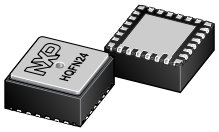


UM11227

NTM88 family of tire pressure monitor sensors

Rev. 7 — 29 March 2021 User manual

Document information

Information	Content
Keywords	NTM88, features, architecture, programming model, 8-bit microcontroller (MCU), pressure sensor, accelerometer, programmable RF transmitter and flexible LF receiver
Abstract	This user manual describes the features, architecture, and programming model of the NTM88 family of devices.

Revision history

Document ID	Release date	Description
UM11227 v.7	20210329	• Global changes as follows: – Performed minor grammatical, content and tyopgraphic revisions throughout. – Revised "SRS" to "SIMRS" in six locations. • Inserted new document information table on the first page of the data sheet. • Section 1, relocated the revision history to the front of the document to comply with NXP content guidelines for user manuals. • Section 3, revised as follows: – Revised "Pressure range: 90 kPa to 930 kPa" to "Optional pressure ranges". – Removed "Optional accelerometer range: See Section 4." – Revised "Slave SPI to support..." to "Client SPI to support..." • Section 4, Table 1, revised the "Type number" from "NTM88Hxx5" to "NTM88Hxxx" and "NTM88Jxxx". • Section 4.5, revised "At address $FC00, 1024 bytes..." to "At address $FC00, 512 bytes..." • Section 5.1 revised as follows: – Step 1: revised the content. – Table 3: revised the part number from "NTM88H05xT1" to "NTM88xxxxT1", the pressure value from "H" to "y" and updated footnote 3. • Section 9.1, revised the first paragraph adding a statement to visit the NXP website for user guides, application notes and evaluation hardware collateral. • Section 10.19.1.3, Figure 47, revised the title. • Section 10.19.2.4, revised the note below Table 155. • Section 11, Figure 60, revised the title.
UM11227 v.6	20200424	• Section 3, revised second bullet for "Optional accelerometer ranges" and added a footnote. • Section 5.1, revised as follows: – Table 3, revised tables notes 1, 3, 4, and 5. – Table 4, revised table note 1 • Section 10.1.1, deleted rows for register map addresses $E7E0 through $E7FF. • Section 10.2.2, deleted the last paragraph starting with "The LF, SMI, and ADU user...." • Section 10.3, Table 17, revised vector priority 8 removing "reserved" and providing values. • Section 10.12.1, revised as follows: – Moved the figure titled "KBI block diagram" to Section 10.12.2.1 prior to Table 36. – Moved the figure titled "External interrupt logic" to Section 10.12.2.4 prior to Table 42. – Figure 14, revised image. – Table 23, in the "Pull enable" row, revised the "x" in the "KBI pin enable" column to "0". • Section 10.12.1.1, revised as follows: – Removed second paragraph starting with "PTA[4:0] pins are shared with on-chip peripheral functions." – Removed redundant Figures titled "General purpose I/O block diagram" and "General purpose I/O logic". – Removed redundant Table titled "Truth table for pullup and pulldown resistors". – Table 37, revised the "Description" for "KBACK". • Section 10.12.2.4, Table 43, revised the "Description" for "IRQACK". • Section 10.14.1.1, Table 63, revised the "Description" for "WUFACK" and "PRFACK". • Section 10.15.17.5, Table 84, revised the "Description" for "LFIAK". • Section 10.16.11.8, Table 118, revised the "Description" for "RFIAK". • Section 10.19.2.1, Table 147, revised the "Description" for "SMIFAK". • Section 10.19.2.3, Table 151, revised the description for 1:0, FILT[1:0]

Revision history...*continued*

Document ID	Release date	Description
		UM11227 v.6 (Continued) • Section 10.23.3.1, Table 175, revised and unmerged the Bit 6 cell for "R" and "W", inserting "0" in Bit 6 for "R" and revised the "Description" for "RTIACK" in Table 176. • Section 10.23.3.2, Table 177, revised and unmerged the Bit 6 cell for "R" and "W", inserting "0" in Bit 6 for "R" and revised the "Description" for "LVDACK" in Table 178. • Section 10.23.3.3, Table 179, revised and unmerged the Bit 2 cell for "R" and "W", inserting "0" in Bit 2 for "R" and revised the "Description" for "PPDACK" in Table 180. • Section 10.25, inserted a new first bullet, revised the second bullet, and inserted a new bullet before the last bullet. • Section 10.26, Added new first paragraph. • Section 10.26.6, revised the first sentence and the figure title for Figure 59. • Section 11, revised the paragaph starting with "A gel is used to provide media protection...", adding two new sentences at the end of the paragraph.
UM11227 v.5	20200124	• Section 10.19, revised entire section.
UM11227 v.4	20191004	• Section 3: Revised third bullet under "Transducer measurement interfaces" from "12-bit compensated..." to "8-bit compensated...." • Section 4.1: Revised the first bullet adding "For devices programmed by NXP with an embedded firmware..." and added new paragraph beginning with "Prototype samples...." • Section 4.4, revised the second paragraph. • Section 7.3, Figure 4: Revised Figure 4. • Section 10.1.1, Table 15, revised rows $1860, $1861 and $FD66:$FDFA. • Section 10.8.5.2, Table 20: Removed "Normal Temperature Restart" row. • Section 10.8.5.4: Removed section titled "Temperature restart" that followed Section 10.8.5.4.14. • Section 10.9.1, Table 21, revised as follows: – Start Address $FC00, revised the "End Address" from "$FD65" to "$FD3F" and updated the "Block description". – Start Address $FD40, revised "Start Address "from $FD66" to "$FD40" and updated the "Block description". – Start Address $FFC0, revised "End address" from "$FFDB" to "$FFDF" and updated the "Block description". – Start Address $FFDC, removed entire row. • Section 10.16.11.1, Table 104: Added new table. • Section 10.16.11.3, Table 108: Revised the description for "4:0, PWR[4:0]". • Section 10.16.11.9, Table 123: Revised the description for "15:3, AFREQ[12:0]". • Section 10.16.11.11: Revised "RFCR8" to "EPR" in three locations. • Section 10.16.11.12: Revised "RFCR9" to "RFPRECHARGE" in three locations. • Section 10.19.2.3, Table 151, added "Recommend adjusting to 500 Hz or higher when either or both ISD[3:0] / SP[3:0] are configured for times < 1024 ms." to the description for FILT[1:0] for the case of "0 0 = 250 Hz". • Section 11, Figure 60: revised the image.
UM11227 v.3	20190822	• Section 2, revised the general description paragraph. • Section 3, revised as follows: – Revised the supporting bulleted items of the bullet "Transducer measurement interfaces with low-power AFE." – Revised "16k bytes flash memory" to "16 kB flash memory" below "8-bit S08 compact instruction set controller." • Section 4, Table 1: revised the description.

Revision history...*continued*

Document ID	Release date	Description
		UM11227 v.3 Modifications (Continued.) • Section 4.1, replaced table titled "CodeF - tolerance and firmware configuration encoding" with a paragraph and bullets. • Section 4.2, replaced table titled "CodeH - hardware configuration encoding" with a paragraph and bullets. • Section 4.3: revised as follows: – Table 2, bits 7 through 0, removed table 2 reference for CODEF, removed table 3 reference for CODEH and replaced references with "Consult the appropriate NTM88 product data sheet for a description." – Revised "ID27 — 0 to identify NTM88 family" to "ID27 — 1 to identify NTM88 family." • Section 5.1, Table 3: Revised footnote 5. • Section 6: revised the first paragraph. • Section 6, Figure 2, revised the figure caption. • Section 7, added introductory paragraph. • Section 7.1: Revised the image in Figure 3. • Section 7.2, Table 5: Revised the symbols for pins 1 through 6 from "NC" to "n.c." to support changes made to the image in Figure 3. • Section 10.1.1, Table 15, revised as follows – Address $0008, revised all entries to "reserved." – Address $1809, revised Bit1 to "reserved." – Address $180C, revised Bit3, Bit2, and Bit0 to "reserved." – Address $FDFF, revised Bit7 to "ID31," Bit6 to "ID30," Bit5 to "ID29," and Bit4 to "ID28." • Removed the section titled "Port input filter enable register (PORTIFE)" that followed Section 10.12.1.7. • Section 10.16.9, Figure 38: revised the figure. • Section 10.16.11.12, revised as follows: – Table 128, revised "Reset ($00)" to "Reset ($40) and the "TIMEOUT0" bit from "0" to "1". – Table 129, revised the description for 7:6. • Section 10.19, Figure 44: revised the figure. • Section 10.19.1: revised as follows: – Revised the last sentence before Table 145. – Table 145: Added "Stop4 entry not recommended." to the "Comments" for "Direct" • Section 10.19.1.2: Revised 5th paragraph, 2nd sentence. • Section 10.23.3.2, Table 177, removed "BGBDS" from Bit1 • Section 10.23.3.2, Table 178, removed "BGBDS" row from table. • Section 10.23.3.4, Table 181, removed "HVWF" from Bit3, "HVWACK" from Bit2, and "HVWE" from Bit 0. • Section 10.23.3.4, Table 182, removed rows for "HVWF", "HVWACK," and "HVWE." • Section 11: Added new paragraph before Figure 60.
UM11227 v.2	20190516	• The format of this document has been redesigned to comply with the identity guidelines of NXP Semiconductors. • Legal texts have been adapted to the new company name where appropriate. • Revised document number from "NTM88RM" to "UM11227". • UM11227 v.2 supercedes NTM88RM v.1.
NTM88RM v.1	20181214	• Initial release

1 Introduction

1.1 Purpose

This user manual describes the features, architecture, and programming model of the NTM88 family of devices.

1.2 Audience

This document is primarily for system architects and software application developers who are using or considering the use of the NTM88 in a system.

2 General description

The NTM88 is a small (4 mm x 4 mm x 1.98 mm), fully integrated tire pressure monitoring sensor (TPMS). It also provides low transmitting power consumption, large customer memory size, and a choice of either dual- or single-axis accelerometer architecture. The NTM88 TPMS solution integrates an 8-bit microcontroller (MCU), pressure sensor, accelerometer, and RF transmitter.

3 Features and benefits

- Optional pressure ranges[1]
- Optional single- or dual-axis accelerometer ranges[1]
- Transducer measurement interfaces with low-power AFE:
  - 10-bit compensated pressure sense element
  - 10-bit compensated accelerometers
  - 8-bit compensated internal device temperature measurement
  - 8-bit compensated internal device voltage measurement
  - Two I/O pins can be used for external signals
- 8-bit S08 compact instruction set controller:
  - 64 bytes low-power “always on“ NVM parameter registers
  - 512 bytes SRAM
  - 16 kB flash memory (512 bytes reserved for NXP coefficients)
  - Family of NXP firmware libraries available via royalty-free license
- Programmable RF transmitter
  - Characterized for RF carrier typical of 315 MHz or 434 MHz
  - Characterized for FSK in ~3 kHz increments or OOK modulation
  - Characterized for baud rate examples of 9.6 kbp/s, 19.2 kbp/s, and 38.4 kbp/s
- Flexible 125 kHz LF receiver:
  - Capability for ASK or OOK demodulation
  - Automated Manchester decoding
- Two channel timer / pulse-width module
- Client SPI to support host access to internal peripherals, registers, and memory
- Seven GPIOs with programmable multiplexing to support software development, external ADC input, timer, SPI, and wake-up
- Qualified in compliance with AEC-Q100, Rev. H

1 Consult NXP sales for details or specific requests.

- Long battery service life
- Temperature sensor
- Voltage reference measured by ADC10
- Six-channel, 10-bit analog-to-digital converter (ADC10) with two external I/O inputs
- Internal 315-/434-M Hz RF transmitter
  - External crystal oscillator
  - PLL-based output with fractional-n divider
  - OOK and FSK modulation capability
  - Programmable data rate generator
  - Manchester, Bi-Phase, or NRZ data encoding
  - 256-bit RF data buffer variable length interrupt
  - Direct access to RF transmitter from MCU for unique formats
  - Low-power consumption
- Differential input LF detector/decoder on independent signal pins
- Real-time Interrupt driven by LFO with intervals of 2, 4, 8, 16, 32, 64, or 128 ms
- Free-running counter, low-power, wake up timer and periodic reset driven by LFO
- Watchdog timeout with selectable times and clock sources
- Two-channel general-purpose timer/PWM module (TPM1)
- Internal oscillators
  - MCU bus clock of 0.5, 1, 2, and 4 MHz (1, 2, 4, and 8 MHz HFO)
  - Low frequency, low-power time clock (LFO) with 1 ms period
  - Medium frequency, controller clock (MFO) of 8 µs period
- Low-voltage detection

4 Configuration options

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
NTM88Hxxx NTM88Jxxx	HQFN24	Plastic thermal enhanced quad flat package; no leads, 0.1 dimple wettable flank; 24 terminals; 0.5 mm pitch, 4 mm x 4 mm x 1.98 mm body	SOT1931-1(D)

4.1 Electronic encoding - "CodeF"

Consult the appropriate NTM88 product data sheet for a description of the CodeF traceability which allows the user to extract:

- For devices programmed by NXP with an embedded firmware, configuration values holding the firmware library used for final test
- Accelerometer variant type

Prototype samples may be configured and delivered with the firmware remaining in the flash memory upon special request. The series production process will erase the firmware from flash memory to facilitate customers choice of the firmware routines, while excluding specific firmware routines the application software does not require. Consult the appropriate NTM88 firmware user guide for a description of the available firmware routines, either as firmware in flash, or as library releases.

4.2 Electronic encoding - "CodeH"

Consult the appropriate NTM88 product data sheet for a description of the CodeH traceability which allows users to extract:

- configuration values holding the assembly revision
- final test pressure
- accelerometer calibrations

4.3 Device identification

The bytes assigned to identify the device and its options are described below. This data can be read using the TPMS_READ_ID routine.

Table 2. Device ID coding summary

ID Address	Register Name	BIT							
		7	6	5	4	3	2	1	0
00	CODEF	Consult the appropriate NTM88 product data sheet for a description.							
01	CODEH	Consult the appropriate NTM88 product data sheet for a description.							
02	CODE2	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
03	CODE3	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
04	CODE4	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
05	CODE5	ID31	ID30	ID29	ID28	ID27	ID26	ID25	ID24

ID13:0 — Device ID within each assembly lot - 16k devices in each lot

ID26:14 — Lower 13 bits of assembly lot ID - 32k lots

ID27 — 1 to identify NTM88 family

ID28:29 — Upper 2 bits of assembly lot ID

ID30 — 0x1 to identify sub-con B, 0x0 to identify sub-con A

ID31 — 0x1 to identify NXP as device supplier

Note: *Prior to erasing the flash memory, users are advised to first copy the contents of the CODEF through CODE5 data into a secure and retrievable database when using, for example, a custom gang programmer in lieu of the CodeWarrior IDE tool. The contents of CODEF through CODE5 are unique to each part number, configuration of pressure and accelerometer ranges, and serial numbers, and must be replaced as part of the user flash programming processes.*

4.4 Definition of signal ranges

Each measured parameter (pressure, voltage, temperature, acceleration) results from an ADC10 conversion of an analog signal. This ADC10 result may then be passed by the firmware to the application software as either the raw ADC10 result or further compensated and scaled for an output between one and the maximum digital value minus one. The minimum digital value of zero and the maximum digital value are reserved as error codes.

The signal ranges and their significant data points are shown in Figure 1. In this definition, the signal source would normally output a signal between S_{INLO} and S_{INHI}. Due

to process, temperature, and voltage variations, this signal may increase its range to S_{INMIN} to S_{INMAX}. In the example case of 10-bit raw conversions and 9-bit compensation, the signal is between the supply rails, so that the ADC10 converts it to a range of digital numbers between 0 and 1023. These digital numbers have corresponding D_{INMIN}, D_{INLO}, D_{INHI}, D_{INMAX} values. The ADC10 digital value is taken by the firmware and compensated and scaled to give the required output code range.

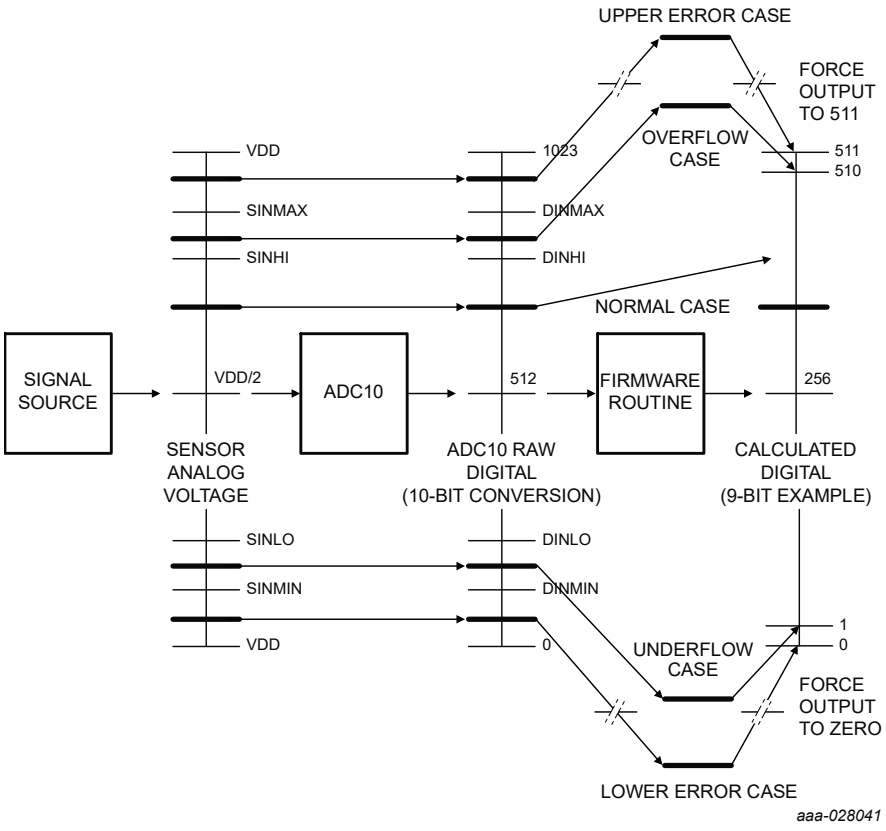


Figure 1. Measurement signal range definitions

Digital input values below D_{INMIN} and above D_{INMAX} are immediately flagged as being out of range and generate error bits and the output is forced to the 0 value.

Digital values below D_{INLO} (but above D_{INMIN}) or above D_{INHI} (but not D_{INMAX}) will most likely cause an output that would be less than 1 or greater than 510, respectively. These cases are considered underflow or overflow, respectively. Underflow results will be forced to a value of 1. Overflow results will be forced to a value of 510.

Digital values between D_{INLO} and D_{INHI} will normally produce an output between 1 to 510 (for a 9-bit result). In some isolated cases due to compensation calculations and rounding, the result may be less than 1 or greater than 510, in which case the underflow and overflow rule mentioned above is used.

4.5 Memory resource usage

At address $FC00, 512 bytes are protected from erasure, containing the sensitivity and offset coefficients for the transducers and clocks.

The firmware uses no specific bytes of the RAM but will cause additional stacking of temporary values.

The firmware uses 2 bytes ($008E and $008F) of the Parameter Registers for global flags for all routines.

5 Marking

5.1 Exterior markings

The marking[2] on the NTM88 family contain three lines of text, described as follows:

1. Line 1 identifies the location of pin 1 and, when appropriate, shows the corporate logo
2. Line 2 identifies part marking information, see Table 3 for details on the NTM88 markings.
3. Line 3 is the trace code. See Table 4 for trace code definitions.

