSY error. As there is lag between application of back-pressure and response of the far-end, the pause request must be issued while there are still BDs free in the ring. In the traditional eTSEC descriptor ring programming model, there is no way for hardware to know how many free BDs are available, so software must initiate any pause requests required during operation. If software is backlogged, the request may be not be issued in time to prevent BSY errors. To allow the eTSEC to generate the pause request automatically, additional information (a pointer the last free BD and ring length) is required.

18.6.5.1 Back Pressure Determination through Free Buffers

Ultimately, the rate of data reception is determined by how quickly software can release buffers back into the receive ring(s). Each time a buffer is freed, the associated BD has its empty bit set and hardware is free to consume both. Thus the number of free BDs in a given Rx ring indicates how close hardware is to the end of that ring. To prevent data loss, back pressure should be applied when the number of free BDs drops below some critical level. The number of BDs that can be consumed by an incoming packet stream while back-pressure takes effect is determined by several factors, such as: receive traffic profile, transmit traffic profile, Rx buffer size, physical transmission time between eTSEC and far-end device and intra-device latency. Theoretically, the worst case is as follows:

$$\mathrm{FreeBDsRequired} = \frac{\mathrm{MaxFrameSize}}{\mathrm{MinFrameSize} + \mathrm{IFG}} + \frac{\mathrm{MaxFrameSize}}{\mathrm{RxBufferSize}} + \mathrm{LinkDelay}$$

This case comes about when:

- The eTSEC has just started transmitting a large frame and thus cannot send out a pause frame
- Upon reception of the pause request the far-end has just started transmission of a large frame
- The eTSEC receives a burst of short frames with minimum inter-frame-gap (96bit times for ethernet)

Once the user has determined the worst case scenario for their application, they program the required free BD threshold into the eTSEC (through RQPRM[PBTHR]). Since different BD rings may have different sizes and expected packet arrival rates, a separate threshold is provided for each active ring. It is recommended that a threshold of at least four BDs is the practical minimum for gigabit ethernet links.

For the Rx descriptor controller to determine the number of free BDs remaining in the ring, it needs to know the following:

1. The location of the current BD being used by hardware
2. The location of the last BD that was released (freed) by software
3. The length of the Rx BD ring.

For each active ring, the current BD pointer (RBPTR*n*) is maintained by the eTSEC. Software knows both the size of the Rx ring and the location of the last freed BD. By providing the eTSEC with those values (through RQPRM[LEN] and RFBPTR respectively) the eTSEC always know how many receive buffers are available to be consumed by incoming data.

The number of guaranteed free BDs in the ring is then determined by:

When RFBPTRn < RBPTR*n*

$$\mathrm{FreeBDs} = \mathrm{RQPRM}n[\mathrm{LEN}] - \mathrm{RBPTR}n + \mathrm{RFBPTR}n$$

When RFBPTRn > RBPTR*n*

$$\mathrm{FreeBDs} = \mathrm{RFBPTR}n - \mathrm{RBPTR}n$$

When RBPTR*n* = RFBPTR*n* the number of free BDs in the ring is either one (since RFBPTR*n* points to a free BD) or equal to the ring length. Since the BD pointed to by RBPTR*n* may be either in use or about to be used, it is not considered in the free BD count. To resolve the case where the two pointers collide, the following logic applies:

If RBASE*n* was updated and thus initializes both RBPTR*n* and RFBPTR*n*, the ring is deemed empty.

If RFBPTR*n* is updated by a software write and matches RBPTR*n*, the ring is deemed empty.

If HW updates RBPTR*n* and the result matches RFBPTR*n*, the ring is deemed to have one BD remaining. Upon writing this BD back to memory (indicating the buffer is occupied) the ring is deemed to be full.

Important. There is a possibility that if software is severely backlogged in updating RFBPTR*n*, the hardware could wrap around the ring entirely, consume exactly the remaining number of BDs and not halt with a BSY error. If software then increments RFBPTR*n* to the next address (thereby equalling RBPTR*n*), the hardware assumes the ring is now empty (when in fact there is only a single BD freed up). This results in the hardware failing to maintain back pressure on the far end. Upon software incrementing RFBPTR*n* a subsequent time, the wrap condition is successfully detected and hardware recognizes a nearly full ring (rather than a nearly empty one). Since software can increment RFBPTR*n* by any amount, it is not possible for hardware to determine in this case whether the user has cleared the entire ring or just one BD. Users can eliminate the possibility of this condition occurring by ensuring that RFBPTR*n* is incremented by at least two BDs each time (that is, clear at least two buffers whenever the RxBD unload routine is called).

Once the eTSEC determines that this threshold has been reached, back pressure is applied accordingly. The type of back pressure that is applied varies according to the physical interface that is used.

- **Half duplex Ethernet:** No support in this mode.
- **Full duplex Ethernet:** An IEEE 802.3 PAUSE frame (see Section 18.6.2.9/18-149) is issued as if the TCTRL[TFC_PAUSE] bit was set. An internal counter tracks the time the far end controller is expected to remain in pause (based on the setting of PTV[PT]). When that counter reaches half the value of PTV[PT], the eTSEC reissues a pause frame if the free BD calculation for any ring is below the threshold for that ring. For example, if PTV[PT] is set to 10 quanta, a pause frame is re-issued when five quanta have elapsed if the free BD threshold is still not met. A practical minimum for PTV[PT] of 4 quanta is recommended.
- **FIFO packet interface:** Link layer flow control is asserted through use of the RFC signal (CRS pin). Flow control is asserted for the entire time that free BD threshold is not met. The same mechanism is used for both GMII-style and encoded packet modes.

18.6.5.2 Software Use of Hardware-Initiated Back Pressure

18.6.5.2.1 Initialization

Software configures RBASE*n* and RQPRM*n*[LEN] according to the parameters for that ring. Then the number of free BDs that are required to prevent the eTSEC from automatically asserting flow control are programmed in RQPRM[FBTHR]. The receiver is then enabled.

Note: the act of programming RBASE*n* initializes RFBPTR*n* to the start of the of the ring. When the ring is in this initial empty state, there is no concept of a last freed BD. In this case, the calculated number of free BDs is the size of the ring. Since the BD that the hardware is currently pointing to is to be considered in-use, the free BD count is actually one higher than the total available. As soon as the hardware consumes a BD (by writing it back to memory), RBPTR*n* advances and the free BD count reflects the correct number of available free BDs.

18.6.5.2.2 Operation

As software frees BDs from the ring, it writes the physical address of the BD just freed to RFBPTR*n*. The eTSEC asserts flow control if the distance (using modulo arithmetic) between RBPTR*n* and RFBPTR*n* is < RQPRM*n*[FBTHR]. In multi-ring operation, if the free BD count of **any** active ring drops below the threshold for that ring, flow control is asserted. Once enough BDs are freed for **all** active rings to meet their respective free BD thresholds, application of back pressure cases.

Note: The eTSEC does not issue an exit pause frame (that is, pause frame with PTV of 0x0000) once all active rings have sufficient BDs. Instead, it waits for the far-end pause timer to expire and start re-transmission.

18.7.1.1, 18-186	In Table 18-162, “MII Mode Register Initialization Steps,” in the row for “Setup the MII Mgmt clock speed,” change the last sentence to say that the minimum frequency is 2.5 MHz as follows:

Setup the MII Mgmt clock speed, MIIMCFG[0000_0000_0000_0000_0000_0000_0000_0101] set source clock divide by 14 for example to insure that MDC clock speed is not less than 2.5 MHz

Section, Page No.	Changes
18.7.1.2, 18-190	In Table 18-165, “RGMII Mode Register Initialization Steps,” in the row for “Setup the MII Mgmt clock speed,” change the last sentence to say that the minimum frequency is 2.5 MHz as follows:

Setup the MII Mgmt clock speed, MIIMCFG[0000_0000_0000_0000_0000_0000_0000_0101] Set source clock divide by 14, for example, to insure that TSEC_MDC clock speed is not less than 2.5 MHz.

Section, Page No.	Changes
18.6.5.2, 18-171	Add the following after Figure 18-136, “1588 Timer Design Partition.”

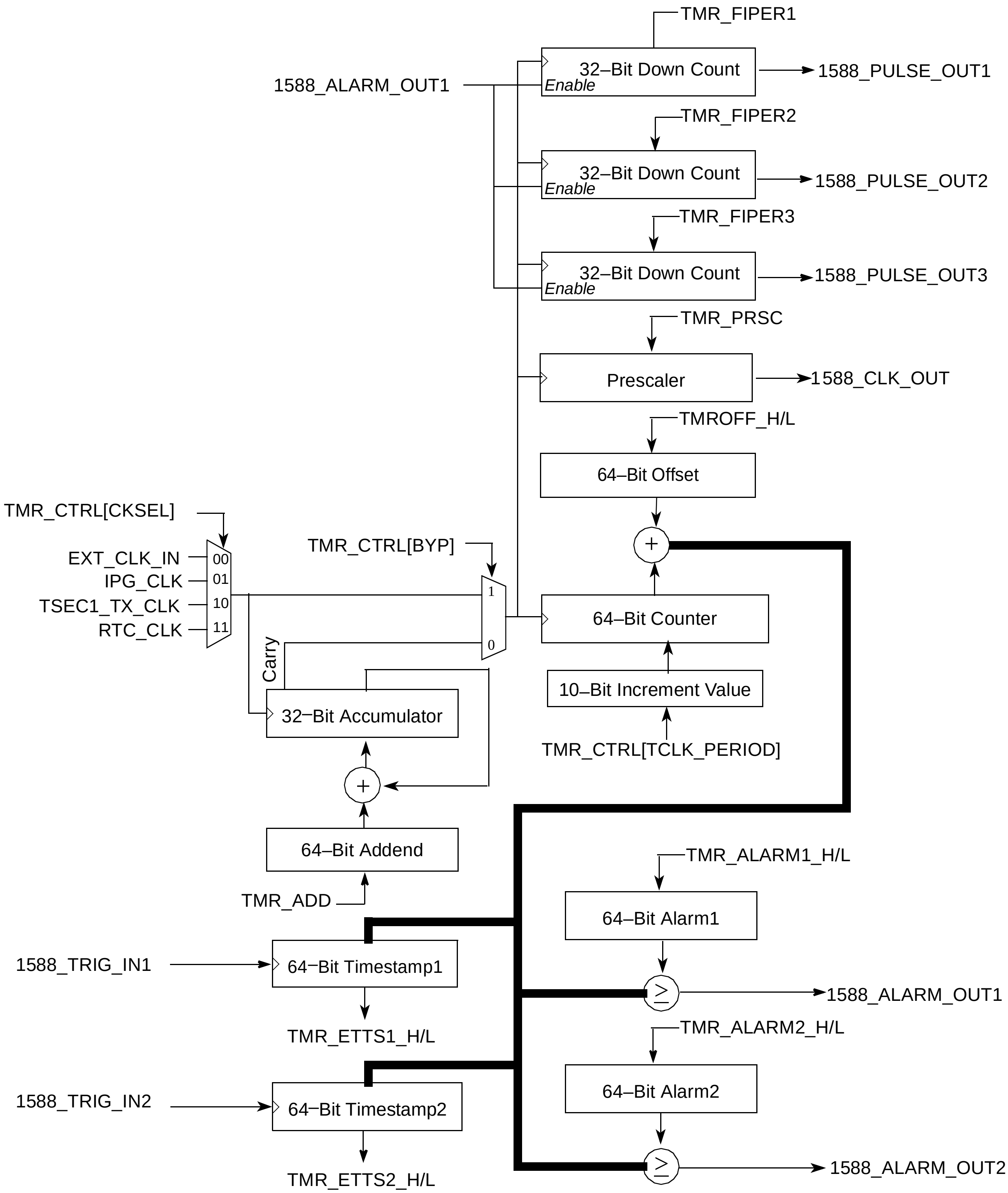


Figure 18-137 1588 Current Time Control

Section, Page No.	Changes

The 64-bit current time is controlled by timer enable, TMR_CTRL[TE], selected clock, TMR_CTRL[CKSEL], bypass mode, TMR_CTRL[BYP], the timer clock period, TMR_CTRL[TCLK_PERIOD], and the ADDEND value, TMR_ADD registers, as shown in Figure 18-137.