Table 3. Example Exterior Marking

Marking						
Part Number	Company[1]	Family[2]	Pressure[3]	Accelerometer[4]		Mechanical[5]
NTM88xxxxT1	N	8	y	a	a	x

[1] Company column: N = qualified.
[2] Family column: Always "8".
[3] Pressure column: Where "y" is a letter representing the pressure configuration.
[4] Accelerometer columns: Where "a a" are two letters representing the accelerometer configuration.
[5] Mechanical column: Where "x" is a letter representing the mechanical configuration.

Table 4. Trace code definitions

Trace code	Definition
A	Assembly site[1]
L	Wafer lot
YW	Year and work week
Z	Assembly lot split[2]

[1] "X" for site #1; additional letters for other assembly sites as needed.
[2] “Z” can be up to two characters "ZZ" when the number of subassembly lots > 26

2 Subject to change by NXP without notice.

6 Block diagram

Figure 2 presents the device's main blocks and their signal interactions. Power management controls and bus control signals are not shown in this block diagram for clarity.

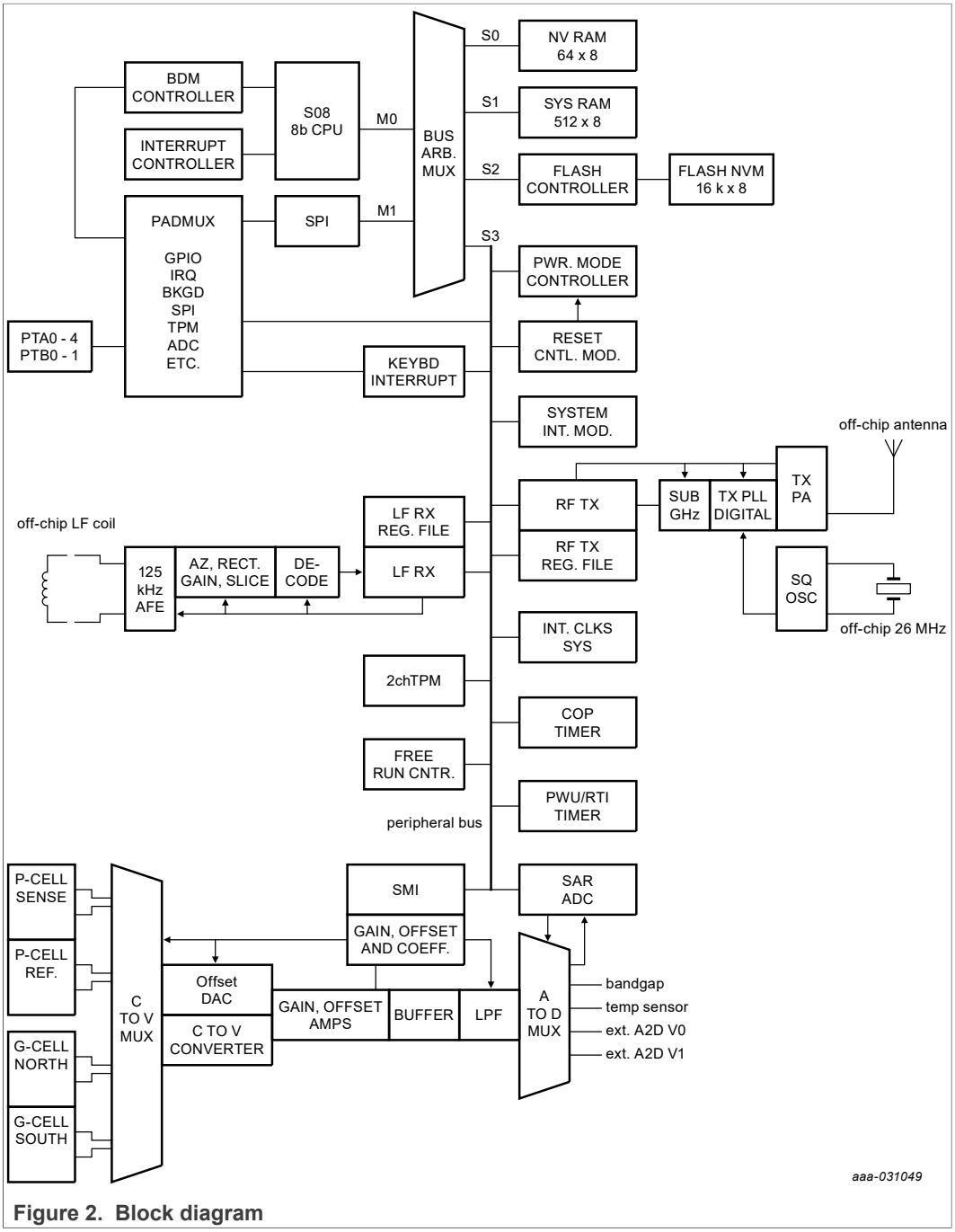


Figure 2. Block diagram

7 Pinning information

This section describes the pin layout and general function of each pin.

7.1 Pinout

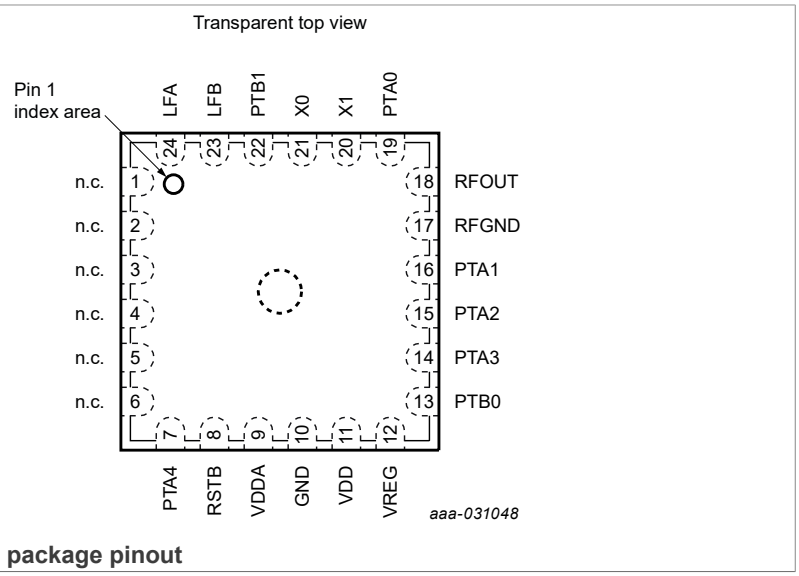


Figure 3. NTM88 QFN package pinout

7.2 Pin description

Table 5. Pin description

Symbol	Pin	Function	Description
n.c.	1	—	Do not connect electrical signals to this pin; solder joint only.
n.c.	2	—	Do not connect electrical signals to this pin; solder joint only.
n.c.	3	—	Do not connect electrical signals to this pin; solder joint only.
n.c.	4	—	Do not connect electrical signals to this pin; solder joint only.
n.c.	5	—	Do not connect electrical signals to this pin; solder joint only.
n.c.	6	—	Do not connect electrical signals to this pin; solder joint only.
PTA4	7	PTA4 / BKGD	PTA4 Pin - The PTA4 pin places the device in the BACKGROUND DEBUG mode (BDM) to evaluate MCU code and transfer data to/from the internal memory. If the BKGD/PTA4 pin is held low when the device comes out of a power-on-reset (POR), the device switches into the ACTIVE BACKGROUND DEBUG mode (BDM). The BKGD/PTA4 pin has an internal pullup device or can be connected to VDD in the application, unless there is a need to enter BDM operation after the device as been soldered into the PWB. If in-circuit BDM is desired, the BKGD/PTA4 pin should be connected to VDD through a resistor (~10 kΩ or greater) which can be over-driven by an external signal. This resistor reduces the possibility of inadvertently activating the debug mode in the application due to an EMC event. When the application programs port A to GPIOs, PTA4 becomes output-only.

Table 5. Pin description*...continued*

Symbol	Pin	Function	Description
RST_B	8	Reset / V_{PP} programming voltage	The RST_B pin is used for test and establishing the BDM condition and providing the programming voltage source to the internal FLASH memory. This pin can also be used to direct to the MCU to the reset vector. The RST_B pin has an internal pullup device and can be connected to VDD in the application unless there is a need to enter BDM operation after the device as been soldered to the PWB. If in-circuit BDM is desired, the RST_B pin can be left unconnected; but should be connected to VDD through a low impedance resistor (<10 kΩ) which can be over-driven by an external signal. This low impedance resistor reduces the possibility of getting into the debug mode in the application due to an EMC event. Activation of the external reset function occurs when the voltage on the RST_B pin goes below 0.3 × V_{DD} for at least 100 ns before rising above 0.7 × V_{DD}.
VDDA	9	Analog supply	The analog circuits operate from a single power supply connected to the unit through the VDDA pin. VDDA is the positive supply and GND is the ground. The conductors to the power supply should be connected to the VDDA and GND pins and locally decoupled. Care should be taken to reduce measurement signal noise by separating the VDD, GND, VDDA, and RFGND pins using a “star” connection such that each metal trace does not share any load currents with other external devices.
GND	10	Digital and analog ground	The digital circuits operate from a single power supply connected to the unit through the VDD and GND pins. GND is the ground. Care should be taken to reduce measurement signal noise by separating the GND and RFGND pins using a “star” connection such that each metal trace does not share any load currents with other external devices.
VDD	11	Digital supply	The digital circuits operate from a single power supply connected to the unit through the VDD and GND pins. VDD is the positive supply. The conductors to the power supply should be connected to the VDD and GND pins and locally decoupled.
VREG	12	1.8 V regulation	The internal regulator for the RF analog circuits requires an external stabilization capacitor to GND.
PTB0	13	PTB0 / TPMCH0 / AD3	The PTB[0] pin is a general-purpose I/O pin. This pin can be configured as a nominal bidirectional I/O pin with programmable pullup devices. User software must configure the general-purpose I/O pin (PTB[1:0]) so that they do not result in “floating” inputs. PTB0 can be mapped to TPM channel 0, or to ADC channel 3.
PTA3	14	PTA3 / KBI3 / MOSI	The PTA[3] pin is a general-purpose I/O pin. The pulldown devices can only be activated if the wake-up interrupt capability is enabled. User software must configure the general-purpose I/O pins so that they do not result in “floating” inputs. PTA[3] maps to keyboard interrupt function bit [3]. When SPI is enabled, PTA[3] serves as MOSI.
PTA2	15	PTA2 / KBI2 / MISO	The PTA[2] pin is a general-purpose I/O pin. The pulldown devices can only be activated if the wake-up interrupt capability is enabled. User software must configure the general-purpose I/O pins so that they do not result in “floating” inputs. PTA[2] maps to keyboard interrupt function bit [2]. When SPI is enabled, PTA[2] serves as MISO.

Table 5. Pin description*...continued*

Symbol	Pin	Function	Description
PTA1	16	PTA1 / KBI1 / SCLK	The PTA[1] pin is a general-purpose I/O pin. The pulldown devices can only be activated if the wake-up interrupt capability is enabled. User software must configure the general-purpose I/O pins so that they do not result in “floating” inputs. PTA[1] maps to keyboard interrupt function bit [1]. When SPI is enabled, PTA[1] serves as SCLK
RFGND	17	RF ground	Power in the RF output amplifier is returned to the supply through the RFGND pin. This conductor should be connected to the power supply using a “star” connection such that each metal trace does not share any load currents with other supply pins.
RFOUT	18	RF output	The RFOUT pin is the RF energy data supplied by the unit to an external antenna.
PTA0	19	PTA0 / KBI0 / SS_B / IRQ	The PTA[0] pin is a general-purpose I/O pin. PTA[0] can be configured as a normal bidirectional I/O pin with programmable pullup or pulldown devices and/or wake-up interrupt capability. PTA[0] can be configured for external interrupt (IRQ). The pulldown devices can only be activated if the wake-up interrupt capability is enabled. User software must configure the general-purpose I/O pins so that they do not result in “floating” inputs. PTA[0] maps to keyboard interrupt function bit [0]. When SPI is enabled, PTA0 serves as SS_B.
X1	20	RF crystal input	The X1 pin is for an external 26 MHz crystal to be used by the internal PLL for creating the carrier frequencies and data rates for the RF pin.
X0	21	RF crystal output	The X0 pin is for an external 26 MHz crystal to be used by the internal PLL for creating the carrier frequencies and data rates for the RF pin.
PTB1	22	PTB1 / TPMCH1 / AD4	The PTB[1] pin is a general-purpose I/O pin. This pin can be configured as a nominal bidirectional I/O pin with programmable pullup devices. User software must configure the general-purpose I/O pins (PTB[1:0]) so that they do not result in “floating” inputs. PTB1 can be mapped to TPM channel 1, or to ADC channel 4.
LFB	23	LF input '-'	The LF[A:B] pins can be used by the LF receiver (LFR) as one differential input channel for sensing low-level signals from an external low frequency (LF) coil. The external LF coil should be connected between the LF[A] and the LF[B] pins. Signaling into the LFR pins can place the unit into various diagnostic or operational modes. The LFR is comprised of the detector and the decoder. Each LF[A:B] pin always has an impedance of approximately 500 kΩ to GND due to the LFR input circuitry. The LFA/LFB pins are used by the LFR when the LFEN control bit is set and are not functional when the LFEN control bit is clear.
LFA	24	LF input '+'	The LF[A:B] pins can be used by the LF receiver (LFR) as one differential input channel for sensing low-level signals from an external low frequency (LF) coil. The external LF coil should be connected between the LF[A] and the LF[B] pins. Signaling into the LFR pins can place the unit into various diagnostic or operational modes. The LFR is comprised of the detector and the decoder. Each LF[A:B] pin always has an impedance of approximately 500 kΩ to GND due to the LFR input circuitry. The LFA/LFB pins are used by the LFR when the LFEN control bit is set and are not functional when the LFEN control bit is clear.

7.3 Orientation

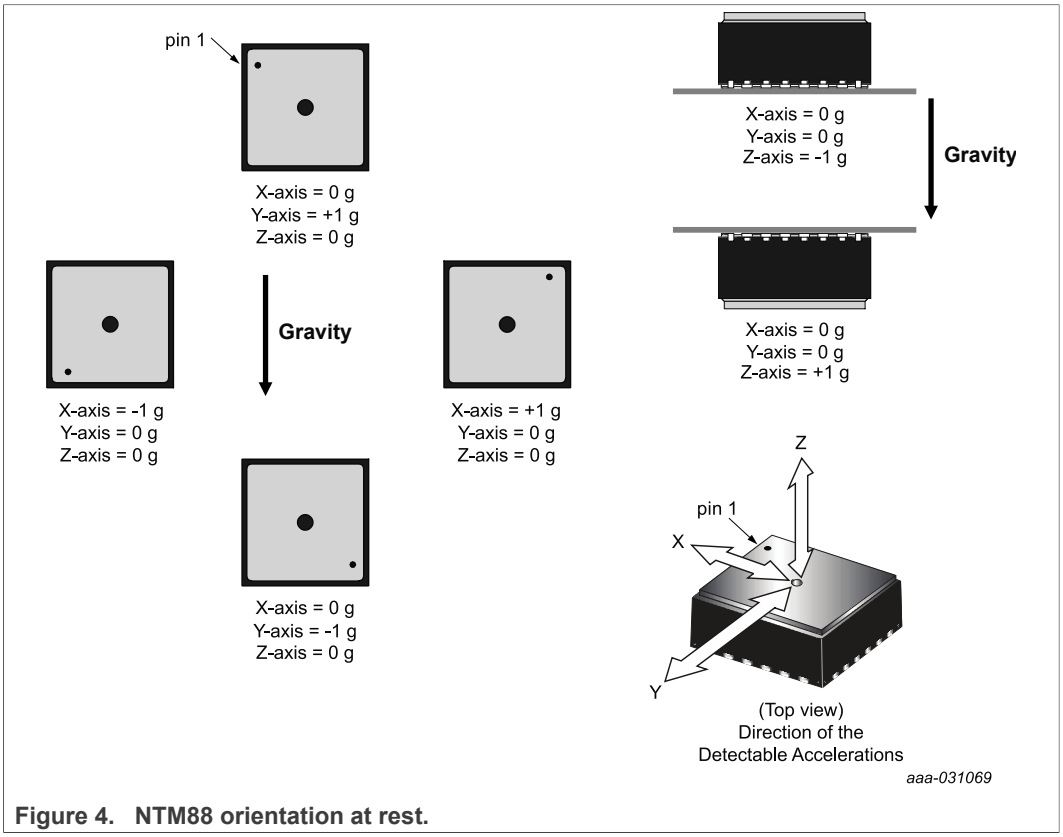


Figure 4. NTM88 orientation at rest.

8 Central processing unit

8.1 Introduction

This section provides summary information about the registers, addressing modes, and instruction set of the CPU of the HCS08 Family. For a more detailed discussion, refer to the HCS08 Family Reference Manual, volume 1, NXP Semiconductor document order number HCS08RMV1/D.

The HCS08 CPU is fully source- and object-code-compatible with the M68HC08 CPU. Several instructions and enhanced addressing modes were added to improve C compiler efficiency and to support a new BACKGROUND DEBUG system which replaces the monitor mode of earlier M68HC08 microcontrollers (MCU).

8.2 Features

Features of the HCS08 CPU include:

- Object code fully upward compatible with M68HC05 and M68HC08 Families
- All registers and memory are mapped to a single 64 kB address space
- 16-bit stack pointer (any size stack anywhere in 64 kB address space)
- 16-bit index register (H:X) with powerful indexed addressing modes
- 8-bit accumulator (A)
- Many instructions treat X as a second general-purpose 8-bit register

- Seven addressing modes:
  - Inherent — Operands in internal registers
  - Relative — 8-bit signed offset to branch destination
  - Immediate — Operand in next object code byte(s)
  - Direct — Operand in memory at 0x0000–0x00FF
  - Extended — Operand anywhere in 64 kB address space
  - Indexed relative to H:X — Five submodes including auto-increment
  - Indexed relative to SP — Improves C efficiency dramatically
- Memory-to-memory data move instructions with four address mode combinations
- Overflow, half-carry, negative, zero, and carry condition codes support conditional branching on the results of signed, unsigned, and binary-coded decimal (BCD) operations
- Efficient bit manipulation instructions
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- STOP and WAIT instructions to invoke low-power operating modes

8.3 Programmer's model and CPU registers

Figure 5 shows the five CPU registers. CPU registers are not part of the memory map.

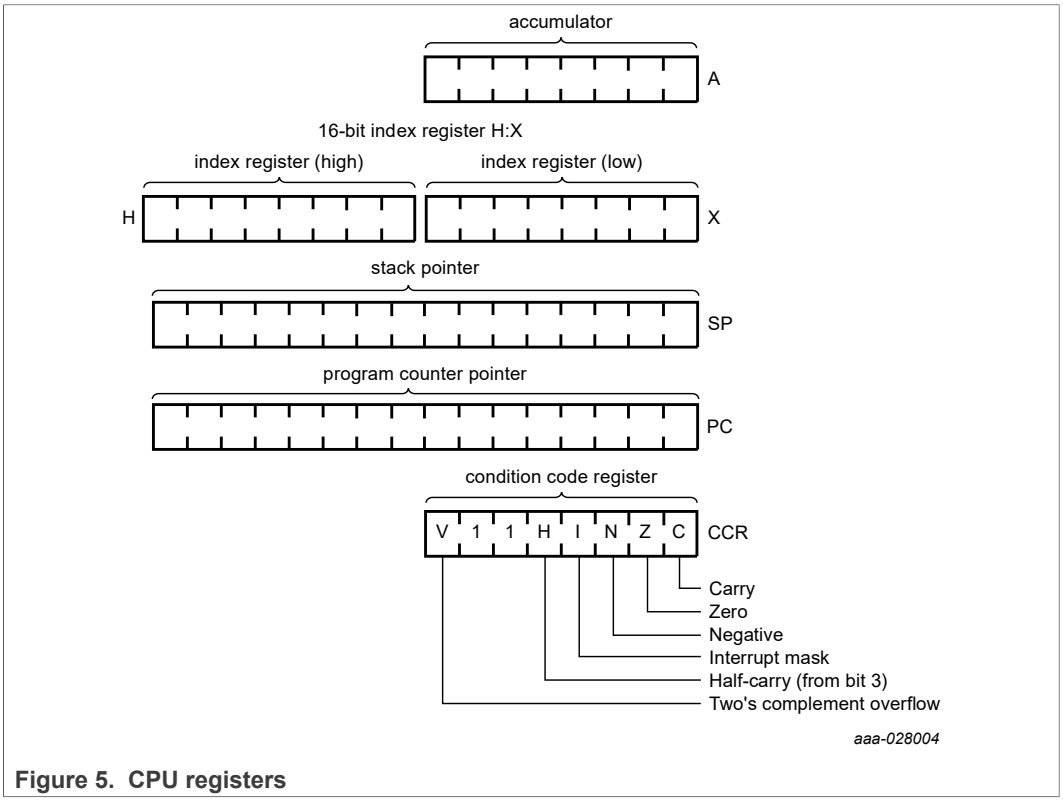


Figure 5. CPU registers

8.3.1 Accumulator (A)

The A accumulator is a general-purpose 8-bit register. One operand input to the arithmetic logic unit (ALU) is connected to the accumulator and the ALU results are often stored into the A accumulator after arithmetic and logical operations. The accumulator can be loaded from memory using various addressing modes to specify the address

where the loaded data comes from, or the contents of A can be stored to memory using various addressing modes to specify the address where data from A will be stored.

Reset has no effect on the contents of the A accumulator.

8.3.2 Index register (H:X)

This 16-bit register is actually two separate 8-bit registers (H and X), which often work together as a 16-bit address pointer where H holds the upper byte of an address and X holds the lower byte of the address. All indexed addressing mode instructions use the full 16-bit value in H:X as an index reference pointer; however, for compatibility with the earlier M68HC05 Family, some instructions operate only on the low-order 8-bit half (X).

Many instructions treat X as a second general-purpose 8-bit register that can be used to hold 8-bit data values. X can be cleared, incremented, decremented, complemented, negated, shifted, or rotated. Transfer instructions allow data to be transferred from A or transferred to A where arithmetic and logical operations can then be performed.

For compatibility with the earlier M68HC05 Family, H is forced to 0x00 during reset. Reset has no effect on the contents of X.

8.3.3 Stack pointer (SP)

This 16-bit address pointer register points at the next available location on the automatic last-in-first-out (LIFO) stack. The stack may be located anywhere in the 64 kB address space that has RAM and can be any size up to the amount of available RAM. The stack is used to automatically save the return address for subroutine calls, the return address and CPU registers during interrupts, and for local variables. The AIS (add immediate to stack pointer) instruction adds an 8-bit signed immediate value to SP. This is most often used to allocate or deallocate space for local variables on the stack.

SP is forced to 0x00FF at reset for compatibility with the earlier M68HC05 Family. HCS08 programs normally change the value in SP to the address of the last location (highest address) in on-chip RAM during reset initialization to free up direct page RAM (from the end of the on-chip registers to 0x00FF).

The RSP (reset stack pointer) instruction was included for compatibility with the M68HC05 Family and is seldom used in new HCS08 programs because it only affects the low-order half of the stack pointer.

8.3.4 Program counter (PC)

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched.

During normal program execution, the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, interrupt, and return operations load the program counter with an address other than that of the next sequential location. This is called a change-of-flow.

During reset, the program counter is loaded with the reset vector that is located at 0xFFFE and 0xFFFF. The vector stored there is the address of the first instruction that will be executed after exiting the reset state.

8.3.5 Condition code register (CCR)

The 8-bit condition code register contains the interrupt mask (I) and five flags that indicate the results of the instruction just executed. Bits 6 and 5 are set permanently to 1. The following paragraphs describe the functions of the condition code bits in general terms. For a more detailed explanation of how each instruction sets the CCR bits, refer to the HCS08 Family Reference Manual, volume 1, NXP Semiconductors document order number HCS08RMv1.

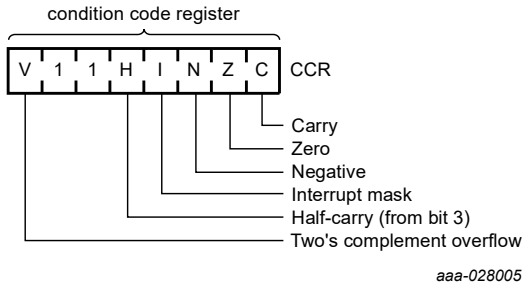


Figure 6. Condition code register

Table 6. CCR register field descriptions

Field	Description
7 V	Two's Complement Overflow Flag — The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag. 0 No overflow 1 Overflow
4 H	Half-Carry Flag — The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C condition code bits to automatically add a correction value to the result from a previous ADD or ADC on BCD operands to correct the result to a valid BCD value. 0 No carry between bits 3 and 4 1 Carry between bits 3 and 4
3 I	Interrupt Mask Bit — When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the first instruction of the interrupt service routine is executed. Interrupts are not recognized at the instruction boundary after any instruction that clears I (CLI or TAP). This ensures that the next instruction after a CLI or TAP will always be executed without the possibility of an intervening interrupt, provided I was set. 0 Interrupts enabled 1 Interrupts disabled
2 N	Negative Flag — The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result. Simply loading or storing an 8-bit, or 16-bit value causes N to be set if the most significant bit of the loaded or stored value was 1. 0 Non-negative result 1 Negative result

Table 6. CCR register field descriptions...*continued*

Field	Description
1 Z	Zero Flag — The CPU sets the zero flag when an arithmetic operation, logic operation, or data manipulation produces a result of 0x00 or 0x0000. Simply loading or storing an 8-bit, or 16-bit value causes Z to be set if the loaded or stored value was all 0s. 0 Non-zero result 1 Zero result
0 C	Carry/Borrow Flag — The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some instructions — such as bit test and branch, shift, and rotate — also clear or set the carry/borrow flag. 0 No carry out of bit 7 1 Carry out of bit 7

8.4 Addressing modes

Addressing modes define the way the CPU accesses operands and data. In the HCS08, all memory, status and control registers, and input/output (I/O) ports share a single 64 kB linear address space so a 16-bit binary address can uniquely identify any memory location. This arrangement means that the same instructions that access variables in RAM can also be used to access I/O and control registers or nonvolatile program space.

Some instructions use more than one addressing mode. For instance, move instructions use one addressing mode to specify the source operand and a second addressing mode to specify the destination address. Instructions such as BRCLR, BRSET, CBEQ, and DBNZ use one addressing mode to specify the location of an operand for a test and then use relative addressing mode to specify the branch destination address when the tested condition is true. For BRCLR, BRSET, CBEQ, and DBNZ, the addressing mode listed in the instruction set tables is the addressing mode needed to access the operand to be tested, and relative addressing mode is implied for the branch destination.

8.4.1 Inherent addressing mode (INH)

In this addressing mode, operands needed to complete the instruction (if any) are located within CPU registers so the CPU does not need to access memory to get any operands.

8.4.2 Relative addressing mode (REL)

Relative addressing mode is used to specify the destination location for branch instructions. A signed 8-bit offset value is located in the memory location immediately following the opcode. During execution, if the branch condition is true, the signed offset is sign-extended to a 16-bit value and is added to the current contents of the program counter, which causes program execution to continue at the branch destination address.

8.4.3 Immediate addressing mode (IMM)

In immediate addressing mode, the operand needed to complete the instruction is included in the object code immediately following the instruction opcode in memory. In the case of a 16-bit immediate operand, the high-order byte is located in the next memory location after the opcode, and the low-order byte is located in the next memory location after that.

8.4.4 Direct addressing mode (DIR)

In direct addressing mode, the instruction includes the low-order 8 bits of an address in the direct page (0x0000–0x00FF). During execution, a 16-bit address is formed by concatenating an implied 0x00 for the high-order half of the address and the direct address from the instruction to get the 16-bit address where the desired operand is located. DIR is faster and more memory efficient than specifying a complete 16-bit address for the operand.

8.4.5 Extended addressing mode (EXT)

In extended addressing mode, the full 16-bit address of the operand is located in the next 2 bytes of program memory after the opcode (high byte first).

8.4.6 Indexed addressing mode

Indexed addressing mode has seven variations including five that use the 16-bit H:X index register pair and two that use the stack pointer as the base reference.

8.4.6.1 Indexed, no offset (IX)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair as the address of the operand needed to complete the instruction.

8.4.6.2 Indexed, no offset with post increment (IX+)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair as the address of the operand needed to complete the instruction. The index register pair is then incremented (H:X = H:X + 0x0001) after the operand has been fetched. This addressing mode is only used for MOV and CBEQ instructions.

8.4.6.3 Indexed, 8-bit offset (IX1)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair plus an unsigned 8-bit offset included in the instruction as the address of the operand needed to complete the instruction.

8.4.6.4 Indexed, 8-bit offset with post increment (IX1+)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair plus an unsigned 8-bit offset included in the instruction as the address of the operand needed to complete the instruction. The index register pair is then incremented (H:X = H:X + 0x0001) after the operand has been fetched. This addressing mode is used only for the CBEQ instruction.

8.4.6.5 Indexed, 16-bit offset (IX2)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair plus a 16-bit offset included in the instruction as the address of the operand needed to complete the instruction.

8.4.6.6 SP-Relative, 8-bit offset (SP1)

This variation of indexed addressing uses the 16-bit value in the stack pointer (SP) plus an unsigned 8-bit offset included in the instruction as the address of the operand needed to complete the instruction.

8.4.6.7 SP-Relative, 16-bit offset (SP2)

This variation of indexed addressing uses the 16-bit value in the stack pointer (SP) plus a 16-bit offset included in the instruction as the address of the operand needed to complete the instruction.

8.5 Special operations

The CPU performs a few special operations that are similar to instructions but do not have opcodes like other CPU instructions. In addition, a few instructions such as STOP and WAIT directly affect other MCU circuitry. This section provides additional information about these operations.

8.5.1 Reset sequence

Reset can be caused by a power-on-reset (POR) event, internal conditions such as the COP (computer operating properly) watchdog, or by assertion of an external active-low reset pin. When a reset event occurs, the CPU immediately stops whatever it is doing (the MCU does not wait for an instruction boundary before responding to a reset event). For a more detailed discussion about how the MCU recognizes resets and determines the source, see Section 10.11 "Reset, interrupts and system configuration".

The reset event is considered concluded when the sequence to determine whether the reset came from an internal source is done and when the reset pin is no longer asserted. At the conclusion of a reset event, the CPU performs a 6-cycle sequence to fetch the reset vector from 0xFFFE and 0xFFFF and to fill the instruction queue in preparation for execution of the first program instruction.

8.5.2 Interrupt sequence

When an interrupt is requested, the CPU completes the current instruction before responding to the interrupt. At this point, the program counter is pointing at the start of the next instruction, which is where the CPU should return after servicing the interrupt. The CPU responds to an interrupt by performing the same sequence of operations as for a software interrupt (SWI) instruction, except the address used for the vector fetch is determined by the highest priority interrupt that is pending when the interrupt sequence started.

The CPU sequence for an interrupt is:

1. Store the contents of PCL, PCH, X, A, and CCR on the stack, in that order.
2. Set the I bit in the CCR.
3. Fetch the high-order half of the interrupt vector.
4. Fetch the low-order half of the interrupt vector.
5. Delay for one free bus cycle.
6. Fetch 3 bytes of program information, starting at the address indicated by the interrupt vector, to fill the instruction queue in preparation for execution of the first instruction in the interrupt service routine.

After the CCR contents are pushed onto the stack, the I bit in the CCR is set to prevent other interrupts while in the interrupt service routine. Although it is possible to clear the I bit with an instruction in the interrupt service routine, this would allow nesting of interrupts (which is not recommended because it leads to programs that are difficult to debug and maintain).

For compatibility with the earlier M68HC05 MCUs, the high-order half of the H:X index register pair (H) is not saved on the stack as part of the interrupt sequence. The user must use a PSHH instruction at the beginning of the service routine to save H and then use a PULH instruction just before the RTI that ends the interrupt service routine. It is not necessary to save H if you are certain that the interrupt service routine does not use any instructions or auto-increment addressing modes that might change the value of H.

The software interrupt (SWI) instruction is like a hardware interrupt except that it is not masked by the global I bit in the CCR and it is associated with an instruction opcode within the program so it is not asynchronous to program execution.

8.5.3 WAIT mode operation

The WAIT instruction enables interrupts by clearing the I bit in the CCR. It then halts the clocks to the CPU to reduce overall power consumption while the CPU is waiting for the interrupt or reset event that will wake the CPU from WAIT mode. When an interrupt or reset event occurs, the CPU clocks resume and the interrupt or reset event are processed normally.

If a serial BACKGROUND command is issued to the MCU through the BACKGROUND DEBUG interface while the CPU is in WAIT mode, CPU clocks resume and the CPU enters ACTIVE BACKGROUND mode where other serial BACKGROUND commands can be processed. This ensures that a host development system can still gain access to a target MCU even if it is in WAIT mode.

8.5.4 STOP mode operation

Usually, all system clocks, including the crystal oscillator (when used), are halted during STOP mode to minimize power consumption. In such systems, external circuitry is needed to control the time spent in STOP mode and to issue a signal to wake up the target MCU when it is time to resume processing. Unlike the earlier M68HC05 and M68HC08 MCUs, the HCS08 can be configured to keep a minimum set of clocks running in STOP mode. This optionally allows an internal periodic signal to wake the target MCU from STOP mode.

When a host debug system is connected to the BACKGROUND DEBUG pin (BKGD) and the ENBDM control bit has been set by a serial command through the BACKGROUND interface (or because the MCU was reset into ACTIVE BACKGROUND mode), the oscillator is forced to remain active when the MCU enters STOP mode. In this case, if a serial BACKGROUND command is issued to the MCU through the BACKGROUND DEBUG interface while the CPU is in STOP mode, CPU clocks resume and the CPU enters ACTIVE BACKGROUND mode where other serial BACKGROUND commands can be processed. This ensures that a host development system can still gain access to a target MCU even if it is in STOP mode.

Recovery from STOP mode depends on the particular HCS08 and whether the oscillator was stopped in STOP mode. See Section 10.8 "Modes of operation" for more details.

8.5.5 BGND instruction

The BGND instruction is new to the HCS08 compared to the M68HC08. BGND would not be used in normal user programs because it forces the CPU to stop processing user instructions and enter the ACTIVE BACKGROUND mode. The only way to resume execution of the user program is through reset or by a host debug system issuing a GO, TRACE1, or TAGGO serial command through the BACKGROUND DEBUG interface.

Software-based breakpoints can be set by replacing an opcode at the desired breakpoint address with the BGND opcode. When the program reaches this breakpoint address, the CPU is forced to ACTIVE BACKGROUND mode rather than continuing the user program.

8.6 HCS08 instruction set summary

8.6.1 Instruction set summary nomenclature

The nomenclature listed here is used in the instruction descriptions in Table 7.

8.6.2 Operators

( ) = Contents of register or memory location shown inside parentheses

← = Is loaded with (read: "gets")

& = Boolean AND

| = Boolean OR

⊕ = Boolean exclusive-OR

× = Multiply

÷ = Divide

: = Concatenate

+ = Add

– = Negate (two’s complement)

8.6.3 CPU registers

A = Accumulator

CCR = Condition code register

H = Index register, higher order (most significant) 8 bits

X = Index register, lower order (least significant) 8 bits

PC = Program counter

PCH = Program counter, higher order (most significant) 8 bits

PCL = Program counter, lower order (least significant) 8 bits

SP = Stack pointer

8.6.4 Memory and addressing

M = A memory location or absolute data, depending on addressing mode

M:M + 0x0001 = A 16-bit value in two consecutive memory locations. The higher order (most significant) 8 bits are located at the address of M, and the lower order (least significant) 8 bits are located at the next higher sequential address.

8.6.5 Condition code register (CCR) bits

V = Two’s complement overflow indicator, bit 7

H = Half carry, bit 4

I = Interrupt mask, bit 3

N = Negative indicator, bit 2

Z = Zero indicator, bit 1

C = Carry/borrow, bit 0 (carry out of bit 7)

8.6.6 CCR activity notation

– = Bit not affected

0 = Bit forced to 0

1 = Bit forced to 1

Þ = Bit set or cleared according to results of operation

U = Undefined after the operation

8.6.7 Machine coding notation

dd = Low-order 8 bits of a direct address 0x0000–0x00FF (high byte assumed to be 0x00)

ee = Upper 8 bits of 16-bit offset

ff = Lower 8 bits of 16-bit offset or 8-bit offset

ii = One byte of immediate data

jj = High-order byte of a 16-bit immediate data value

kk = Low-order byte of a 16-bit immediate data value

hh = High-order byte of 16-bit extended address

ll = Low-order byte of 16-bit extended address

rr = Relative offset

8.6.8 Source form

Everything in the source forms columns, except expressions in italic characters, is literal information that must appear in the assembly source file exactly as shown. The initial 3- to 5-letter mnemonic is always a literal expression. All commas, pound signs (#), parentheses, and plus signs (+) are literal characters.

n — Any label or expression that evaluates to a single integer in the range 0–7

opr8i — Any label or expression that evaluates to an 8-bit immediate value

opr16i — Any label or expression that evaluates to a 16-bit immediate value

opr8a — Any label or expression that evaluates to an 8-bit value. The instruction treats this 8-bit value as the low order 8 bits of an address in the direct page of the 64 kB address space (0x00xx).

opr16a — Any label or expression that evaluates to a 16-bit value. The instruction treats this value as an address in the 64 kB address space.

oprx8 — Any label or expression that evaluates to an unsigned 8-bit value, used for indexed addressing

oprx16 — Any label or expression that evaluates to a 16-bit value. Because the HCS08 has a 16-bit address bus, this can be either a signed or an unsigned value.

rel — Any label or expression that refers to an address that is within –128 to +127 locations from the next address after the last byte of object code for the current instruction. The assembler calculates the 8-bit signed offset and include it in the object code for this instruction.