18.7.1.3, 18-193 Update the third row of Table 18-168, "RMII Mode Register Initialization Steps," from:

"Initialize MACCFG2,

MACCFG2[0000_0000_0000_0000_0111_0010_0000_0101]

(I/F Mode = 2, Full Duplex = 1)"

to:

"Initialize MACCFG2,

MACCFG2[0000_0000_0000_0000_0111_0001_0000_0101]

(I/F Mode = 1, Full Duplex = 1)"

18.7.1.3, 18-193 In Table 18-168, "RMII Mode Register Initialization Steps," in the row for "Setup the MII Mgmt clock speed," change the last sentence to say that the minimum frequency is 2.5 MHz as follows:

Setup the MII Mgmt clock speed, MIIMCFG[0000_0000_0000_0000_0000_0000_0000_1101] set system clock divide by 14 for example to insure that MDC clock speed = < 2.5 MHz

18.7.1.4, 18-197 In Table 18-171, "RTBI Mode Register Initialization Steps," in the row for "Setup the MII Mgmt clock speed," change the last sentence to say that the minimum frequency is 2.5 MHz as follows:

Setup the MII Mgmt clock speed, MIIMCFG[0000_0000_0000_0000_0000_0000_0000_0101] Set source clock divide by 14, for example, to insure that TSEC_MDC clock speed is not less than 2.5 MHz.

18.7.1.5, 18-200 In Table 18-173, "SGMII Mode Register Initialization Steps," in the row for "Setup the MII Mgmt clock speed," change the last sentence to say that the minimum frequency is 2.5 MHz as follows:

Setup the MII Mgmt clock speed, MIIMCFG[0000_0000_0000_0000_000_0000_0000_0101] set source clock divide by 14 for example to insure that MDC clock speed is not less than 2.5 MHz

Section, Page No.	Changes
19.3.1, 19-6	In Table 19-3, “SRDSnCR0 Field Descriptions,” update RXEQA (bits 2–3) bit field descriptions as follows: “Receive equalization selection bus for lane A—when asserted in PCI Express mode: 00 No equalization 01 2 dB of equalization 10 4 dB of equalization 11 Reserved Recommended setting per protocol: SATA: 01 PCI Express: 01 SGMII: 01”
19.3.1, 19-6	In Table 19-3, “SRDSnCR0 Field Descriptions,” update RXEQE (bits 6–7) bit field descriptions as follows: “Receive equalization selection bus for lane E—when asserted in PCI Express mode: 00 No equalization 01 2 dB of equalization 10 4 dB of equalization 11 Reserved Recommended setting per protocol: SATA: 01 PCI Express: 01 SGMII: 01”

Section, Page No. **Changes**

19.3.1, 19-7 In Table 19-3, “SRDSnCR0 Field Descriptions,” replace “SATA” recommended settings for the bits “TXEQA” and “TXEQE” with “000.” The updated rows are shown below:

Table 19-3. SRDSnCR0 Field Descriptions

Bits	Name	Description
17–19	TXEQA	Sets the peak value for output swing of transmitters and the amount of transmit equalization for lane A. Transmit equalization selection bus for lane A. If register field SRDSCR3[21:23] = 000, then the equalization definitions are: 000 No equalization 001 1.09x relative amplitude 010 1.2x relative amplitude 011 1.33x relative amplitude 100 1.5x relative amplitude 101 1.71x relative amplitude 110 2.0x relative amplitude 111 Reserved Recommended setting per protocol: • SATA: 000 • PCI Express: 100 • SGMII: 100 If register field SRDSCR3[21:23] = 101, then the equalization definitions are: 000 No equalization 001 1.17x relative amplitude 010 1.4x relative amplitude 011 1.75x relative amplitude 100 Reserved 101 Reserved 110 Reserved 111 Reserved
21–23	TXEQE	Sets the peak value for output swing of transmitters and the amount of transmit equalization for lane E. Transmit equalization selection bus for lane E. If register field SRDSCR3[29:31] = 000, then the equalization definitions are: 000 No equalization 001 1.09x relative amplitude 010 1.2x relative amplitude 011 1.33x relative amplitude 100 1.5x relative amplitude 101 1.71x relative amplitude 110 2.0x relative amplitude 111 Reserved Recommended setting per protocol: • SATA: 000 • PCI Express: 100 • SGMII: 100 If register field SRDSCR3[29:31]= 101, then the equalization definitions are: 000 No equalization 001 1.17x relative amplitude 010 1.4x relative amplitude 011 1.75x relative amplitude 100 Reserved 101 Reserved 110 Reserved 111 Reserved

Section, Page No.	Changes
19.3.1, 19-7	In Table 19-3, “SRDSnCR0 Field Descriptions,” update TXEQA (bits 17–19) bit field descriptions as follows: “Sets the peak value for output swing of transmitters and the amount of transmit equalization for lane A. Transmit equalization selection bus for lane A. If register field SRDSCR3[21:23] = 000, then the equalization definitions are: 000 No equalization 001 1.09x relative amplitude 010 1.2x relative amplitude 011 1.33x relative amplitude 100 1.5x relative amplitude 101 1.71x relative amplitude 110 2.0x relative amplitude 111 Reserved Recommended setting per protocol: SATA: select an allowed setting, based on the channel and the PCB characteristics PCI Express: 100 SGMII: 100 If register field SRDSCR3[21:23] = 101, then the equalization definitions are: 000 No equalization 001 1.17x relative amplitude 010 1.4x relative amplitude 011 1.75x relative amplitude 100 Reserved 101 Reserved 110 Reserved 111 Reserved”
19.3.1, 19-8	In Table 19-3, “SRDSnCR0 Field Descriptions,” replace TXEQE (bits 21–23) bit field descriptions with the following: “Recommended setting per protocol: SATA: select an allowed setting, based on the channel and the PCB characteristics PCI Express: 100 SGMII: 100 If register field SRDSCR3[29:31]= 101, then the equalization definitions are: 000 No equalization 001 1.17x relative amplitude 010 1.4x relative amplitude

011 1.75x relative amplitude”

19.3.4, 19-15 In Table 19-6, “SRDSnCR3 Field Descriptions,” replace SDTXLE (bits 26–31) bit field description with the following:

“Controls lane E transmitter amplitude levels.