8.6.9 Address modes

INH = Inherent (no operands)

IMM = 8-bit or 16-bit immediate

DIR = 8-bit direct

EXT = 16-bit extended

IX = 16-bit indexed no offset

IX+ = 16-bit indexed no offset, post increment (CBEQ and MOV only)

IX1 = 16-bit indexed with 8-bit offset from H:X

IX1+ = 16-bit indexed with 8-bit offset, post increment (CBEQ only)

IX2 = 16-bit indexed with 16-bit offset from H:X

rel = 8-bit relative offset

SP1 = Stack pointer with 8-bit offset

SP2 = Stack pointer with 16-bit offset

Table 7. HCS08 instruction set summary

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles [1]
			V	H	I	N	Z	C				
ADC #*opr8i*	Add with Carry	A ← (A) + (M) + (C)	Þ	Þ	–	Þ	Þ	Þ	IMM	A9	ii	2
ADC *opr8a*									DIR	B9	dd	3
ADC *opr16a*									EXT	C9	hh ll	4
ADC *oprx16*,X									IX2	D9	ee ff	4
ADC *oprx8*,X									IX1	E9	ff	3
ADC,X									IX	F9		3
ADC *oprx16*,SP									SP2	9ED9	ee ff	5
ADC *oprx8*,SP									SP1	9EE9	ff	4

Table 7. HCS08 instruction set summary*...continued*

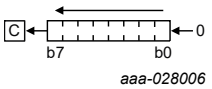
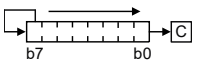
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles [1]
			V	H	I	N	Z	C				
ADD #opr8i ADD *opr8a* ADD *opr16a* ADD *oprx16*,X ADD *oprx8*,X ADD ,X ADD *oprx16*,SP ADD *oprx8*,SP	Add without Carry	A ← (A) + (M)	Þ	Þ	–	Þ	Þ	Þ	IMM DIR EXT IX2 IX1 IX SP2 SP1	AB BB CB DB EB FB 9EDB 9EEB	ii dd hh ll ee ff ff	2 3 4 4 3 3 5 4
AIS *#opr8i*	Add Immediate Value (Signed) to Stack Pointer	SP ← (SP) + (M) M is sign extended to a 16-bit value	–	–	–	–	–	–	IMM	A7	ii	2
AIX *#opr8i*	Add Immediate Value (Signed) to Index Register (H:X)	H:X ← (H:X) + (M) M is sign extended to a 16-bit value	–	–	–	–	–	–	IMM	AF	ii	2
AND *#opr8i* AND *opr8a* AND *opr16a* AND *oprx16*,X AND *oprx8*,X AND ,X AND *oprx16*,SP AND *oprx8*,SP	Logical AND	A ← (A) & (M)	0	–	–	Þ	Þ	–	IMM DIR EXT IX2 IX1 IX SP2 SP1	A4 B4 C4 D4 E4 F4 9ED4 9EE4	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
ASL *opr8a* ASLA ASLX ASL *oprx8*,X ASL ,X ASL *oprx8*,SP	Arithmetic Shift Left (Same as LSL)	 *aaa-028006*	Þ	–	–	Þ	Þ	Þ	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	5 1 1 5 4 6
ASR *opr8a* ASRA ASRX ASR *oprx8*,X ASR ,X ASR *oprx8*,SP	Arithmetic Shift Right	 *aaa-028007*	Þ	–	–	Þ	Þ	Þ	DIR INH INH IX1 IX SP1	37 47 57 67 77 9E67	dd ff ff	5 1 1 5 4 6
BCC *rel*	Branch if Carry Bit Clear	Branch if (C) = 0	–	–	–	–	–	–	*rel*	24	rr	3
BCLR *n,opr8a*	Clear Bit n in Memory	Mn ← 0	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	Branch if (C) = 1	–	–	–	–	–	–	*rel*	25	rr	3
BEQ *rel*	Branch if Equal	Branch if (Z) = 1	–	–	–	–	–	–	*rel*	27	rr	3

Table 7. HCS08 instruction set summary*...continued*

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles [1]
			V	H	I	N	Z	C				
BGE *rel*	Branch if Greater Than or Equal To (Signed Operands)	Branch if (N ⊕ V) = 0	–	–	–	–	–	–	*rel*	90	rr	3
BGND	Enter ACTIVE BACK-GROUND if ENBDM = 1	Waits For and Processes BDM Commands Until GO, TRACE1, or TAGGO	–	–	–	–	–	–	INH	82		5+
BGT *rel*	Branch if Greater Than (Signed Operands)	Branch if (Z) | (N ⊕ V) = 0	–	–	–	–	–	–	*rel*	92	rr	3
BHCC *rel*	Branch if Half Carry Bit Clear	Branch if (H) = 0	–	–	–	–	–	–	*rel*	28	rr	3
BHCS *rel*	Branch if Half Carry Bit Set	Branch if (H) = 1	–	–	–	–	–	–	*rel*	29	rr	3
BHI *rel*	Branch if Higher	Branch if (C) | (Z) = 0	–	–	–	–	–	–	*rel*	22	rr	3
BHS *rel*	Branch if Higher or Same (Same as BCC)	Branch if (C) = 0	–	–	–	–	–	–	*rel*	24	rr	3
BIH *rel*	Branch if IRQ Pin High	Branch if IRQ pin = 1	–	–	–	–	–	–	*rel*	2F	rr	3
BIL *rel*	Branch if IRQ Pin Low	Branch if IRQ pin = 0	–	–	–	–	–	–	*rel*	2E	rr	3
BIT *#opr8i* BIT *opr8a* BIT *opr16a* BIT *oprx16*,X BIT *oprx8*,X BIT ,X BIT *oprx16*,SP BIT *oprx8*,SP	Bit Test	(A) & (M) (CCR Updated but Operands Not Changed)	0	–	–	Þ	Þ	–	IMM DIR EXT IX2 IX1 IX SP2 SP1	A5 B5 C5 D5 E5 F5 9ED5 9EE5	ii dd hh ll ee ff ff	2 3 4 4 3 3 5 4
BLE *rel*	Branch if Less Than or Equal To (Signed Operands)	Branch if (Z) | (N ⊕ V) = 1	–	–	–	–	–	–	*rel*	93	rr	3
BLO *rel*	Branch if Lower (Same as BCS)	Branch if (C) = 1	–	–	–	–	–	–	*rel*	25	rr	3
BLS *rel*	Branch if Lower or Same	Branch if (C) | (Z) = 1	–	–	–	–	–	–	*rel*	23	rr	3
BLT *rel*	Branch if Less Than (Signed Operands)	Branch if (N ⊕ V) = 1	–	–	–	–	–	–	*rel*	91	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	Branch if (I) = 0	–	–	–	–	–	–	*rel*	2C	rr	3
BMI *rel*	Branch if Minus	Branch if (N) = 1	–	–	–	–	–	–	*rel*	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	Branch if (I) = 1	–	–	–	–	–	–	*rel*	2D	rr	3
BNE *rel*	Branch if Not Equal	Branch if (Z) = 0	–	–	–	–	–	–	*rel*	26	rr	3
BPL *rel*	Branch if Plus	Branch if (N) = 0	–	–	–	–	–	–	*rel*	2A	rr	3
BRA *rel*	Branch Always	No Test	–	–	–	–	–	–	*rel*	20	rr	3

Table 7. HCS08 instruction set summary*...continued*

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles [1]
			V	H	I	N	Z	C				
BRCLR n,opr8a,rel	Branch if Bit n in Memory Clear	Branch if (Mn) = 0	–	–	–	–	–	Þ	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	Uses 3 Bus Cycles	–	–	–	–	–	–	*rel*	21	rr	3
BRSET *n,opr8a,rel*	Branch if Bit n in Memory Set	Branch if (Mn) = 1	–	–	–	–	–	Þ	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BSET *n,opr8a*	Set Bit n in Memory	Mn ← 1	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BSR *rel*	Branch to Subroutine	PC ← (PC) + 0x0002 push (PCL); SP ← (SP) – 0x0001 push (PCH); SP ← (SP) – 0x0001 PC ← (PC) + *rel*	–	–	–	–	–	–	*rel*	AD	rr	5
CBEQ *opr8a,rel* CBEQA *#opr8i,rel* CBEQX *#opr8i,rel* CBEQ *oprx8,X+,rel* CBEQ ,X+,*rel* CBEQ *oprx8,SP,rel*	Compare and Branch if Equal	Branch if (A) = (M) Branch if (A) = (M) Branch if (X) = (M) Branch if (A) = (M) Branch if (A) = (M) Branch if (A) = (M)	–	–	–	–	–	–	DIR IMM IMM IX1+ IX+ SP1	31 41 51 61 71 9E61	dd rr ii rr ii rr ff rr rr ff rr	5 4 4 5 5 6
CLC	Clear Carry Bit	C ← 0	–	–	–	–	–	0	INH	98		1
CLI	Clear Interrupt Mask Bit	I ← 0	–	–	0	–	–	–	INH	9A		1
CLR *opr8a* CLRA CLRX CLRH CLR *oprx8*,X CLR ,X CLR *oprx8*,SP	Clear	M ← 0x00 A ← 0x00 X ← 0x00 H ← 0x00 M ← 0x00 M ← 0x00 M ← 0x00	0	–	–	0	1	–	DIR INH INH INH IX1 IX SP1	3F 4F 5F 8C 6F 7F 9E6F	dd ff ff	5 1 1 1 5 4 6

Table 7. HCS08 instruction set summary*...continued*

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles [1]
			V	H	I	N	Z	C				
CMP *#opr8i* CMP *opr8a* CMP *opr16a* CMP *oprx16*,X CMP *oprx8*,X CMP ,X CMP *oprx16*,SP CMP *oprx8*,SP	Compare Accumulator with Memory	(A) – (M) (CCR Updated But Operands Not Changed)	Þ	–	–	Þ	Þ	Þ	IMM DIR EXT IX2 IX1 IX SP2 SP1	A1 B1 C1 D1 E1 F1 9ED1 9EE1	ii dd hh ll ee ff ff	2 3 4 4 3 3 5 4
COM *opr8a* COMA COMX COM *oprx8*,X COM ,X COM *oprx8*,SP	Complement (One's Complement)	M ← ($\overline{M}$)= 0xFF – (M) A ← ($\overline{A}$) = 0xFF – (A) X ← ($\overline{X}$) = 0xFF – (X) M ← ($\overline{M}$) = 0xFF – (M) M ← ($\overline{M}$) = 0xFF – (M) M ← ($\overline{M}$) = 0xFF – (M)	0	–	–	Þ	Þ	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd ff ff	5 1 1 5 4 6
CPHX *opr16a* CPHX *#opr16i* CPHX *opr8a* CPHX *oprx8*,SP	Compare Index Register (H:X) with Memory	(H:X) – (M:M + 0x0001) (CCR Updated But Operands Not Changed)	Þ	–	–	Þ	Þ	Þ	EXT IMM DIR SP1	3E 65 75 9EF3	hh ll jj kk dd ff	6 3 5 6
CPX *#opr8i* CPX *opr8a* CPX *opr16a* CPX *oprx16*,X CPX *oprx8*,X CPX ,X CPX *oprx16*,SP CPX *oprx8*,SP	Compare X (Index Register Low) with Memory	(X) – (M) (CCR Updated But Operands Not Changed)	Þ	–	–	Þ	Þ	Þ	IMM DIR EXT IX2 IX1 IX SP2 SP1	A3 B3 C3 D3 E3 F3 9ED3 9EE3	ii dd hh ll ee ff ff	2 3 4 4 3 3 5 4
DAA	Decimal Adjust Accumulator After ADD or ADC of BCD Values	$(A)_{10}$	U	–	–	Þ	Þ	Þ	INH	72		1
DBNZ *opr8a,rel* DBNZA *rel* DBNZX *rel* DBNZ *oprx8,X,rel* DBNZ ,X,*rel* DBNZ *oprx8*,SP,*rel*	Decrement and Branch if Not Zero	Decrement A, X, or M Branch if (result) ≠ 0 DBNZX Affects X Not H	–	–	–	–	–	–	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	7 4 4 7 6 8
DEC *opr8a* DECA DECX DEC *oprx8*,X DEC ,X DEC *oprx8*,SP	Decrement	M ← (M) – 0x01 A ← (A) – 0x01 X ← (X) – 0x01 M ← (M) – 0x01 M ← (M) – 0x01 M ← (M) – 0x01	Þ	–	–	Þ	Þ	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff	5 1 1 5 4 6
DIV	Divide	A ← (H:A) ÷ (X) H ← Remainder	–	–	–	–	Þ	Þ	INH	52		6

Table 7. HCS08 instruction set summary*...continued*

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles [1]
			V	H	I	N	Z	C				
EOR *#opr8i* EOR *opr8a* EOR *opr16a* EOR *oprx16*,X EOR *oprx8*,X EOR ,X EOR *oprx16*,SP EOR *oprx8*,SP	Exclusive OR Memory with Accumulator	A ← (A ⊕ M)	0	–	–	Þ	Þ	–	IMM DIR EXT IX2 IX1 IX SP2 SP1	A8 B8 C8 D8 E8 F8 9ED8 9EE8	ii dd hh ll ee ff ff	2 3 4 4 3 3 5 4
INC *opr8a* INCA INCX INC *oprx8*,X INC ,X INC *oprx8*,SP	Increment	M ← (M) + 0x01 A ← (A) + 0x01 X ← (X) + 0x01 M ← (M) + 0x01 M ← (M) + 0x01 M ← (M) + 0x01	Þ	–	–	Þ	Þ	–	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff	5 1 1 5 4 6
JMP *opr8a* JMP *opr16a* JMP *oprx16*,X JMP *oprx8*,X JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff	3 4 4 3 3
JSR *opr8a* JSR *opr16a* JSR *oprx16*,X JSR *oprx8*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 0x0001 Push (PCH); SP ← (SP) – 0x0001 PC ← Unconditional Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	5 6 6 5 5
LDA *#opr8i* LDA *opr8a* LDA *opr16a* LDA *oprx16*,X LDA *oprx8*,X LDA ,X LDA *oprx16*,SP LDA *oprx8*,SP	Load Accumulator from Memory	A ← (M)	0	–	–	Þ	Þ	–	IMM DIR EXT IX2 IX1 IX SP2 SP1	A6 B6 C6 D6 E6 F6 9ED6 9EE6	ii dd hh ll ee ff ff	2 3 4 4 3 3 5 4
LDHX *#opr16i* LDHX *opr8a* LDHX *opr16a* LDHX ,X LDHX *oprx16*,X LDHX *oprx8*,X LDHX *oprx8*,SP	Load Index Register (H:X) from Memory	H:X ← (M:M + 0x0001)	0	–	–	Þ	Þ	–	IMM DIR EXT IX IX2 IX1 SP1	45 55 32 9EAE 9EBE 9ECE 9EFE	jj kk dd hh ll ee ff ff ff	3 4 5 5 6 5 5
LDX *#opr8i* LDX *opr8a* LDX *opr16a* LDX *oprx16*,X LDX *oprx8*,X LDX ,X LDX *oprx16*,SP LDX *oprx8*,SP	Load X (Index Register Low) from Memory	X ← (M)	0	–	–	Þ	Þ	–	IMM DIR EXT IX2 IX1 IX SP2 SP1	AE BE CE DE EE FE 9EDE 9EEE	ii dd hh ll ee ff ff ff ee ff ff	2 3 4 4 3 3 5 4

Table 7. HCS08 instruction set summary*...continued*

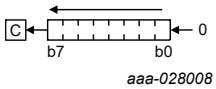
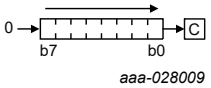
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles [1]
			V	H	I	N	Z	C				
LSL *oprx8*,SP	Logical Shift Left (Same as ASL)	 aaa-028008	Þ	–	–	Þ	Þ	Þ	SP1	9E68	ff	6
LSR *oprx8*,SP	Logical Shift Right	 aaa-028009	Þ	–	–	0	Þ	Þ	SP1	9E64	ff	6
MOV *oprx8*,SP	Move	$(M)_{destination} \leftarrow (M)_{source}$ H:X ← (H:X) + 0x0001 in IX+/DIR and DIR/IX+ Modes	0	–	–	Þ	Þ	–	IX+/DIR	7E	dd	5
MUL	Unsigned multiply	X:A ← (X) × (A)	–	0	–	–	–	0	INH	42		5
NEG *oprx8*,SP	Negate (Two's Complement)	M ← – (M) = 0x00 – (M)	Þ	–	–	Þ	Þ	Þ	SP1	9E60	ff	6
NOP	No Operation	Uses 1 Bus Cycle	–	–	–	–	–	–	INH	9D		1
NSA	Nibble Swap Accumulator	A ← (A[3:0]:A[7:4])	–	–	–	–	–	–	INH	62		1
ORA *oprx8*,SP	Inclusive OR Accumulator and Memory	A ← (A) | (M)	0	–	–	Þ	Þ	–	SP1	9EEA	ff	4
PSHA	Push Accumulator onto Stack	Push (A); SP ← (SP) – 0x0001	–	–	–	–	–	–	INH	87		2
PSHH	Push H (Index Register High) onto Stack	Push (H); SP ← (SP) – 0x0001	–	–	–	–	–	–	INH	8B		2
PSHX	Push X (Index Register Low) onto Stack	Push (X); SP ← (SP) – 0x0001	–	–	–	–	–	–	INH	89		2
PULA	Pull Accumulator from Stack	SP ← (SP + 0x0001); Pull (A)	–	–	–	–	–	–	INH	86		3

Table 7. HCS08 instruction set summary*...continued*

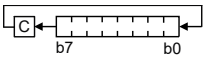
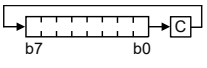
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles [1]
			V	H	I	N	Z	C				
PULH	Pull H (Index Register High) from Stack	SP ← (SP + 0x0001); Pull (H)	–	–	–	–	–	–	INH	8A		3
PULX	Pull X (Index Register Low) from Stack	SP ← (SP + 0x0001); Pull (X)	–	–	–	–	–	–	INH	88		3
ROL *opr8a* ROLA ROLX ROL *oprx8*,X ROL ,X ROL *oprx8*,SP	Rotate Left through Carry	 *aaa-028011*	Þ	–	–	Þ	Þ	Þ	DIR INH INH IX1 IX SP1	39 49 59 69 79 9E69	dd ff ff	5 1 1 5 4 6
ROR *opr8a* RORA RORX ROR *oprx8*,X ROR ,X ROR *oprx8*,SP	Rotate Right through Carry	 *aaa-028010*	Þ	–	–	Þ	Þ	Þ	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	5 1 1 5 4 6
RSP	Reset Stack Pointer	SP ← 0xFF (High Byte Not Affected)	–	–	–	–	–	–	INH	9C		1
RTI	Return from Interrupt	SP ← (SP) + 0x0001; Pull (CCR) SP ← (SP) + 0x0001; Pull (A) SP ← (SP) + 0x0001; Pull (X) SP ← (SP) + 0x0001; Pull (PCH) SP ← (SP) + 0x0001; Pull (PCL)	Þ	Þ	Þ	Þ	Þ	Þ	INH	80		9
RTS	Return from Subroutine	SP ← SP + 0x0001; Pull (PCH) SP ← SP + 0x0001; Pull (PCL)	–	–	–	–	–	–	INH	81		6
SBC *#opr8i* SBC *opr8a* SBC *opr16a* SBC *oprx16*,X SBC *oprx8*,X SBC ,X SBC *oprx16*,SP SBC *oprx8*,SP	Subtract with Carry	A ← (A) – (M) – (C)	Þ	–	–	Þ	Þ	Þ	IMM DIR EXT IX2 IX1 IX SP2 SP1	A2 B2 C2 D2 E2 F2 9ED2 9EE2	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
SEC	Set Carry Bit	C ← 1	–	–	–	–	–	1	INH	99		1
SEI	Set Interrupt Mask Bit	I ← 1	–	–	1	–	–	–	INH	9B		1

Table 7. HCS08 instruction set summary*...continued*

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles [1]
			V	H	I	N	Z	C				
STA *opr8a* STA *opr16a* STA *oprx16*,X STA *oprx8*,X STA ,X STA *oprx16*,SP STA *oprx8*,SP	Store Accumulator in Memory	M ← (A)	0	–	–	Þ	Þ	–	DIR EXT IX2 IX1 IX SP2 SP1	B7 C7 D7 E7 F7 9ED7 9EE7	dd hh ll ee ff ff	3 4 4 3 2 5 4
STHX *opr8a* STHX *opr16a* STHX *oprx8*,SP	Store H:X (Index Reg.)	(M:M + 0x0001) ← (H:X)	0	–	–	Þ	Þ	–	DIR EXT SP1	35 96 9EFF	dd hh ll ff	4 5 5
STOP	Enable Interrupts: Stop Processing Refer to MCU Documentation	I bit ← 0; Stop Processing	–	–	0	–	–	–	INH	8E		2+
STX *opr8a* STX *opr16a* STX *oprx16*,X STX *oprx8*,X STX ,X STX *oprx16*,SP STX *oprx8*,SP	Store X (Low 8 Bits of Index Register) in Memory	M ← (X)	0	–	–	Þ	Þ	–	DIR EXT IX2 IX1 IX SP2 SP1	BF CF DF EF FF 9EDF 9EEF	dd hh ll ee ff ff	3 4 4 3 2 5 4
SUB #*opr8i* SUB *opr8a* SUB *opr16a* SUB *oprx16*,X SUB *oprx8*,X SUB ,X SUB *oprx16*,SP SUB *oprx8*,SP	Subtract	A ← (A) – (M)	Þ	–	–	Þ	Þ	Þ	IMM DIR EXT IX2 IX1 IX SP2 SP1	A0 B0 C0 D0 E0 F0 9ED0 9EE0	ii dd hh ll ee ff ff	2 3 4 4 3 3 5 4
SWI	Software Interrupt	PC ← (PC) + 0x0001 Push (PCL); SP ← (SP) – 0x0001 Push (PCH); SP ← (SP) – 0x0001 Push (X); SP ← (SP) – 0x0001 Push (A); SP ← (SP) – 0x0001 Push (CCR); SP ← (SP) – 0x0001 I ← 1; PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	–	–	1	–	–	–	INH	83		11
TAP	Transfer Accumulator to CCR	CCR ← (A)	Þ	Þ	Þ	Þ	Þ	Þ	INH	84		1

Table 7. HCS08 instruction set summary*...continued*

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles [1]
			V	H	I	N	Z	C				
TAX	Transfer Accumulator to X (Index Register Low)	X ← (A)	–	–	–	–	–	–	INH	97		1
TPA	Transfer CCR to Accumulator	A ← (CCR)	–	–	–	–	–	–	INH	85		1
TST *opr8a* TSTA TSTX TST *oprx8*,X TST ,X TST *oprx8*,SP	Test for Negative or Zero	(M) – 0x00 (A) – 0x00 (X) – 0x00 (M) – 0x00 (M) – 0x00 (M) – 0x00	0	–	–	Þ	Þ	–	DIR INH INH IX1 IX SP1	3D 4D 5D 6D 7D 9E6D	dd ff ff	4 1 1 4 3 5
TSX	Transfer SP to Index Reg.	H:X ← (SP) + 0x0001	–	–	–	–	–	–	INH	95		2
TXA	Transfer X (Index Reg. Low) to Accumulator	A ← (X)	–	–	–	–	–	–	INH	9F		1
TXS	Transfer Index Reg. to SP	SP ← (H:X) – 0x0001	–	–	–	–	–	–	INH	94		2
WAIT	Enable Interrupts; Wait for Interrupt	I bit ← 0; Halt CPU	–	–	0	–	–	–	INH	8F		2+

[1] Bus clock frequency is one-half of the CPU clock frequency.

Table 8. Opcode map (Sheet 1 of 2)

Bit-Manipulation		Branch	Read-Modify-Write						Control				Register/Memory					
00 5 BRSET0 3 DIR	10 5 BSET0 2 DIR	20 3 BRA 2 *rel*	30 5 NEG 2 DIR	40 1 NEGA 1 INH	50 1 NEGX 1 INH	60 5 NEG 2 IX1	70 4 NEG 1 IX	80 9 RTI 1 INH	90 3 BGE 2 *rel*	A0 2 SUB 2 IMM	B0 3 SUB 2 DIR	C0 4 SUB 3 EXT	D0 4 SUB 3 IX2	E0 3 SUB 2 IX1	F0 3 SUB 1 IX
01 5 BRCLR0 3 DIR	11 5 BCLR0 2 DIR	21 3 BRN 2 *rel*	31 5 CBEQ 3 DIR	41 4 CBEQA 3 IMM	51 4 CBEQX 3 IMM	61 5 CBEQ 3 IX1+	71 5 CBEQ 2 IX+	81 6 RTS 1 INH	91 3 BLT 2 *rel*	A1 2 CMP 2 IMM	B1 3 CMP 2 DIR	C1 4 CMP 3 EXT	D1 4 CMP 3 IX2	E1 3 CMP 2 IX1	F1 3 CMP 1 IX
02 5 BRSET1 3 DIR	12 5 BSET1 2 DIR	22 3 BHI 2 *rel*	32 5 LDHX 3 EXT	42 5 MUL 1 INH	52 6 DIV 1 INH	62 1 NSA 1 INH	72 1 DAA 1 INH	82 5+ BGND 1 INH	92 3 BGT 2 *rel*	A2 2 SBC 2 IMM	B2 3 SBC 2 DIR	C2 4 SBC 3 EXT	D2 4 SBC 3 IX2	E2 3 SBC 2 IX1	F2 3 SBC 1 IX
03 5 BRCLR1 3 DIR	13 5 BCLR1 2 DIR	23 3 BLS 2 *rel*	33 5 COM 2 DIR	43 1 COMA 1 INH	53 1 COMX 1 INH	63 5 COM 2 IX1	73 4 COM 1 IX	83 11 SWI 1 INH	93 3 BLE 2 *rel*	A3 2 CPX 2 IMM	B3 3 CPX 2 DIR	C3 4 CPX 3 EXT	D3 4 CPX 3 IX2	E3 3 CPX 2 IX1	F3 3 CPX 1 IX
04 5 BRSET2 3 DIR	14 5 BSET2 2 DIR	24 3 BCC 2 *rel*	34 5 LSR 2 DIR	44 1 LSRA 1 INH	54 1 LSRX 1 INH	64 5 LSR 2 IX1	74 4 LSR 1 IX	84 1 TAP 1 INH	94 2 TXS 1 INH	A4 2 AND 2 IMM	B4 3 AND 2 DIR	C4 4 AND 3 EXT	D4 4 AND 3 IX2	E4 3 AND 2 IX1	F4 3 AND 1 IX
05 5 BRCLR2 3 DIR	15 5 BCLR2 2 DIR	25 3 BCS 2 *rel*	35 4 STHX 2 DIR	45 3 LDHX 3 IMM	55 4 LDHX 2 DIR	65 3 CPHX 3 IMM	75 5 CPHX 2 DIR	85 1 TPA 1 INH	95 2 TSX 1 INH	A5 2 BIT 2 IMM	B5 3 BIT 2 DIR	C5 4 BIT 3 EXT	D5 4 BIT 3 IX2	E5 3 BIT 2 IX1	F5 3 BIT 1 IX
06 5 BRSET3 3 DIR	16 5 BSET3 2 DIR	26 3 BNE 2 *rel*	36 5 ROR 2 DIR	46 1 RORA 1 INH	56 1 RORX 1 INH	66 5 ROR 2 IX1	76 4 ROR 1 IX	86 3 PULA 1 INH	96 5 STHX 3 EXT	A6 2 LDA 2 IMM	B6 3 LDA 2 DIR	C6 4 LDA 3 EXT	D6 4 LDA 3 IX2	E6 3 LDA 2 IX1	F6 3 LDA 1 IX
07 5 BRCLR3 3 DIR	17 5 BCLR3 2 DIR	27 3 BEQ 2 *rel*	37 5 ASR 2 DIR	47 1 ASRA 1 INH	57 1 ASRX 1 INH	67 5 ASR 2 IX1	77 4 ASR 1 IX	87 2 PSHA 1 INH	97 1 TAX 1 INH	A7 2 AIS 2 IMM	B7 3 STA 2 DIR	C7 4 STA 3 EXT	D7 4 STA 3 IX2	E7 3 STA 2 IX1	F7 2 STA 1 IX
08 5 BRSET4 3 DIR	18 5 BSET4 2 DIR	28 3 BHCC 2 *rel*	38 5 LSL 2 DIR	48 1 LSLA 1 INH	58 1 LSLX 1 INH	68 5 LSL 2 IX1	78 4 LSL 1 IX	88 3 PULX 1 INH	98 1 CLC 1 INH	A8 2 EOR 2 IMM	B8 3 EOR 2 DIR	C8 4 EOR 3 EXT	D8 4 EOR 3 IX2	E8 3 EOR 2 IX1	F8 3 EOR 1 IX
09 5 BRCLR4 3 DIR	19 5 BCLR4 2 DIR	29 3 BHCS 2 *rel*	39 5 ROL 2 DIR	49 1 ROLA 1 INH	59 1 ROLX 1 INH	69 5 ROL 2 IX1	79 4 ROL 1 IX	89 2 PSHX 1 INH	99 1 SEC 1 INH	A9 2 ADC 2 IMM	B9 3 ADC 2 DIR	C9 4 ADC 3 EXT	D9 4 ADC 3 IX2	E9 3 ADC 2 IX1	F9 3 ADC 1 IX
0A 5 BRSET5 3 DIR	1A 5 BSET5 2 DIR	2A 3 BPL 2 *rel*	3A 5 DEC 2 DIR	4A 1 DECA 1 INH	5A 1 DECX 1 INH	6A 5 DEC 2 IX1	7A 4 DEC 1 IX	8A 3 PULH 1 INH	9A 1 CLI 1 INH	AA 2 ORA 2 IMM	BA 3 ORA 2 DIR	CA 4 ORA 3 EXT	DA 4 ORA 3 IX2	EA 3 ORA 2 IX1	FA 3 ORA 1 IX
0B 5 BRCLR5 3 DIR	1B 5 BCLR5 2 DIR	2B 3 BMI 2 *rel*	3B 7 DBNZ 3 DIR	4B 4 DBNZA 2 INH	5B 4 DBNZX 2 INH	6B 7 DBNZ 3 IX1	7B 6 DBNZ 2 IX	8B 2 PSHH 1 INH	9B 1 SEI 1 INH	AB 2 ADD 2 IMM	BB 3 ADD 2 DIR	CB 4 ADD 3 EXT	DB 4 ADD 3 IX2	EB 3 ADD 2 IX1	FB 3 ADD 1 IX

Table 8. Opcode map (Sheet 1 of 2)*...continued*

Bit-Manipulation		Branch	Read-Modify-Write						Control				Register/Memory				
0C 5 BRSET6 3 DIR	1C 5 BSET6 2 DIR	2C 3 BMC 2 *rel*	3C 5 INC 2 DIR	4C 1 INCA 1 INH	5C 1 INCX 1 INH	6C 5 INC 2 IX1	7C 4 INC 1 IX	8C 1 CLRH 1 INH	9C 1 RSP 1 INH		BC 3 JMP 2 DIR	CC 4 JMP 3 EXT	DC 4 JMP 3 IX2	EC 3 JMP 2 IX1	FC 3 JMP 1 IX
0D 5 BRCLR6 3 DIR	1D 5 BCLR6 2 DIR	2D 3 BMS 2 *rel*	3D 4 TST 2 DIR	4D 1 TSTA 1 INH	5D 1 TSTX 1 INH	6D 4 TST 2 IX1	7D 3 TST 1 IX		9D 1 NOP 1 INH	AD 5 BSR 2 *rel*	BD 5 JSR 2 DIR	CD 6 JSR 3 EXT	DD 6 JSR 3 IX2	ED 5 JSR 2 IX1	FD 5 JSR 1 IX
0E 5 BRSET7 3 DIR	1E 5 BSET7 2 DIR	2E 3 BIL 2 *rel*	3E 6 CPHX 3 EXT	4E 5 MOV 3 DD	5E 5 MOV 2 DIX+	6E 4 MOV 3 IMD	7E 5 MOV 2 IX+D	8E 2+ STOP 1 INH	9E Page 2	AE 2 LDX 2 IMM	BE 3 LDX 2 DIR	CE 4 LDX 3 EXT	DE 4 LDX 3 IX2	EE 3 LDX 2 IX1	FE 3 LDX 1 IX
0F 5 BRCLR7 3 DIR	1F 5 BCLR7 2 DIR	2F 3 BIH 2 *rel*	3F 5 CLR 2 DIR	4F 1 CLRA 1 INH	5F 1 CLRX 1 INH	6F 5 CLR 2 IX1	7F 4 CLR 1 IX	8F 2+ WAIT 1 INH	9F 1 TXA 1 INH	AF 2 AIX 2 IMM	BF 3 STX 2 DIR	CF 4 STX 3 EXT	DF 4 STX 3 IX2	EF 3 STX 2 IX1	FF 2 STX 1 IX

Table 8. Opcode map (Sheet 1 of 2)

INH	Inherent	*rel*	relative	SP1	Stack pointer, 8-bit offset
IMM	Immediate	IX	Indexed, no offset	SP2	Stack pointer, 16 bit offset
DIR	Direct	IX1	Indexed, 8-bit offset	IX+	Indexed, No offset with post increment
EXT	Extended	IX2	Indexed, 16 bit offset	IX1+	Indexed, 1 byte offset with post increment
DD	DIR to DIR	IMD	IMM to DIR		
IX+D	IX+ to DIR	DIX+	DIR to IX+		

Table 8. Opcode map (Sheet 1 of 2)

Opcode in Hexadecimal	F0 3	HCS08 Cycles
	SUB	Instruction Mnemonic
Number of Bytes	1 IX	Addressing Mode

Table 9. Opcode map (Sheet 2 of 2)

Bit-Manipulation		Branch	Read-Modify-Write						Control				Register/Memory				
						9E60 6 NEG 3 SP1							9ED0 5 SUB 4 SP2	9EE0 4 SUB 3 SP1	
						9E61 6 CBEQ 4 SP1							9ED1 5 CMP 4 SP2	9EE1 4 CMP 3 SP1	
													9ED2 5 SBC 4 SP2	9EE2 4 SBC 3 SP1	
						9E63 6 COM 3 SP1							9ED3 5 CPX 4 SP2	9EE3 4 CPX 3 SP1	9EF3 6 CPHX 3 SP1
						9E64 6 LSR 3 SP1							9ED4 5 AND 4 SP2	9EE4 4 AND 3 SP1	
													9ED5 5 BIT 4 SP2	9EE5 4 BIT 3 SP1	
						9E66 6 ROR 3 SP1							9ED6 5 LDA 4 SP2	9EE6 4 LDA 3 SP1	
						9E67 6 ASR 3 SP1							9ED7 5 STA 4 SP2	9EE7 4 STA 3 SP1	
						9E68 6 LSL 3 SP1							9ED8 5 EOR 4 SP2	9EE8 4 EOR 3 SP1	
						9E69 6 ROL 3 SP1							9ED9 5 ADC 4 SP2	9EE9 4 ADC 3 SP1	
						9E6A 6 DEC 3 SP1							9EDA 5 ORA 4 SP2	9EEA 4 ORA 3 SP1	
						9E6B 8 DBNZ 4 SP1							9EDB 5 ADD 4 SP2	9EEB 4 ADD 3 SP1	
						9E6C 6 INC 3 SP1									
						9E6D 5 TST 3 SP1									

Table 9. Opcode map (Sheet 2 of 2)*...continued*

Bit-Manipulation		Branch	Read-Modify-Write				Control			Register/Memory					
										9EAE 5 LDHX 2 IX	9EBE 6 LDHX 4 IX2	9ECE 5 LDHX 3 IX1	9EDE 5 LDX 4 SP2	9EEE 4 LDX 3 SP1	9EFE 5 LDHX 3 SP1
						9E6F 6 CLR 3 SP1							9EDF 5 STX 4 SP2	9EEF 4 STX 3 SP1	9EFF 5 STHX 3 SP1

Table 9. Opcode map (Sheet 2 of 2)

INH	Inherent	REL	relative	SP1	Stack Pointer, 8-bit offset
IMM	Immediate	IX	Indexed, no offset	SP2	Stack Pointer, 16 bit offset
DIR	Direct	IX1	Indexed, 8-bit offset	IX+	Indexed, No offset with post increment
EXT	Extended	IX2	Indexed, 16 bit offset	IX1+	Indexed, 1 byte offset with post increment
DD	DIR to DIR	IMD	IMM to DIR		
IX+D	IX+ to DIR	DIX+	DIR to IX+		
Note: *All Sheet 2 Opcodes are preceded by the Page 2 Prebyte (9E)*					

Table 9. Opcode map (Sheet 2 of 2)

Prebyte (9E) and Opcode in Hexadecimal	9E60 6	HCS08 Cycles
	SUB	Instruction Mnemonic
Number of Bytes	3 SP1	Addressing Mode

9 Development support

9.1 Introduction

This chapter describes the single-wire BACKGROUND DEBUG mode (BDM), which uses the on-chip BACKGROUND DEBUG controller (BDC) module. Visit https://www.nxp.com/ to obtain additional user guides, application notes, and evaluation hardware collateral references.

9.1.1 Features

Features of the BDC module include:

- Single pin for mode selection and background communications
- BDC registers are not located in the memory map
- SYNC command to determine target communications rate
- Non-intrusive commands for memory access
- ACTIVE BACKGROUND mode commands for CPU register access
- GO and TRACE1 commands
- BACKGROUND command can wake CPU from STOP or WAIT modes
- One hardware address breakpoint built into BDC
- Oscillator runs in STOP mode, if BDC enabled
- COP watchdog disabled while in ACTIVE BACKGROUND mode

9.2 Background debug controller (BDC)

All MCUs in the HCS08 Family contain a single-wire BACKGROUND DEBUG interface that supports in-circuit programming of on-chip nonvolatile memory and sophisticated non-intrusive debug capabilities. Unlike debug interfaces on earlier 8-bit MCUs, this system does not interfere with normal application resources. It does not use any user memory or locations in the memory map and does not share any on-chip peripherals.

BDC commands are divided into two groups:

- ACTIVE BACKGROUND mode commands require that the target MCU is in ACTIVE BACKGROUND mode (the user program is not running). ACTIVE BACKGROUND mode commands allow the CPU registers to be read or written, and allow the user to trace one user instruction at a time, or GO to the user program from ACTIVE BACKGROUND mode.
- Non-intrusive commands can be executed at any time even while the user's program is running. Non-intrusive commands allow a user to read or write MCU memory locations or access status and control registers within the BACKGROUND DEBUG controller.

Typically, a relatively simple interface pod is used to translate commands from a host computer into commands for the custom serial interface to the single-wire BACKGROUND DEBUG system. Depending on the development tool vendor, this interface pod may use a standard RS-232 serial port, a parallel printer port, or some other type of communications such as a universal serial bus (USB) to communicate between the host PC and the pod. The pod typically connects to the target system with ground, the BKGD/PTA4 pin, $\overline{\text{RESET}}$, and sometimes V_{DD}. An open-drain connection to reset allows the host to force a target system reset, which is useful to regain control of a lost target system or to control startup of a target system before the on-chip nonvolatile memory has been programmed. Sometimes V_{DD} can be used to allow the pod to use power from the target system to avoid the need for a separate power supply. However, if the pod is powered separately, it can be connected to a running target system without forcing a target system reset or otherwise disturbing the running application program.

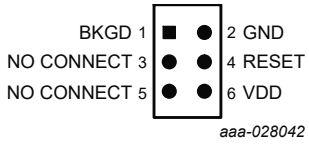


Figure 7. BDM tool connector

9.2.1 BKGD/PTA4 pin description

BKGD/PTA4 is the single-wire BACKGROUND DEBUG interface pin. The primary function of this pin is for bidirectional serial communication of ACTIVE BACKGROUND mode commands and data. During reset, this pin is used to select between starting in ACTIVE BACKGROUND mode or starting the user's application program. This pin is also used to request a timed sync response pulse to allow a host development tool to determine the correct clock frequency for BACKGROUND DEBUG serial communications.