[5:3] = Reserved

000 No amplitude reduction
001 0.916 × full swing
010 0.833 × full swing
011 0.750 × full swing
100 0.666 × full swing
101 0.583 × full swing
110 0.500 × full swing
111 Reserved

Recommended setting per protocol (SRDS*n*CR0[DPPA/E] = 0):

SATA: 101

PCI Express: 000

SGMII: 000”

Chapter 20, throughout

Replace “USB_DR” with “USB_DR controller” throughout the chapter.

Also, replace all instances of “FTSN” with “FSTN”.

20.3, 20-8 In Table 20-4, “USB Interface Memory Map,” modify the rows for the following registers:

Table 20-4. USB Interface Memory Map

Offset	Register	Access	Reset	Section/Page
0x14C	FRINDEX—USB frame index	R/W	0x0000_0000	20.3.2.4/20-20
0x2_3154	PERIODICLISTBASE—Frame list base address[1]	R/W	0x0000_0000	20.3.2.6/20-22
	DEVICEADDR—USB device address	R/W	0x0000_0000	20.3.2.7/20-22
0x184	PORTSC—Port status/control	Mixed	0x9000_0010	20.3.2.14/20-28
0x1CC	ENDPTCTRL3—Endpoint control 3	R/W	0x0000_0000	20.3.2.23/20-41

[1] This register has separate functions for the host and device operation; the host function is listed first in the table.

20.2.3, 20-8 Add the following note at the end of the first paragraph:

NOTE

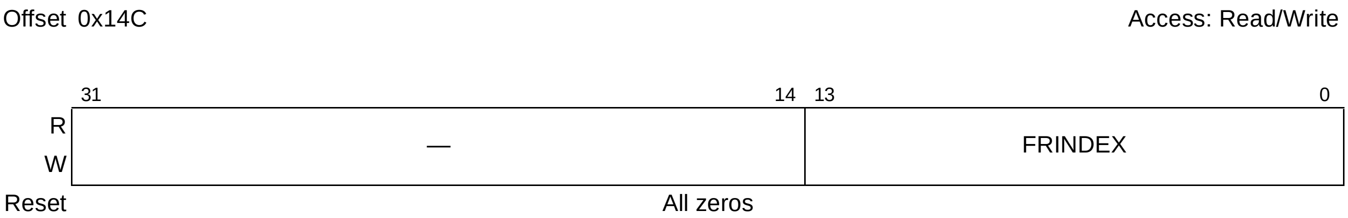
A write to registers in the USB controller memory map may cause the system to hang if PORTSC[PHCD]=0 when no USB PHY clock is applied.

Section, Page No.	Changes
20.3.1.2, 20-21	In Table 20-6, “HCIVERSION Register Field Descriptions,” change the “Name” field value from “—” to “HCIVERSION.” The updated row is shown below:

Table 20-6. HCIVERSION Register Field Descriptions

Bits	Name	Description
15–0	HCIVERSION	EHCI revision number. Value is 0x0100 indicating version 1.0.

20.3.2.4, 20-21	In Figure 20-11, “USB Frame Index (FRINDEX),” replace all values in the “Reset” row with “All Zeros.” The updated figure is shown below:



Offset 0x14C   Access: Read/Write

	31–14	13–0
R W	—	FRINDEX
Reset	All zeros	

Figure 20-11 USB Frame Index (FRINDEX)

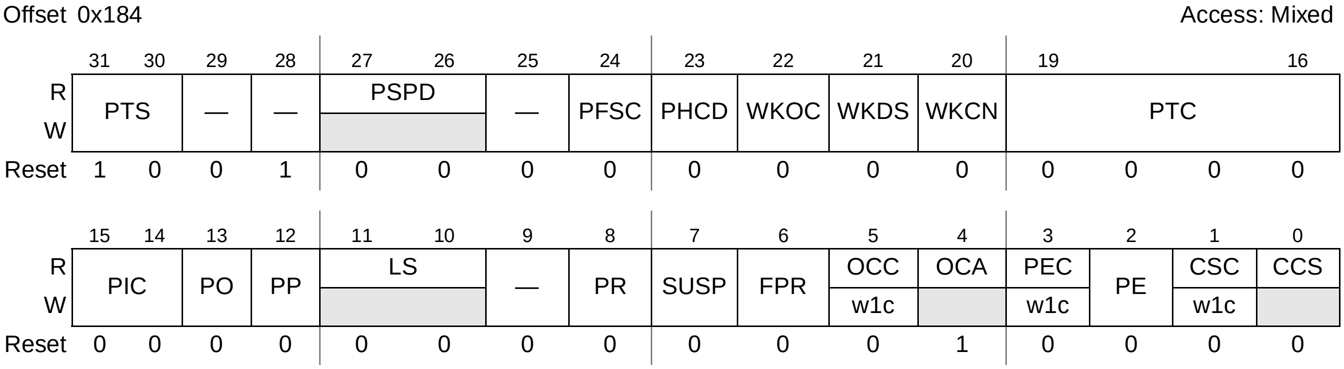
20.3.2.6, 20-22	In Figure 20-12, “Periodic Frame List Base Address (PERIODICLISTBASE),” replace all values in the “Reset” row with “All Zeros.” The updated figure is shown below:

Offset 0x154   Access: Read/Write

	31–12	11–0
R W	PERBASE	—
Reset	All zeros	

Figure 20-12 Periodic Frame List Base Address (PERIODICLISTBASE)

20.3.2.14, 20-28	In Figure 20-20, “Port Status and Control (PORTSC),” update the reset values as shown in the figure below:



Offset 0x184   Access: Mixed

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PTS		—	—	PSPD		—	PFSC	PHCD	WKOC	WKDS	WKCN	PTC			
W					▨											
Reset	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIC		PO	PP	LS		—	PR	SUSP	FPR	OCC	OCA	PEC	PE	CSC	CCS
W					▨						w1c	▨	w1c		w1c	▨
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Figure 20-20 Port Status and Control (PORTSC)