BDC serial communications use a custom serial protocol first introduced on the M68HC12 Family of microcontrollers. This protocol assumes the host knows the communication clock rate that is determined by the target BDC clock rate. All communication is initiated and controlled by the host that drives a high-to-low edge to signal the beginning of each bit time. Commands and data are sent most significant bit first (MSB first). For a detailed description of the communications protocol, see Section 9.2.2 "Communication details".

If a host is attempting to communicate with a target MCU that has an unknown BDC clock rate, a SYNC command may be sent to the target MCU to request a timed sync response signal from which the host can determine the correct communication speed.

BKGD/PTA4 is a pseudo-open-drain pin and there is an on-chip pullup so no external pullup resistor is required. Unlike typical open-drain pins, the external RC time constant on this pin, which is influenced by external capacitance, plays almost no role in signal

rise time. The custom protocol provides for brief, actively driven speedup pulses to force rapid rise times on this pin without risking harmful drive level conflicts. See Figure 1 for more detail.

When no debugger pod is connected to the 6-pin BDM interface connector, the internal pullup on BKGD/PTA4 chooses normal operating mode. When a debug pod is connected to BKGD/PTA4, it is possible to force the MCU into ACTIVE BACKGROUND mode after reset. The specific conditions for forcing ACTIVE BACKGROUND depend upon the HCS08 derivative. See Section 9.1. It is not necessary to reset the target MCU to communicate with it through the BACKGROUND DEBUG interface.

9.2.2 Communication details

The BDC serial interface requires the external controller to generate a falling edge on the BKGD/PTA4 pin to indicate the start of each bit time. The external controller provides this falling edge whether data is transmitted or received.

BKGD/PTA4 is a pseudo-open-drain pin that can be driven either by an external controller or by the MCU. Data is transferred MSB first at 16 BDC clock cycles per bit (nominal speed). The interface times out if 512 BDC clock cycles occur between falling edges from the host. Any BDC command that was in progress when this timeout occurs is aborted without affecting the memory or operating mode of the target MCU system.

The custom serial protocol requires the debug pod to know the target BDC communication clock speed.

The clock switch (CLKSW) control bit in the BDC status and control register allows the user to select the BDC clock source. The BDC clock source can either be the bus or the alternate BDC clock source.

The BKGD/PTA4 pin can receive a high or low level or transmit a high or low level. The following diagrams show timing for each of these cases. Interface timing is synchronous to clocks in the target BDC, but asynchronous to the external host. The internal BDC clock signal is shown for reference in counting cycles.

Figure 8 shows an external host transmitting a logic 1 or 0 to the BKGD/PTA4 pin of a target HCS08 MCU. The host is asynchronous to the target so there is a 0-to-1 cycle delay from the host-generated falling edge to where the target perceives the beginning of the bit time. Ten target BDC clock cycles later, the target senses the bit level on the BKGD/PTA4 pin. Typically, the host actively drives the pseudo-open-drain BKGD/PTA4 pin during host-to-target transmissions to speed up rising edges. Because the target does not drive the BKGD/PTA4 pin during the host-to-target transmission period, there is no need to treat the line as an open-drain signal during this period.

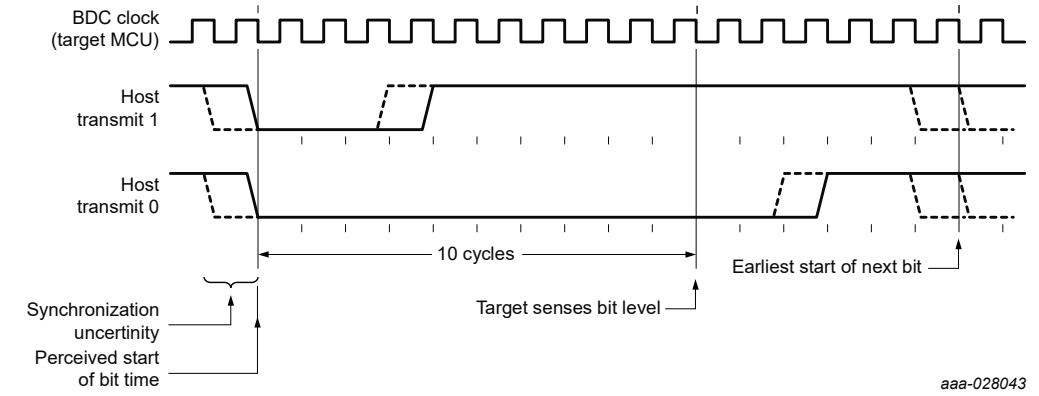


Figure 8. BDC host-to-target serial bit timing

Figure 9 shows the host receiving a logic 1 from the target HCS08 MCU. Because the host is asynchronous to the target MCU, there is a 0-to-1 cycle delay from the host-generated falling edge on BKGD/PTA4 to the perceived start of the bit time in the target MCU. The host holds the BKGD/PTA4 pin low long enough for the target to recognize it (at least two target BDC cycles). The host must release the low drive before the target MCU drives a brief active-high speedup pulse seven cycles after the perceived start of the bit time. The host should sample the bit level about 10 cycles after it started the bit time.

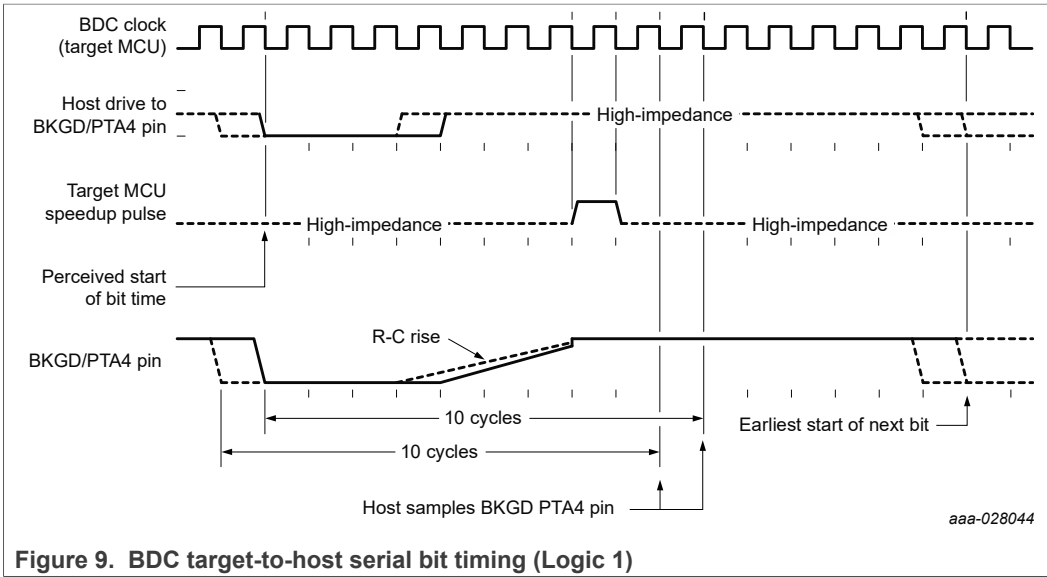


Figure 9. BDC target-to-host serial bit timing (Logic 1)

Figure 10 shows the host receiving a logic 0 from the target HCS08 MCU. Because the host is asynchronous to the target MCU, there is a 0-to-1 cycle delay from the host-generated falling edge on BKGD/PTA4 to the start of the bit time as perceived by the target MCU. The host initiates the bit time but the target HCS08 finishes it. Because the target wants the host to receive a logic 0, it drives the BKGD/PTA4 pin low for 13 BDC clock cycles, then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 cycles after starting the bit time.

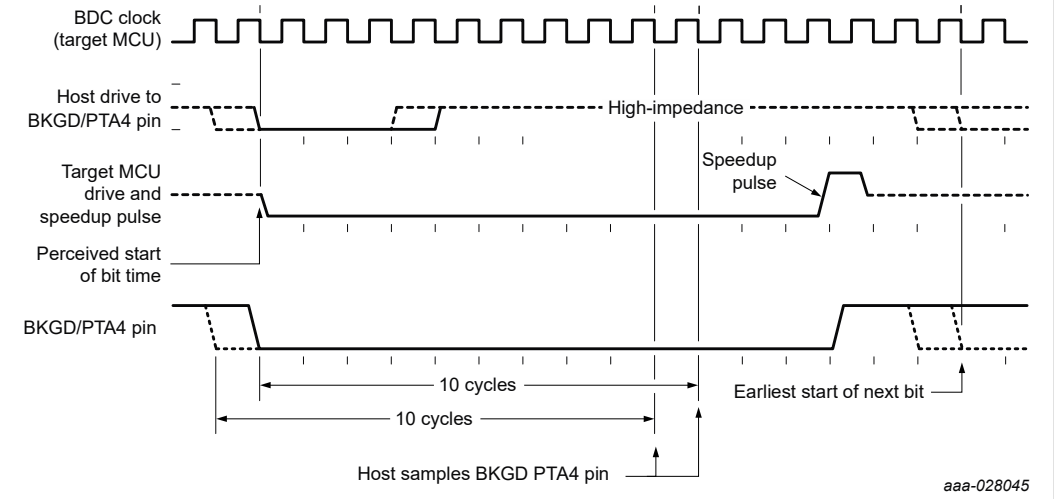


Figure 10. BDM target-to-host serial bit timing (Logic 0)

9.2.3 BDC commands

BDC commands are sent serially from a host computer to the BKGD/PTA4 pin of the target HCS08 MCU. All commands and data are sent MSB-first using a custom BDC communications protocol. ACTIVE BACKGROUND mode commands require that the target MCU is currently in the ACTIVE BACKGROUND mode while non-intrusive commands may be issued at any time whether the target MCU is in ACTIVE BACKGROUND mode or running a user application program. Table 10 shows all HCS08 BDC commands, a shorthand description of their coding structure, and the meaning of each command.

9.2.3.1 Coding structure nomenclature

This nomenclature is used in Table 10 to describe the coding structure of the BDC commands. Commands begin with an 8-bit hexadecimal command code in the host-to-target direction (most significant bit first).

/	=	separates parts of the command
d	=	delay 16 target BDC clock cycles
AAAA	=	a 16-bit address in the host-to-target direction
RD	=	8 bits of read data in the target-to-host direction
WD	=	8 bits of write data in the host-to-target direction
RD!6	=	16 bits of read data in the target-to-host direction
WD16	=	16 bits of write data in the host-to-target direction
SS	=	the contents of BDCSCR in the target-to-host direction (STATUS)
CC	=	8 bits of write data for BDCSCR in the host-to-target direction (CONTROL)
RBKP	=	16 bits of read data in the target-to-host direction (from BDCBKPT breakpoint register)
WBKP	=	16 bits of write data in the host-to-target direction (for BDCBKPT breakpoint register)

Table 10. BDC command summary

Command Mnemonic	Active BDM/ Non-intrusive	Coding Structure	Description
SYNC	Non-intrusive	n/a [1]	Request a timed reference pulse to determine target BDC communication speed
ACK_ENABLE	Non-intrusive	D5/d	Enable acknowledge protocol. Refer to NXP document order no. HCS08RMv1/D.
ACK_DISABLE	Non-intrusive	D6/d	Disable acknowledge protocol. Refer to NXP document order no. HCS08RMv1/D.
BACKGROUND	Non-intrusive	90/d	Enter ACTIVE BACKGROUND mode if enabled (ignore if ENBDM bit equals 0)
READ_STATUS	Non-intrusive	E4/SS	Read BDC status from BDCSCR
WRITE_CONTROL	Non-intrusive	C4/CC	Write BDC controls in BDCSCR
READ_BYTE	Non-intrusive	E0/AAAA/d/RD	Read a byte from target memory
READ_BYTE_WS	Non-intrusive	E1/AAAA /d/SS/RD	Read a byte and report status
READ_LAST	Non-intrusive	E8/SS/RD	Re-read byte from address just read and report status
WRITE_BYTE	Non-intrusive	C0/AAAA/WD/d	Write a byte to target memory
WRITE_BYTE_WS	Non-intrusive	C1/AAAA /WD/d/SS	Write a byte and report status
READ_BKPT	Non-intrusive	E2/RBKP	Read BDCBKPT breakpoint register
WRITE_BKPT	Non-intrusive	C2/WBKP	Write BDCBKPT breakpoint register
GO	Active BDM	08/d	Go to execute the user application program starting at the address currently in the PC
TRACE1	Active BDM	10/d	Trace 1 user instruction at the address in the PC, then return to ACTIVE BACKGROUND mode
TAGGO	Active BDM	18/d	Same as GO but enable external tagging (HCS08 devices have no external tagging pin)
READ_A	Active BDM	68/d/RD	Read accumulator (A)
READ_CCR	Active BDM	69/d/RD	Read condition code register (CCR)
READ_PC	Active BDM	6B/d/RD16	Read program counter (PC)
READ_HX	Active BDM	6C/d/RD16	Read H and X register pair (H:X)
READ_SP	Active BDM	6F/d/RD16	Read stack pointer (SP)
READ_NEXT	Active BDM	70/d/RD	Increment H:X by one then read memory byte located at H:X
READ_NEXT_WS	Active BDM	71/d/SS/RD	Increment H:X by one then read memory byte located at H:X. Report status and data.
WRITE_A	Active BDM	48/WD/d	Write accumulator (A)
WRITE_CCR	Active BDM	49/WD/d	Write condition code register (CCR)
WRITE_PC	Active BDM	4B/WD16/d	Write program counter (PC)
WRITE_HX	Active BDM	4C/WD16/d	Write H and X register pair (H:X)
WRITE_SP	Active BDM	4F/WD16/d	Write stack pointer (SP)

Table 10. BDC command summary*...continued*

Command Mnemonic	Active BDM/ Non-intrusive	Coding Structure	Description
WRITE_NEXT	Active BDM	50/WD/d	Increment H:X by one, then write memory byte located at H:X
WRITE_NEXT_WS	Active BDM	51/WD/d/SS	Increment H:X by one, then write memory byte located at H:X. Also report status.

[1] The SYNC command is a special operation that does not have a command code.

The SYNC command is unlike other BDC commands because the host does not necessarily know the correct communications speed to use for BDC communications until after it has analyzed the response to the SYNC command.

To issue a SYNC command, the host:

- Drives the BKGD/PTA4 pin low for at least 128 cycles of the slowest possible BDC clock (The slowest clock is normally the reference oscillator/64 or the self-clocked rate/64.)
- Drives BKGD/PTA4 high for a brief speedup pulse to get a fast rise time (This speedup pulse is typically one cycle of the fastest clock in the system.)
- Removes all drive to the BKGD/PTA4 pin so it reverts to high impedance
- Monitors the BKGD/PTA4 pin for the sync response pulse

The target, upon detecting the SYNC request from the host (which is a much longer low time than would ever occur during normal BDC communications):

- Waits for BKGD/PTA4 to return to a logic high
- Delays 16 cycles to allow the host to STOP driving the high speedup pulse
- Drives BKGD/PTA4 low for 128 BDC clock cycles
- Drives a 1-cycle high speedup pulse to force a fast rise time on BKGD/PTA4
- Removes all drive to the BKGD/PTA4 pin so it reverts to high impedance

The host measures the low time of this 128-cycle sync response pulse and determines the correct speed for subsequent BDC communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

9.2.4 BDC hardware breakpoint

The BDC includes one relatively simple hardware breakpoint that compares the CPU address bus to a 16-bit match value in the BDCBKPT register. This breakpoint can generate a forced breakpoint or a tagged breakpoint. A forced breakpoint causes the CPU to enter ACTIVE BACKGROUND mode at the first instruction boundary following any access to the breakpoint address. The tagged breakpoint causes the instruction opcode at the breakpoint address to be tagged so that the CPU enters ACTIVE BACKGROUND mode rather than executing that instruction if and when it reaches the end of the instruction queue. This implies that tagged breakpoints can only be placed at the address of an instruction opcode while forced breakpoints can be set at any address.

The breakpoint enable (BKPTEN) control bit in the BDC status and control register (BDCSCR) is used to enable the breakpoint logic (BKPTEN = 1). When BKPTEN = 0, its default value after reset, the breakpoint logic is disabled and no BDC breakpoints are requested regardless of the values in other BDC breakpoint registers and control bits.

The force/tag select (FTS) control bit in BDCSCR is used to select forced (FTS = 1) or tagged (FTS = 0) type breakpoints.

9.3 Register definition

This section contains the descriptions of the BDC registers and control bits.

This section refers to registers and control bits only by their names. A NXP-provided equate or header file is used to translate these names into the appropriate absolute addresses.

9.3.1 BDC registers and control bits

The BDC has two registers:

- The BDC status and control register (BDCSCR) is an 8-bit register containing control and status bits for the BACKGROUND DEBUG controller.
- The BDC breakpoint match register (BDCBKPT) holds a 16-bit breakpoint match address.

These registers are accessed with dedicated serial BDC commands and are not located in the memory space of the target MCU (so they do not have addresses and cannot be accessed by user programs).

Some of the bits in the BDCSCR have write limitations; otherwise, these registers may be read or written at any time. For example, the ENBDM control bit may not be written while the MCU is in ACTIVE BACKGROUND mode. (This prevents the ambiguous condition of the control bit forbidding ACTIVE BACKGROUND mode while the MCU is already in ACTIVE BACKGROUND mode.) Also, the four status bits (BDMACT, WS, WSF, and DVF) are read-only status indicators and can never be written by the WRITE_CONTROL serial BDC command. The clock switch (CLKSW) control bit may be read or written at any time.

9.3.2 BDC status and control register (BDCSCR)

This register can be read or written by serial BDC commands (READ_STATUS and WRITE_CONTROL) but is not accessible to user programs because it is not located in the normal memory map of the MCU.

Table 11. BDC status and control register (BDCSCR)

Bit	7	6	5	4	3	2	1	0
R	ENBDM	BDMACT	BKPTEN	FTS	CLKSW	WS	WSF	DVF
W		reserved				reserved	reserved	reserved
Normal Reset	0	0	0	0	0	0	0	0
Reset in Active BDM	1	1	0	0	1	0	0	0

Table 12. BDCSCR register field descriptions

Field	Description
7 ENBDM	Enable BDM (Permit ACTIVE BACKGROUND Mode) — Typically, this bit is written to 1 by the debug host shortly after the beginning of a debug session or whenever the debug host resets the target and remains 1 until a normal reset clears it. 0 BDM cannot be made active (non-intrusive commands still allowed) 1 BDM can be made active to allow ACTIVE BACKGROUND mode commands
6 BDMACT	BACKGROUND Mode Active Status — This is a read-only status bit. 0 BDM not active (user application program running) 1 BDM active and waiting for serial commands
5 BKPTEN	BDC Breakpoint Enable — If this bit is clear, the BDC breakpoint is disabled and the FTS (force tag select) control bit and BDCBKPT match register are ignored. 0 BDC breakpoint disabled 1 BDC breakpoint enabled
4 FTS	Force/Tag Select — When FTS = 1, a breakpoint is requested whenever the CPU address bus matches the BDCBKPT match register. When FTS = 0, a match between the CPU address bus and the BDCBKPT register causes the fetched opcode to be tagged. If this tagged opcode ever reaches the end of the instruction queue, the CPU enters ACTIVE BACKGROUND mode rather than executing the tagged opcode. 0 Tag opcode at breakpoint address and enter ACTIVE BACKGROUND mode if CPU attempts to execute that instruction 1 Breakpoint match forces ACTIVE BACKGROUND mode at next instruction boundary (address need not be an opcode)
3 CLKSW	Select Source for BDC Communications Clock — CLKSW defaults to 0, which selects the alternate BDC clock source. 0 Alternate BDC clock source 1 MCU bus clock
2 WS	WAIT or STOP Status — When the target CPU is in WAIT or STOP mode, most BDC commands cannot function. However, the BACKGROUND command can be used to force the target CPU out of WAIT or STOP and into ACTIVE BACKGROUND mode where all BDC commands work. Whenever the host forces the target MCU into ACTIVE BACKGROUND mode, the host should issue a READ_STATUS command to check that BDMACT = 1 before attempting other BDC commands. 0 Target CPU is running user application code or in ACTIVE BACKGROUND mode (was not in WAIT or STOP mode when BACKGROUND became active) 1 Target CPU is in WAIT or STOP mode, or a BACKGROUND command was used to change from WAIT or STOP to ACTIVE BACKGROUND mode
1 WSF	WAIT or STOP Failure Status — This status bit is set if a memory access command failed due to the target CPU executing a WAIT or STOP instruction at or about the same time. The usual recovery strategy is to issue a BACKGROUND command to get out of WAIT or STOP mode into ACTIVE BACKGROUND mode, repeat the command that failed, then return to the user program. (Typically, the host would restore CPU registers and stack values and re-execute the WAIT or STOP instruction.) 0 Memory access did not conflict with a WAIT or STOP instruction 1 Memory access command failed because the CPU entered WAIT or STOP mode
0 DVF	Data Valid Failure Status — This status bit is not used in the MC9S08RA16 because it does not have any slow access memory. 0 Memory access did not conflict with a slow memory access 1 Memory access command failed because CPU was not finished with a slow memory access

9.3.3 BDC breakpoint match register (BDCBKPT)

This 16-bit register holds the address for the hardware breakpoint in the BDC. The BKPTEN and FTS control bits in BDCSCR are used to enable and configure the

breakpoint logic. Dedicated serial BDC commands (READ_BKPT and WRITE_BKPT) are used to read and write the BDCBKPT register but is not accessible to user programs because it is not located in the normal memory map of the MCU. Breakpoints are normally set while the target MCU is in ACTIVE BACKGROUND mode before running the user application program. For additional information about setup and use of the hardware breakpoint logic in the BDC, see Section 9.2.4 "BDC hardware breakpoint".

9.3.4 System background debug force reset register (SBDFR)

This register contains a single write-only control bit. A serial BACKGROUND mode command such as WRITE_BYTE must be used to write to SBDFR. Attempts to write this register from a user program are ignored. Reads always return 0x00.

Table 13. System background debug force reset register (SBDFR)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	reserved	reserved	reserved	reserved	reserved	reserved	reserved	BDFR[1]
Reset	0	0	0	0	0	0	0	0

[1] BDFR is writable only through serial BACKGROUND mode debug commands, not from user programs.

Table 14. SBDFR register field description

Field	Description
0 BDFR	Background Debug Force Reset — A serial ACTIVE BACKGROUND mode command such as WRITE_BYTE allows an external debug host to force a target system reset. Writing 1 to this bit forces an MCU reset. This bit cannot be written from a user program.

10 Functional description

10.1 Register information

10.1.1 Register map

Table 15. Register map description

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$0000	PTAD	reserved	reserved	reserved	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
$0001	PTAPE	—	—	—	—	PTAPE3	PTAPE2	PTAPE1	PTAPE0
$0002	reserved	—	—	—	—	—	—	—	—
$0003	PTADD	—	—	—	—	PTADD3	PTADD2	PTADD1	PTADD0
$0004	PTBD	—	—	—	—	—	—	PTBD1	PTBD0
$0005	PTBPE	—	—	—	—	—	—	PTBPE1	PTBPE0
$0006	SPARE06	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0007	PTBDD	—	—	—	—	—	—	PTBDD1	PTBDD0
$0008	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0009	SPARE09	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$000A:B	reserved	—	—	—	—	—	—	—	—
$000C	KBISC	—	—	—	—	KBF	KBACK	KBIE	KBIMOD

Table 15. Register map description*...continued*

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$000D	KBIPE	—	—	—	—	KBIPE3	KBIPE2	KBIPE1	KBIPE0
$000E	KBIES	—	—	—	—	KBEDG3	KBEDG2	KBEDG1	KBEDG0
$000F	IRQSC	—	IRQPDD	IRQEDG	IRQPE	IRQF	IRQACK	IRQIE	IRQMOD
$0010	TPMSC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
$0011	TPMCNTH	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
$0012	TPMCNTL	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$0013	TPMMODH	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
$0014	TPMMODL	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$0015	TPMC0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	—	—
$0016	TPMC0VH	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
$0017	TPMC0VL	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$0018	TPMC1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	—	—
$0019	TPMC1VH	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
$001A	TPMC1VL	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$001B	PWUSR	WUF	WUFACK	PSEL	PRF	PRFACK	—	—	—
$001C	PWUDIV	WDIV7	WDIV6	WDIV5	WDIV4	WDIV3	WDIV2	WDIV1	WDIV0
$001D	PWUCS0	WUT7	WUT6	WUT5	WUT4	WUT3	WUT2	WUT1	WUT0
$001E	PWUCS1	PRST7	PRST6	PRST5	PRST4	PRST3	PRST2	PRST1	PRST0
$001F	PWUS	CSTAT7	CSTAT6	CSTAT5	CSTAT4	CSTAT3	CSTAT2	CSTAT1	CSTAT0
$0020	LFCTL1	LFEN	SRES	CARMOD	—	IDSEL1	IDSEL0	SENS1	SENS0
$0021	LFCTL2	LFSTM3	LFSTM2	LFSTM1	LFSTM0	LFONTM3	LFONTM2	LFONTM1	LFONTM0
$0022	LFCTL3	LFDO	TOGMOD	SYNC1	SYNC0	LFCDTM3	LFCDTM2	LFCDTM1	LFCDTM0
$0023	LFCTL4	LFDRIE	LFERIE	LFCDIE	LFIDIE	DCEN	VALEN	TIMOUT1	TIMOUT0
$0024	LFS	LFDRF	LFERF	LFCDF	LFIDF	LFOVF	LFEOMF	LPSM	LFIAK
$0025	LFDATA	LFRXD7	LFRXD6	LFRXD5	LFRXD4	LFRXD3	LFRXD2	LFRXD1	LFRXD0
$0026	LFIDL	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
$0027	LFIDH	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
$0028	LFCTRLE	reserved	reserved	reserved	reserved	TRIMEE	AZSC2	AZSC1	AZSC0
$0029	LFCTRLD	AVFOF1	AVFOF0	DEQS	AZDC1	AZDC0	ONMODE	CH125K1	CH125K0
$002A	LFCTRLC	AMPGAIN1	AMPGAIN0	FINSEL1	FINSEL0	AZEN	LOWQ1	LOWQ0	DEQEN
$002B	LFCTRLB	HYST1	HYST0	LFFAF	LFCAF	LFPOL	LFCPTAZ2	LFCPTAZ1	LFCPTAZ0
$002C	LFCTRLA	reserved	reserved	reserved	reserved	LFCC3	LFCC2	LFCC1	LFCC0
$002D	TRIM1	TRIMLFRO 3	TRIMLFRO 2	TRIMLFRO 1	TRIMLFRO 0	TRIMDET3	TRIMDET2	TRIMDET1	TRIMDET0
$002E	TRIM2	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$002F	LFRMCUAS CANDATA	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0030	ADSC1	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
$0031	ADSC2	ADACT	ADTRG	ACFE	ACFGT	—	—	REFSEL1	REFSEL0
$0032	ADRH	—	—	—	—	ADR11	ADR910	ADR9	ADR8
$0033	ADRL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0

Table 15. Register map description*...continued*

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$0034	ADCVH	—	—	—	—	ADCV11	ADCV10	ADCV9	ADCV8
$0035	ADCVL	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
$0036	ADCFG	ADLPC	ADIV1	ADIV0	ADLSMP	MODE1	MODE0	ADICLK1	ADICLK0
$0037	ADPCTL1	—	—	—	ADPC4	ADPC3	—	—	—
$0038	SPIOPS	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0039	SPITESTEN	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$003A	PADCONFIG	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$003B	DTBOUTSEL0	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$003C	DTBOUTSEL1	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$003D	DTBSEL0	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$003E	DTBSEL1	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$003F	SPIDFTCTRL	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0040	SMICS	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0041	SMIC	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0042	SMICFG	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0043	SMIST	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0044	SMITM	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0045	SMITRIM0	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0046	SMITRIM1	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0047	SMITRIM2	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0048	SMITRIM3	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$0049	SMITRIM4	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$004A	SMITRIM5	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$004B	SMITRIM6	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$004C	SMITRIM7	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$004D:$004F	reserved	—	—	—	—	—	—	—	—
$0050:$006F	PARAM0:PARAM31	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$0070:$008F	PARAM32:PARAM63	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$0090:$028F	RAM0:RAM511	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$0800:$17FF	FLS_ADDR0:FLS_ADDR4095	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1800	SIMRS	POR	PIN	COP	ILOP	ILAD	PWU	LVR	SOFT
$1801	SIMC	—	—	—	—	—	—	—	BDFR
$1802	SIMOPT1	COPE	COPCLKS	STOPE	RFEN	—	SPIEN	BKGDPE	—

Table 15. Register map description*...continued*

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$1803	SIMOPT2	—	COPT2	COPT1	COPT0	LFOSEL	TCLKDIV	BUSCLKS1	BUSCLKS0
$1804	SIMTCSC	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1805	SIMCO	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1806	SIMPID1	REV3	REV2	REV1	REV0	0	0	0	0
$1807	SIMPID2	0	0	1	0	1	1	0	0
$1808	SRTISC (PMCRSC)	RTIF	RTIACK	RTICLKS	RTIE	—	RTIS2	RTIS1	RTIS0
$1809	SPMSC1 (PMCSC1)	LVDF	LVDACK	LVDIE	LVDRE	LVDSE	LVDE	reserved	BGBE
$180A	SPMSC2 (PMCSC2)	—	—	—	PDF	—	PPDACK	PDC	—
$180B	PMCT(1)	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$180C	PMCSC3	LVWF	LVWACK	LVDV	LVWV	reserved	reserved	—	reserved
$180D	SIMSES	—	—	KBF	IRQF	FRCF	PWUF	LFF	RFF
$180E	SIMOTRM	SOTRM7	SOTRM6	SOTRM5	SOTRM4	SOTRM3	SOTRM2	SOTRM1	SOTRM0
$180F	SIMTEST	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1810:$181F	reserved	—	—	—	—	—	—	—	—
$1820	FCDIV	DIVLD	PRDIV8	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
$1821	FOPT	KEYEN	FNORED	—	—	—	—	SEC1	SEC0
$1822	FTSTMOD	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1823	FCNFG	—	—	KEYACC	—	—	—	—	—
$1824	FPROT	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
$1825	FSTAT	FCBEF	FCCF	FPVIOL	FACCERR	—	FBLANK	FFAIL	FDONE
$1826	FCMD	FTMR	FCMDB6	FCMDB5	FCMDB4	FCMDB3	FCMDB2	FCMDB1	FCMDB0
$1827	FCTL(1)	FERASE	FPROG	FIFREN	FNVSTR	FXE	FYE	FSE	FMAS1
$1828	FADDRHI(1)	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1829	FADDRLO(1)	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$182A	reserved	—	—	—	—	—	—	—	—
$182B	FDATA(1)	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$182C:$182F	reserved	—	—	—	—	—	—	—	—
$1830	RFCR0	BPS7	BPS6	BPS5	BPS4	BPS3	BPS2	BPS1	BPS0
$1831	RFCR1	FRM7	FRM6	FRM5	FRM4	FRM3	FRM2	FRM1	FRM0
$1832	RFCR2	SEND	reserved	reserved	PWR4	PWR3	PWR2	PWR1	PWR0
$1833	RFCR3	DATA	IFPD	ISPC	IFID	FNUM3	FNUM2	FNUM1	FNUM0
$1834	RFCR4	RFBT7	RFBT6	RFBT5	RFBT4	RFBT3	RFBT2	RFBT1	RFBT0
$1835	RFCR5	BOOST	LFSR6	LFSR5	LFSR4	LFSR3	LFSR2	LFSR1	LFSR0
$1836	RFCR6	VCO_GAIN1	VCO_GAIN0	RFFT5	RFFT4	RFFT3	RFFT2	RFFT1	RFFT0
$1837	RFCR7	RFIF	RFEF	RFVF	RFIAK	RFIEN	RFLVDEN	RCTS	RFMRST

Table 15. Register map description*...continued*

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$1838	PLLCR0	AFREQ12	AFREQ11	AFREQ10	AFREQ9	AFREQ8	AFREQ7	AFREQ6	AFREQ5
$1839	PLLCR1	AFREQ4	AFREQ3	AFREQ2	AFREQ1	AFREQ0	POL	CODE1	CODE0
$183A	PLLCR2	BFREQ12	BFREQ11	BFREQ10	BFREQ9	BFREQ8	BFREQ7	BFREQ6	BFREQ5
$183B	PLLCR3	BFREQ4	BFREQ3	BFREQ2	BFREQ1	BFREQ0	CF	MOD	CKREF
$183C	RFTX0	RFTXD7	RFTXD6	RFTXD5	RFTXD4	RFTXD3	RFTXD2	RFTXD1	RFTXD0
$183D	RFTX1	RFTXD15	RFTXD14	RFTXD13	RFTXD12	RFTXD11	RFTXD10	RFTXD9	RFTXD8
$183E	RFTX2	RFTXD23	RFTXD22	RFTXD21	RFTXD20	RFTXD19	RFTXD18	RFTXD17	RFTXD16
$183F	RFTX3	RFTXD31	RFTXD30	RFTXD29	RFTXD28	RFTXD27	RFTXD26	RFTXD25	RFTXD24
$1840	RFTX4	RFTXD39	RFTXD38	RFTXD37	RFTXD36	RFTXD35	RFTXD34	RFTXD33	RFTXD32
$1841	RFTX5	RFTXD47	RFTXD46	RFTXD45	RFTXD44	RFTXD43	RFTXD42	RFTXD41	RFTXD40
$1842	RFTX6	RFTXD55	RFTXD54	RFTXD53	RFTXD52	RFTXD51	RFTXD50	RFTXD49	RFTXD48
$1843	RFTX7	RFTXD63	RFTXD62	RFTXD61	RFTXD60	RFTXD59	RFTXD58	RFTXD57	RFTXD56
$1844	RFTX8	RFTXD71	RFTXD70	RFTXD69	RFTXD68	RFTXD67	RFTXD66	RFTXD65	RFTXD64
$1845	RFTX9	RFTXD79	RFTXD78	RFTXD77	RFTXD76	RFTXD75	RFTXD74	RFTXD73	RFTXD72
$1846	RFTX10	RFTXD87	RFTXD86	RFTXD85	RFTXD84	RFTXD83	RFTXD82	RFTXD81	RFTXD80
$1847	RFTX11	RFTXD95	RFTXD94	RFTXD93	RFTXD92	RFTXD91	RFTXD90	RFTXD89	RFTXD88
$1848	RFTX12	RFTXD103	RFTXD102	RFTXD101	RFTXD100	RFTXD99	RFTXD98	RFTXD97	RFTXD96
$1849	RFTX13	RFTXD111	RFTXD110	RFTXD109	RFTXD108	RFTXD107	RFTXD106	RFTXD105	RFTXD104
$184A	RFTX14	RFTXD119	RFTXD118	RFTXD117	RFTXD116	RFTXD115	RFTXD114	RFTXD113	RFTXD112
$184B	RFTX15	RFTXD127	RFTXD126	RFTXD125	RFTXD124	RFTXD123	RFTXD122	RFTXD121	RFTXD120
$184C	RFTX16	RFTXD135	RFTXD134	RFTXD133	RFTXD132	RFTXD131	RFTXD130	RFTXD129	RFTXD128
$184D	RFTX17	RFTXD143	RFTXD142	RFTXD141	RFTXD140	RFTXD139	RFTXD138	RFTXD137	RFTXD136
$184E	RFTX18	RFTXD151	RFTXD150	RFTXD149	RFTXD148	RFTXD147	RFTXD146	RFTXD145	RFTXD144
$184F	RFTX19	RFTXD159	RFTXD158	RFTXD157	RFTXD156	RFTXD155	RFTXD154	RFTXD153	RFTXD152
$1850	RFTX20	RFTXD167	RFTXD166	RFTXD165	RFTXD164	RFTXD163	RFTXD162	RFTXD161	RFTXD160
$1851	RFTX21	RFTXD175	RFTXD174	RFTXD173	RFTXD172	RFTXD171	RFTXD170	RFTXD169	RFTXD168
$1852	RFTX22	RFTXD183	RFTXD182	RFTXD181	RFTXD180	RFTXD179	RFTXD178	RFTXD177	RFTXD176
$1853	RFTX23	RFTXD191	RFTXD190	RFTXD189	RFTXD188	RFTXD187	RFTXD186	RFTXD185	RFTXD184
$1854	RFTX24	RFTXD199	RFTXD198	RFTXD197	RFTXD196	RFTXD195	RFTXD194	RFTXD193	RFTXD192
$1855	RFTX25	RFTXD207	RFTXD206	RFTXD205	RFTXD204	RFTXD203	RFTXD202	RFTXD201	RFTXD200
$1856	RFTX26	RFTXD215	RFTXD214	RFTXD213	RFTXD212	RFTXD211	RFTXD210	RFTXD209	RFTXD208
$1857	RFTX27	RFTXD223	RFTXD222	RFTXD221	RFTXD220	RFTXD219	RFTXD218	RFTXD217	RFTXD216
$1858	RFTX28	RFTXD231	RFTXD230	RFTXD229	RFTXD228	RFTXD227	RFTXD226	RFTXD225	RFTXD224
$1859	RFTX29	RFTXD239	RFTXD238	RFTXD237	RFTXD236	RFTXD235	RFTXD234	RFTXD233	RFTXD232
$185A	RFTX30	RFTXD247	RFTXD246	RFTXD245	RFTXD244	RFTXD243	RFTXD242	RFTXD241	RFTXD240
$185B	RFTX31	RFTXD255	RFTXD254	RFTXD253	RFTXD252	RFTXD251	RFTXD250	RFTXD249	RFTXD248
$185C	IBEN	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$185D	VCAL	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$185E	RFTEST	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$185F	ASCANSHIFTINOUT	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved

Table 15. Register map description*...continued*

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$1860	EPR	EOM	PLL_LPF_2	PLL_LPF_1	PLL_LPF_0	EPR	DRBP	PA_SLOPE1	PA_SLOPE0
$1861	RFPRECHARGE	TIMEOUT1	TIMEOUT0	reserved	reserved	ENAREGCOMP	AREGPC	AREGOK	reserved
$1862	RFRW1	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1863	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1864	RFATRIM1	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1865	RFATRIM2	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1866	RFATRIM3	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1867	RFMMCUASCAN	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1868	MCUASCANDATA	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1869:$186F	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1870	PMCTMCR1	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1871	PMCTRIM1	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1872	PMCTRIM2	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1873	PMCTMCR2	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1874	PMCSR	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1875	PMCATB1	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1876	PMCATB0	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
$1877:$187F	reserved	—	—	—	—	—	—	—	—
$1880	FRCCR	FRC_CLR	—	FRC_EN_HALT	FRC_COMP_EN	FRC_COMP_IACK	FRC_IF	—	—
$1881	FRCTIMERH	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
$1882	FRCTIMERL	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$1883	FRCCOMP2	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
$1884	FRCCOMP1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$1885:$188F	reserved	—	—	—	—	—	—	—	—
$FD40:$FDFA	COEFFICIENTS	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$FDFB	CODEH	MCU1	MCU0	PRESS1	PRESS0	ACCEL3	ACCEL2	ACCEL2	ACCEL0
$FDFC	CODE2	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
$FDFD	CODE3	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
$FDFE	CODE4	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
$FDFF	CODE5	ID31	ID30	ID29	ID28	ID27	ID26	ID25	ID24