Section, Page No.	Changes
20.3.2.14, 20-28	In Figure 20-20, “Port Status and Control (PORTSC),” change bit 20 from “WLCN” to “WKCN”. In addition, in Table 20-24, “PORTSC Register Field Descriptions,” change bit 20 from “WLCN” to “WKCN”.
20.3.2.14, 20-29	In Table 20-24, “PORTSC Register Field Descriptions,” for bit 22 (WKOC), bit 21 (WKDS), and bit 20 (WKCN) change the following description from: “This bit is (OTG/host mode only) for use by an external power control circuit.” to: “This bit is used only in OTG/Host mode.”
20.3.2.14, 20-33	Remove the following text in the description of PORTSC[CCS] (bit 0) from Table 20-24, “PORTSC Register Field Descriptions:” “A one indicates that the device successfully attached and is operating in either high-speed or full-speed as indicated by the High Speed Port bit in this register.” to: “A one indicates that the device successfully attached and is operating in either high-speed or full-speed as indicated by the PSPD field in this register.” Also update the description of bit 28 to “Reserved, should be cleared.”
20.3.2.15, 20-34	In Figure 20-21, “OTG Status Control (OTGSC),” update the reset value as shown in the figure below:

Offset 0x1A4 Access: Mixed

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	—	DPIE	1msE	BSEIE	BSVIE	ASVIE	AVVIE	IDIE	—	DPIS	1msS	BSEIS	BSVIS	ASVIS	AVVIS	IDIS
W										w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	All zeros															

	15	14	13	12	11	10	9	8	7		5	4	3	2	1	0
R	—	DPS	1msT	BSE	BSV	ASV	AVV	ID	—			DP	OT	—	VC	VD
W																
Reset	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0[1]

Figure 20-21 OTG Status Control (OTGSC)

[1] Simulation reset value. The reset value of this register depends upon external conditions, such as ID value and VBUS presence.

Section, Page No. **Changes**

20.3.2.16, 20-37 In Table 20-26, “USBMODE Register Field Descriptions,” change the “Description” field values for Bit “1–0,” Name “CM” to the following text:

“Controller mode

This register can only be written once after reset. If it is necessary to switch modes, software must reset the controller by writing to USBCMD[RST] before reprogramming this register.

00	Idle (default).
01	Reserved, should be cleared.
10	Device controller.
11	Host controller.

Defaults to the idle state and needs to be initialized to the desired operating mode after reset.”

20.3.2.17, 20-37 In Figure 20-23, “Endpoint Setup Status (ENDPTSETUPSTAT),” change the access of bits “5–0, ENDPTSETUPSTAT” from “R/W” to “w1c.” Also, change the register access to “w1c.” The updated figure is shown below:

Offset 0x1AC Access: w1c

	31 … 6	5 … 0
R	—	ENDPTSETUPSTAT
W		w1c
Reset	All zeros	

Figure 20-23 Endpoint Setup Status (ENDPTSETUPSTAT)

20.3.2.23, 20-41 In Figure 20-29, “Endpoint Control 1 to 5 (ENDPTCTRLn),” change the offset value of “ENDPTCTRL3” register from “0x2_31CA” to “0x1CC.” The updated figure is shown below:

Offset 0x1C4 (ENDPTCTRL1), 0x1C8 (ENDPTCTRL2), 0x1CC (ENDPTCTRL3), 0x1D0 (ENDPTCTRL4), 0x1D4 (ENDPTCTRL5) Access: Read/Write

	31 … 24	23	22	21	20	19 18	17	16	15 … 8	7	6	5	4	3 2	1	0
R W	—	TXE	TXR	TXI	—	TXT	TXD	TXS	—	RXE	RXR	RXI	—	RXT	RXD	RXS
Reset	All zeros															

Figure 20-29 Endpoint Control 1 to 5 (ENDPTCTRL*n*)

Section, Page No.	Changes
20.3.2.26, 20-45	In Table 20-36, "PRI_CTRL Register Field Descriptions," add the following sentence to the "Description" field of bits "28–29" and "30–31:" "The highest priority is 2'h3 and the lowest priority is 2'b0." The updated rows are shown below:

Table 20-36. PRI_CTRL Register Field Descriptions

Bits	Name	Description
28–29	pri_lvl1	Priority level for priority state 1. The highest priority is 2'h3 and the lowest priority is 2'b0.
30–31	pri_lvl0	Priority level for priority state 0. The highest priority is 2'h3 and the lowest priority is 2'b0.

20.3.2.27, 20-45

In Table 20-37, "SI_CTRL Register Field Descriptions," modify the second sentence of the "Description" field of the "rd_prefetch_val" bit as follows:

"When this input is ZERO, 64 bytes are fetched, and when it is ONE 32 bytes are fetched." The updated row is shown below:

Table 20-37. SI_CTRL Register Field Descriptions

Bits	Name	Description
31	rd_prefetch_val	Selects whether 32 bytes or 64 bytes are fetched during burst read transactions at the system interface. When this input is ZERO 64 bytes are fetched, and when it is ONE 32 bytes are fetched. The setting of rd_prefetch_val must match the setting of the larger of TXPBURST and RXPBURST fields in the BURSTSIZE register. If either of these fields is 64 bytes, then rd_prefetch_val must be left cleared. Otherwise, this value should be set. 0 64-byte fetch 1 32-byte fetch

20.5.1, 20-49

Delete the following sentence appearing immediately after Figure 20-35, "Periodic Schedule Organization:"

"Split transaction interrupt, bulk and control are also managed using queue heads and queue element transfer descriptors."

Also, modify the second paragraph starting "The periodic frame list" as follows:

"The periodic frame list is a 4K-page aligned array of Frame List Link pointers. The length of the frame list is programmable. The programmability of the periodic frame list is exported to system software through the HCCPARAMS register. The length can be selected by system software as one of 8, 16, 32, 64, 128, 256, 512 or 1024 elements. Programming the size (that is, the number of elements) is accomplished by system software writing the appropriate value into Frame List Size field in the USBCMD register."

20.5.7, 20-68

Remove the following sentences from the first paragraph:

"Software must not use the FSTN feature with a host controller whose HCIVERSION register indicates a revision implementation under 0x0096. Note that FSTNs were not defined for EHCI implementations before Revision 0.96 of the EHCI Specification and their use may yield undefined results."

Add the following note at the end of the section:

Section, Page No.	Changes
	“The host controller performs only read operations to the FSTN data structure.”
20.5.7.2, 20-69	In Table 20-64, “FSTN Normal Path Pointer” modify the “Description” field of bit “0” as follows:

Table 20-64. FSTN Back Path Link Pointer

Bits	Name	Description
0	T	Terminate. 0 Link pointer is valid (that is, the host controller may use bits 31–5 as a valid memory address). This value also indicates that this FSTN is a Save-Place indicator. 1 Link pointer field is not valid (that is, the host controller must not use bits 31–5 as a valid memory address). This value also indicates that this FSTN is a Restore indicator.

20.6.1, 20-70 Delete the second sentence in the first paragraph:

“After a hardware reset, only the operational registers are at their default values.”

20.6.13, 20-119 In the third bullet, delete the following text:

“Note that an EHCI host controller implementation may optionally allow port testing with RS set. However, all host controllers must support port testing with RS cleared and HCH set.”

and append the following sentence:

“In Device mode, the Test Mode starts only if Run/Stop bit is set to 1. In Host mode, the Test Mode starts regardless of Run/Stop bit.”

20.6.2, 20-71 Add the following sentence at the end of the paragraph:

“The Configured Flag and Port Power Control bits are always 1’b1 in Host Mode. The PPE always follows the state of Port Power (PP) bit that is, if PP is 0, PPE will be 0 and if PP is 1, PPE will be 1.”

20.6.4.1, 20-73 In Table 20-65, “Behavior During Wake-Up Events” delete all instances of “WKDSCNNT_E” from the “Port Status and Signaling Type” column. In the same column, replace the following:

“WKCNNT_E” with “PORTSC[WKCN]”

“WKOC_E” with “PORTSC[WKOC]”

20.6.8.1, 20-80 In the second list bullet, seventh paragraph, modify the sentence beginning from “Note, that the host controller is not required to update...”

to

“Note, that the host controller does not update the iTD field Transaction *n* Length in this error scenario.”

Section, Page No. **Changes**

20.6.12.2.1, 20-96 In Figure 20-54, “General Structure of EHCI Periodic Schedule Utilizing Interrupt Spreading” change the notations of the nodes as per the corresponding diagram in “Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus, Revision 1.0.” The updated figure is shown below:

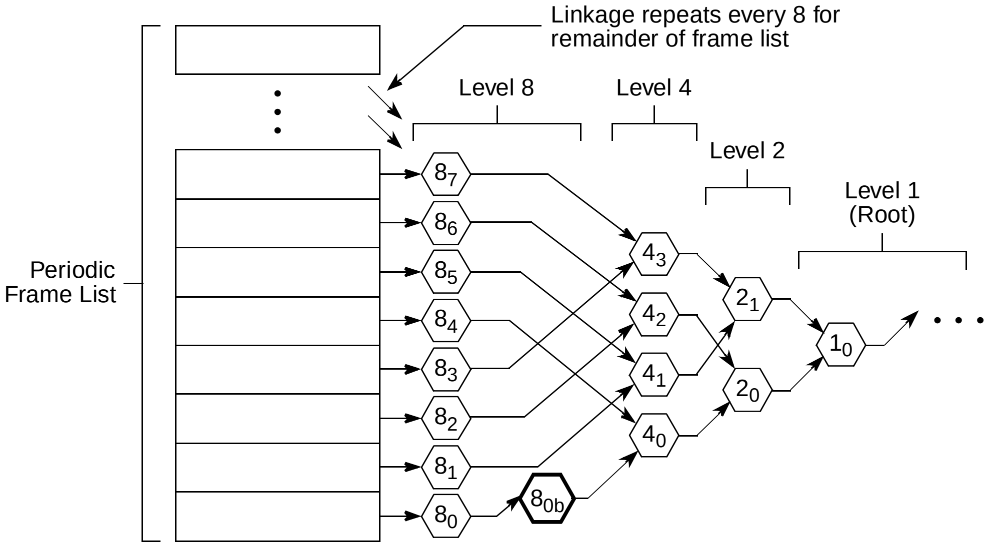


Figure 20-54 General Structure of EHCI Periodic Schedule Utilizing Interrupt Spreading

20.6.12.2.2, 20-98 In Figure 20-55, “Example Host Controller Traversal of Recovery Path via FSTNs” change the notations of the nodes as per the corresponding diagram in “Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus, Revision 1.0.” The updated figure is shown below:

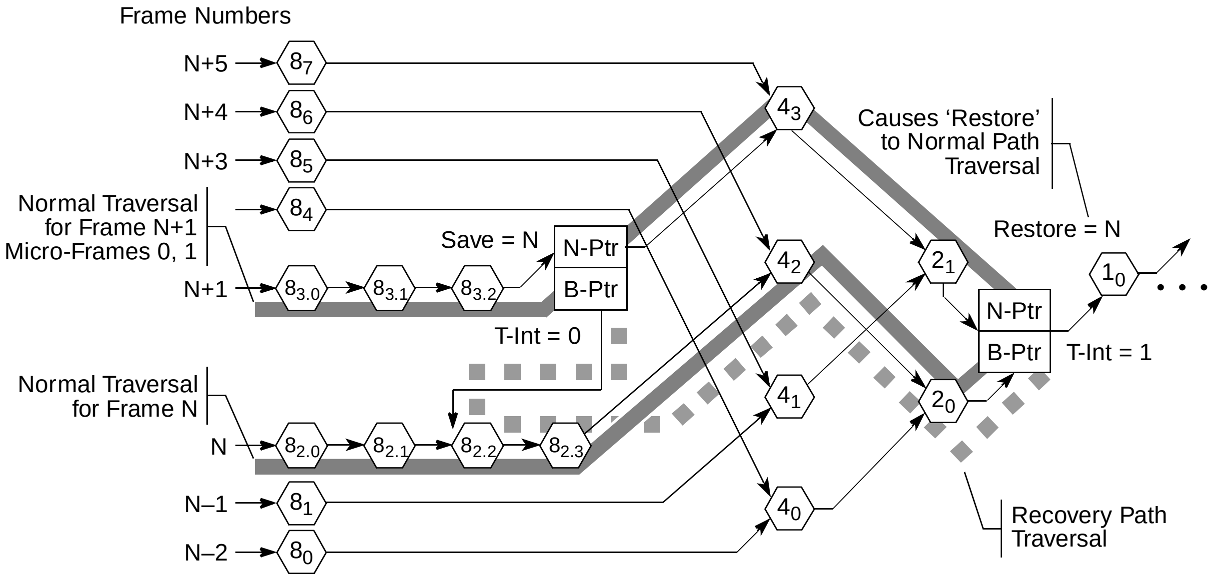


Figure 20-55 Example Host Controller Traversal of Recovery Path via FSTNs

20.6.12.2.5, 20-101 Modify Figure 20-56, “Split Transaction State Machine for Interrupt” as per the corresponding figure given in “Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus, Revision 1.0.” The updated figure is shown below:

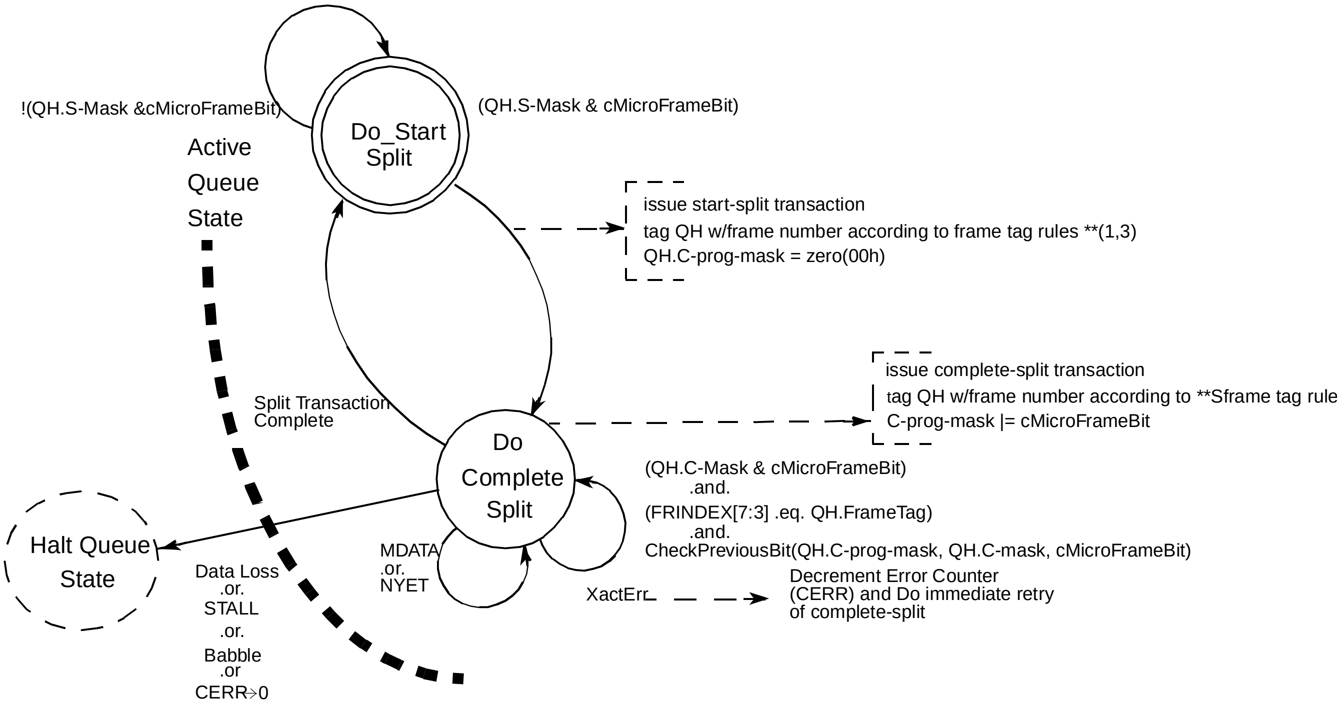


Figure 20-56 Split Transaction State Machine for Interrupt

Section, Page No. **Changes**

20.6.12.3.3, 20-112 Modify Figure 20-59, “Split Transaction State Machine for Isochronous” as per the corresponding figure given in “Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus, Revision 1.0.” The updated figure is shown below:

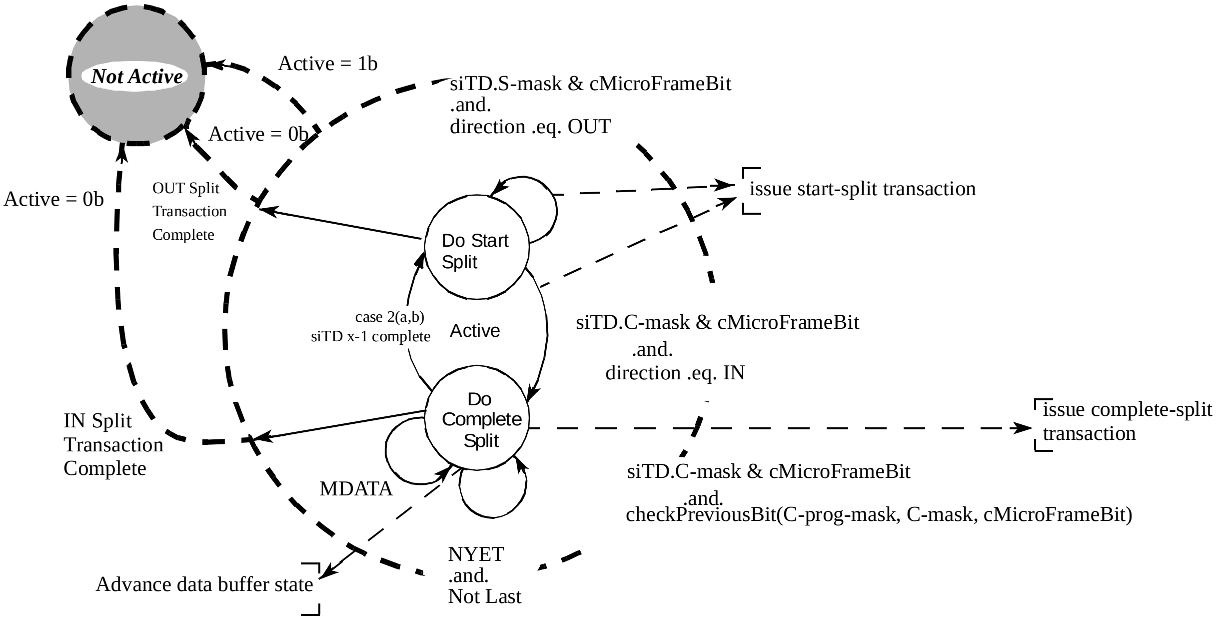


Figure 20-59 Split Transaction State Machine for Isochronous

20.6.12.3.4, 20-114 Update the following sentence in the seventh paragraph from:

“The preferred method is to detect when T-Count decrements to zero as a result of a start-split bus transaction. Equivalently, the host controller can detect when Total Bytes to Transfer decrements to zero. Either implementation must ensure that if the initial condition is Total Bytes to Transfer is equal to zero and T-count is equal to a one, the host controller issues a single start-split, with a zero-length data payload.”

to:

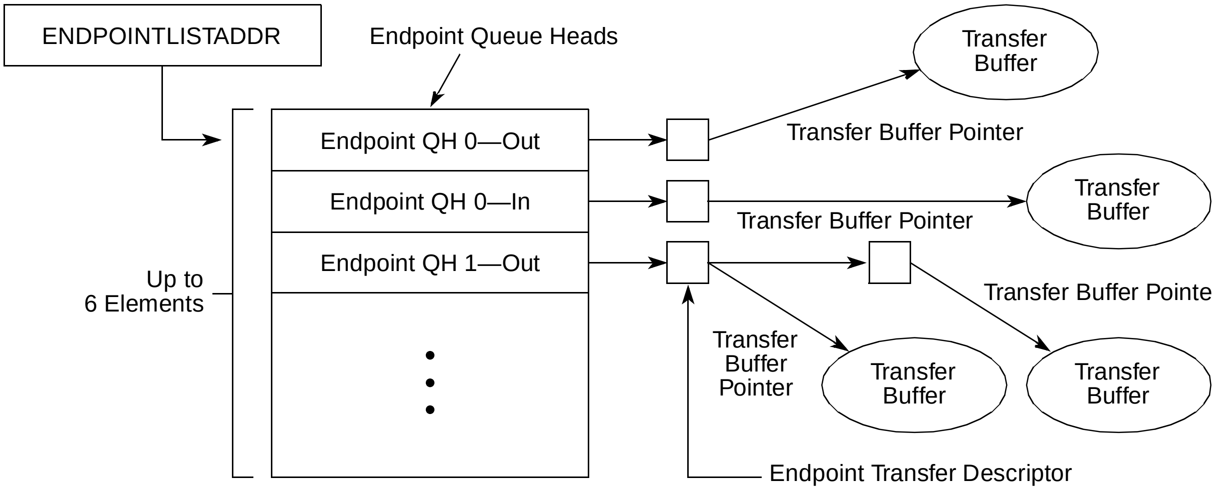
“Setting the Active bit to zero depends on siTD.TP being 00 or 11, and siTD.Total Bytes decrements to 0.”

20.7.2, 20-129 Add the following sentence immediately after Table 20-80, “dTD Token:”

“Table 20-81 to Table 20-83 describe the buffer pointer page *n* fields.”

20.8.3.2.2, 20-136 Modify “data toggle” to “data toggle state bit” in the first sentence of the third paragraph. The sentence now reads:

“In normal operation, the USB_DR checks the DATA0/DATA1 bit against the data toggle state bit to determine if the packet is valid.”

Section, Page No.	Changes
20.8.3.3, 20-137	Change the heading name from “Device For Packet Transfers” to “Device Operational Model For Packet Transfers.”
20.8.3.3.1, 20-137	Modify the following sentence in the second para from: “This FIFO is split into virtual channels so that the leading data can be stored for any endpoint up to the maximum number of endpoints configured at device synthesis time.” to “This FIFO is split into virtual channels so that the leading data can be stored for any endpoint.”
20.8.3.5.2, 20-140	Replace the first sentence of the first paragraph from: “Following the setup phase, the DCD must create a device transfer descriptor for the data phase and prime the transfer.” to: “If the control transfer requires a data stage following the setup phase, the DCD must create a device transfer descriptor for the data phase and prime the transfer.”
20.8.4, 20-144	In Figure 20-64, “Endpoint Queue Head Diagram,” replace “Up to 32 Elements” with “Up to 6 Elements.” The updated figure is shown below:  **Figure 20-64 Endpoint Queue Head Diagram**
20.8.5.3, 20-147	Modify the steps (7) and (8) under “Link list is not empty” case as follows: “7) If status bit read in (4) is ‘1’ DONE. 8) If status bit read in (4) is ‘0’ then Go to Case 1: Step 1.”
20.9.5.1, 20-155	Modify the last sentence as follows: “That is, 60 MHz transceiver clock for 8-bit physical interfaces and full-speed serial interfaces or 30 MHz transceiver clock.”
20.10, 20-156	Remove entire section 20.10, “Timing Diagrams.”

Section, Page No.	Changes
23.2.3.3, 23-6	Remove the following sentence appearing after Figure 23-3, “Multiple-Master Configuration”: “The maximum sustained data rate that the SPI supports is input clock/50.”
23.4.1.2, 23-12	Change the description of LT (bit 17) in Table 23-5, “SPIE Field Descriptions” from: “Last character was transmitted. The last character is transmitted and new data can be written to SPID for further transmission.” to: “The last character of the frame was completely transferred. This bit is set only if the transmitted character was the last character of the frame (if SPCOM[LST] is set). New data can be written to SPITD is indicated by bit NF.”

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or
+1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064
Japan
0120 191014 or
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
1-800 441-2447 or
+1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale, the Freescale logo, CodeWarrior, ColdFire, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc. Reg. U.S. Pat. & Tm. Off. CoreNet, QorIQ, QUICC Engine, and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2011 Freescale Semiconductor, Inc.

Document Number: MPC8379ERMAD
Rev. 1.2
04/2011