Table 15. Register map description*...continued*

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$FFAC	CODEF	FWID7	FWID6	FWID5	FWID4	FWID3	FWID2	FWID1	FWID0
$FFAD	TargetID0								
$FFAE	TargetID1								
$FFAF	spare								
$FFB0	NVBACKKEY0	KEY7	KEY6	KEY5	KEY4	KEY3	KEY2	KEY1	KEY0
$FFB1	NVBACKKEY1	KEY15	KEY14	KEY13	KEY12	KEY11	KEY10	KEY9	KEY8
$FFB2	NVBACKKEY2	KEY23	KEY22	KEY21	KEY20	KEY19	KEY18	KEY17	KEY16
$FFB3	NVBACKKEY3	KEY31	KEY30	KEY29	KEY28	KEY27	KEY26	KEY25	KEY24
$FFB4	NVBACKKEY4	KEY39	KEY38	KEY37	KEY36	KEY35	KEY34	KEY33	KEY32
$FFB5	NVBACKKEY5	KEY47	KEY46	KEY45	KEY44	KEY43	KEY42	KEY41	KEY40
$FFB6	NVBACKKEY6	KEY55	KEY54	KEY53	KEY52	KEY51	KEY50	KEY49	KEY48
$FFB7	NVBACKKEY7	KEY63	KEY62	KEY61	KEY60	KEY59	KEY58	KEY57	KEY56
$FFB8:$FFBC	reserved	—	—	—	—	—	—	—	—
$FFBD	NVPROT	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
$FFBE	reserved	—	—	—	—	—	—	—	—
$FFBF	NVOPT	KEYEN	FNORED	—	—	—	—	SEC01	SEC00
$FFC0:$FFDF	reserved	—	—	—	—	—	—	—	—
$FFE0	Keyboard Int. High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFE1	Keyboard Int. Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFE2	FRC Int. High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFE3	FRC Int. Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFE4	reserved	—	—	—	—	—	—	—	—
$FFE5	reserved	—	—	—	—	—	—	—	—
$FFE6	RTI High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFE7	RTI Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFE8	LF RX Int. High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFE9	LF RX Int. Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFEA	ADC Int. High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFEB	ADC Int. Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0

Table 15. Register map description*...continued*

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$FFEC	RF TX Int. High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFED	RF TX Int. Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFEE	SMI Int. High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFEF	SMI Int. Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFF0	TPM OVF Int. High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFF1	TPM OVF Int. Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFF2	TPM Ch1 Int. High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFF3	TPM Ch1 Int. Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFF4	TPM Ch0 Int. High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFF5	TPM Ch0 Int. Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFF6	WU Int. High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFF7	WU Int. Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFF8	LVD Int. High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFF9	LVD Int. Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFFA	IRQ Int. High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFFB	IRQ Int. Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFFC	SWI High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFFD	SWI Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0
$FFFE	POR et al High	addr15	addr14	addr13	addr12	addr11	addr10	addr9	addr8
$FFFF	POR et al Low	addr7	addr6	addr5	addr4	addr3	addr2	addr1	addr0

10.1.2 Register description format

Table 16 depicts an example of the encoding used throughout this document to describe the registers within each functional block.

Table 16. Register description format

Address		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$xxxx	R W	Read function Write function	RW function	R function —	0 Write function	rwm	slfclr	0 w1c	
POR	$00	0	0	0	0	0	0	0	0
Other resets		U							

Where:

$xxxx = 16-bit address of the register

POR = true power-on reset result, after the power has been applied.

Other resets = the result of resets that occur while power remains applied, such as low-power-mode exits, low-voltage detection, illegal operations, enabling a function block, etc.

U = the state of the bit remains unaffected by the type of reset mentioned in the leftmost column.

Read function = the functional name of a readable bit within the register, appearing in the columns to the right

Write function = the functional name of a writable bit within the register, appearing in the columns to the right

RW function = the functional name of a bit that is both readable and writable

— = a readable bit that is not writable, meaning writes to the bit will have no reaction.

rwm = a read/write bit modified by hardware in some fashion other than by a reset.

slfclr = a self-clearing bit; writing a one has an effect, but the bit always reads as a zero.

w1c = a write-once-to-clear bit; a status bit that can be read, and is cleared by a writing a one.

0 or 1 = the result of a read, write, or reset; 0 meaning clear(ed) / de-asserted / de-activated; 1 meaning set / asserted / activated.

10.2 Interrupts

Interrupts provide a way to save the current CPU status and registers, execute an interrupt service routine (ISR), and then restore the CPU status so processing resumes where it left off before the interrupt. Other than the software interrupt (SWI), which is a program instruction, interrupts are caused by hardware events. The debug module can also generate an SWI under certain circumstances.

If an event occurs in an enabled interrupt source, an associated read-only status flag will become set. The CPU will not respond until and unless the local interrupt enable is a logic 1 to enable the interrupt. The I bit in the CCR must be a logic 0 to allow interrupts. The global interrupt mask (I bit) in the CCR is initially set after reset which masks (prevents) all maskable interrupt sources. The user program initializes the stack pointer and performs other system setup before clearing the I bit to allow the CPU to respond to interrupts. When the CPU receives a qualified interrupt request, it completes the current instruction before responding to the interrupt. The interrupt sequence follows the same cycle-by-cycle sequence as the SWI instruction and consists of:

- Saving the CPU registers on the stack
- Setting the I bit in the CCR to mask further interrupts
- Fetching the interrupt vector for the highest-priority interrupt that is currently pending
- Filling the instruction queue with the first three bytes of program information starting from the address fetched from the interrupt vector locations

While the CPU is responding to the interrupt, the I bit is automatically set to avoid the possibility of another interrupt interrupting the ISR itself (this is called nesting of interrupts). Normally, the I bit is restored to 0 when the CCR is restored from the value stacked on entry to the ISR. In rare cases, the I bit may be cleared inside an ISR (after clearing the status flag that generated the interrupt) so that other interrupts can be serviced without waiting for the first service routine to finish. This practice is not recommended for anyone other than the most experienced programmers because it can lead to subtle program errors that are difficult to debug.

The interrupt service routine ends with a return-from-interrupt (RTI) instruction which restores the CCR, A, X, and PC registers to their pre interrupt values by reading the previously saved information off the stack.

When two or more interrupts are pending when the I bit is cleared, the highest priority source is serviced first.

For compatibility with the M68HC08, the H register is not automatically saved and restored. It is good programming practice to push H onto the stack at the start of the interrupt service routine (ISR) and restore it just before the RTI that is used to return from the ISR.

10.2.1 Interrupt stack frame

Figure 11 shows the contents and organization of a stack frame. Before the interrupt, the stack pointer (SP) points at the next available byte location on the stack. The current values of CPU registers are stored on the stack starting with the low-order byte of the program counter (PCL) and ending with the CCR. After stacking, the SP points at the next available location on the stack which is the address that is one less than the address where the CCR was saved. The PC value that is stacked is the address of the instruction in the main program that would have executed next if the interrupt had not occurred.

When an RTI instruction is executed, these values are recovered from the stack in reverse order. As part of the RTI sequence, the CPU fills the instruction pipeline by reading three bytes of program information, starting from the PC address just recovered from the stack.

The status flag causing the interrupt must be acknowledged (cleared) before returning from the ISR. Typically, the flag should be cleared at the beginning of the ISR so that if another interrupt is generated by this same source, it will be registered so it can be serviced after completion of the current ISR.

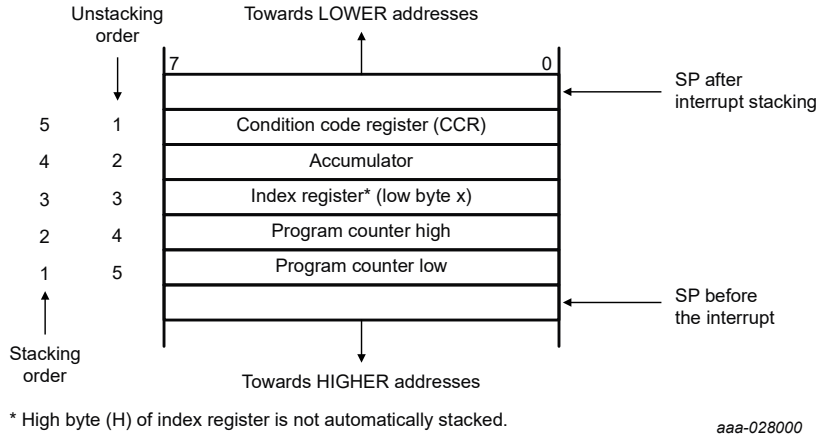


Figure 11. Interrupt stack frame

10.2.2 Vector summary

Table 17 provides a summary of all interrupt sources. Higher-priority sources are located toward the bottom of the table (at the higher vector addresses). All of these vectors are a 2-byte address that the firmware uses as the destination address. This allows the firmware to intercept all vectors and add additional processing as needed. The additional process latency for each interrupt is described in the corresponding firmware user guide.

Therefore, the high-order byte of the address for the user's interrupt service routine is located at the lower address in the vector address column, and the low-order byte of the address for the interrupt service routine is located at the higher address. When an interrupt condition occurs, an associated flag bit becomes set. If the associated local interrupt enable is set, an interrupt request is sent to the CPU. Within the CPU, if the global interrupt mask (I bit in the CCR) is 0, the CPU will finish the current instruction, stack the PCL, PCH, X, A, and CCR CPU registers, set the I bit, and then fetch the interrupt vector for the highest priority pending interrupt. Processing then continues in the interrupt service routine.

The triggering of any of these vector fetches wakes the MCU from any of the STOP modes.

10.3 Interrupt service routines

Interrupt service routines are managed by NXP firmware unless erased and overwritten by customer applications. This section describes the management of hardware vectors to user application vectors.

Each hardware vector is accessed when the prioritized interrupt is recognized. An interrupt service routine (ISR) clears the interrupt and sets appropriate flags for the user to poll, and, when appropriate, jumps to the assigned user vector as described in Table 17.

Table 17. Interrupt service routines

Vector priority	Hardware address	Vector name	Module source	Flag name	Enable name	Description
15	$FFE0 - $FFE1	vkbi	KBI	KBF	KBIE	Keyboard pin edge / level applied
14	$FFE2 - $FFE3	Vfrc	FRC	FRCF_IF	FRC_ COMP_ EN	Free running counter timer and comparison matched.
13	$FFE4 - $FFE5	—	—	—	—	not assigned
12	$FFE6 - $FFE7	Vrti	PMC	RTIF	RTIE	Real-time interrupt timer expiration if not in Stop 1
11	$FFE8 - $FFE9	Vlfrcvr	LFR	LFIDF LFCDF LFERF LFDRF	LFIDE LFCDIE LFERIE LFDRIE	LF receiver valid ID reception in data mode LF receiver carrier detection in carrier mode LF receiver error detection in Manchester decode mode LF receiver 8-bits data received in Manchester decode mode
10	$FFEA - $FFEB	Vadc	ADC	COCO	AIEN	ADC conversion completed
9	$FFEC - $FFED	Vrf	RFM	RFIF RFEF RFVF	RFIEN	RF transmitter x-bits data transmitted RF transmitter error detection RF transmitter low voltage detection
8	$FFEE - $FFEF	Vsmi	SMI	SMIF	SMIIE	Sensor Measurement Interface sequence completed
7	$FFF0 - $FFF1	Vtpm1ovf	TPM	TOF	TOIE	TPM timer overflow
6	$FFF2 - $FFF3	Vtpm1ch1	TPM	CH1F	CH1IE	TPM channel 1 event occurrence
5	$FFF4 - $FFF5	Vtpm1ch0	TPM	CH0F	CH0IE	TPM channel 0 event occurrence
4	$FFF6 - $FFF7	Vwuktmr	PWU	WUF	WUT[7:0]	PWU wake-up timer interval elapsed
3	$FFF8 - $FFF9	Vlvd	PMC	LVDF	LVDIE	PMC supply below LVD warning threshold detection
2	$FFFA - $FFFB	Virq	IRQ	IRQF	IRQE	External PTA0 pin edge / level applied
1	$FFFC - $FFFD	Vswi	CPU	—	—	SWI instruction executed
0	$FFF E- $FFFF	Vreset	SIM SIM SIM SIM SIM SIM SIM PMC PWU	POR PIN COP ILOP ILAD PWU SOFT LVR PRF	— — COPE — — — — LVDRE PRST[7:0]	Power-On Reset (POR) initialization sequence completed External RST _B pin falling edge applied COP watchdog timer expired without service Illegal opcode detected Illegal address detected PWU reset initialization sequence completed Soft reset detected PMC supply below LVR reset threshold detection PWU reset interval timer expired

10.4 Low-Voltage Detect (LVD) System

The NTM88 includes a system to detect low voltage conditions in order to protect memory contents and control MCU system states during supply voltage variations. The system is comprised of a power-on reset (POR) circuit and an LVD circuit with a user

selectable trip voltage, either high (V_{LVDH}) or low (V_{LVDL}). The LVD circuit is enabled when LVDE in SPMSC1 is high and the trip voltage is selected by LVDV in SPMSC3. The LVD is disabled upon entering any of the STOP modes unless the LVDSE bit is set. If LVDSE and LVDE are both set, then the MCU cannot enter STOP1.

10.4.1 Power-on reset operation

When power is initially applied to the NTM88, or when the supply voltage drops below the V_{POR} level, the POR circuit causes a reset condition. As the supply voltage rises, the LVD circuit holds the chip in reset until the supply has risen above the level determined by LVDV bit. Both the POR bit and the LVD bit in SIMRS are set following a POR.

10.4.2 LVD reset operation

The LVD can be configured to generate a reset upon detection of a low voltage condition has occurred by setting LVDRE to 1 when the supply voltage has fallen below the level determined by LVDV bit. After an LVD reset has occurred, the LVD system will hold the NTM88 in reset until the supply voltage has risen above the level determined by LVDV bit. The threshold for falling and rising differ by a small amount of hysteresis. The LVD bit in the SIMRS register is set following either an LVD reset or POR.

10.4.3 LVD interrupt operation

When a low voltage condition is detected and the LVD circuit is configured for interrupt operation (LVDE set, LVDIE set, and LVDRE clear), then LVDF is set and an LVD interrupt occurs.

10.4.4 Low-Voltage Warning (LVW)

The LVD system has a low voltage warning flag, LVWF, to indicate to the user that the supply voltage is approaching, but is still above, the LVD reset voltage. The LVWF can be reset by writing a logical one to the LVWACK bit. The LVW does not have an interrupt associated with it. There are two user selectable trip voltages for the LVW as selected by LVWV in SPMSC3. The LVWF is set when the supply voltage falls below the selected level and cannot be reset until the supply voltage has risen above the selected level. The threshold for falling and rising differ by a small amount of hysteresis.

10.5 System clock control

Several clock rate selections are possible with the NTM88 using the BUSCLKS[1:0] control bits to select the clock frequency division of the HFO as given in Table 18. These bits are cleared by any MCU reset.

Table 18. HFO frequency selections

BUSCLKS1	BUSCLKS0	HFO Frequency (MHz)	CPU Bus Frequency (MHz)
0	0	8	4
0	1	4	2
1	0	2	1
1	1	1	0.5

10.6 Keyboard interrupts

The keyboard interrupts can be used to wake the MCU. These are assigned to specific general I/O pins as given in Table 19.

Note: *Regarding wake-up from Stop1, the reset vector is accessed, taking precedence over the interrupt vector.*

Table 19. Keyboard interrupt assignments

KBI	Pin	Pin Function
0	PTA0	General I/O
1	PTA1	General I/O
2	PTA2	General I/O
3	PTA3	General I/O

10.7 Real-time interrupt

The RTI uses the internal low frequency oscillator (LFO) as its clock source. The RTI can be used as a periodic interrupt in MCU RUN mode, or can be used as a periodic wake-up from all low-power modes. The LFO is always active and cannot be powered off by any software control. The control bits for the RTI are shown in Table 175.

Note: *Regarding wake-up from Stop1, the reset vector is accessed, taking precedence over the interrupt vector.*

10.8 Modes of operation

The operating modes of the NTM88 are described in this section. Entry into each mode, exit from each mode, and functionality while in each of the modes is described.

10.8.1 Features

- ACTIVE BACKGROUND DEBUG mode for code development
- STOP modes:
  - System clocks stopped
  - STOP1: Power down of most internal circuits, including RAM, for maximum power savings; voltage regulator in standby
  - STOP4: All internal circuits powered and full voltage regulation maintained for fastest recovery

10.8.2 RUN mode

This is the normal operating mode for the NTM88. This mode is selected when the BKGD/PTA4 pin is high at the rising edge of reset. In this mode, the CPU executes code from internal memory following a reset with execution beginning at address specified by the reset pseudo-vector ($DFFE and $DFFF).

10.8.3 WAIT mode

The WAIT mode is also present like other members of the NXP S08 family members; but is not normally used by the NTM88 firmware or typical TPMS applications.

10.8.4 ACTIVE BACKGROUND mode

The ACTIVE BACKGROUND mode functions are managed through the BACKGROUND DEBUG controller (BDC) in the HCS08 core. The BDC provides the means for analyzing MCU operation during software development.

ACTIVE BACKGROUND mode is entered in any of four ways:

- When the BKGD/PTA4 pin is low at the rising edge of a power-up reset
- When a BACKGROUND command is received through the BKGD/PTA4 pin
- When a BGND instruction is executed by the CPU
- When encountering a BDC breakpoint

Once in ACTIVE BACKGROUND mode, the CPU is held in a suspended state waiting for serial BACKGROUND commands rather than executing instructions from the user's application program. Background commands are of two types:

- Non-intrusive commands, defined as commands that can be issued while the user program is running. Non-intrusive commands can be issued through the BKGD/PTA4 pin while the MCU is in RUN mode; non-intrusive commands can also be executed when the MCU is in the ACTIVE BACKGROUND mode. Non-intrusive commands include:
  - Memory access commands
  - Memory-access-with-status commands
  - BDC register access commands
  - The BACKGROUND command
- ACTIVE BACKGROUND commands, which can only be executed while the MCU is in ACTIVE BACKGROUND mode. ACTIVE BACKGROUND commands include commands to:
  - Read or write CPU registers
  - Trace one user program instruction at a time
  - Leave ACTIVE BACKGROUND mode to return to the user's application program (GO)

The ACTIVE BACKGROUND mode is used to program a boot loader or user application program into the FLASH program memory before the MCU is operated in RUN mode for the first time. When the NTM88 is shipped from the NXP factory, the FLASH program memory is erased by default (unless specifically requested otherwise) so there is no program that could be executed in RUN mode until the FLASH memory is initially programmed.

The ACTIVE BACKGROUND mode can also be used to erase and reprogram the FLASH memory after it has been previously programmed.

10.8.5 STOP Modes

One of two stop modes are entered upon execution of a STOP instruction when the STOPE bit in the system option register is set. In all STOP modes, all internal clocks are halted except for the low frequency 1 kHz oscillator (LFO) which runs continuously whenever power is applied to the V_{DD} and V_{SS} pins. If the STOPE bit is not set when the CPU executes a STOP instruction, the MCU will not enter any of the STOP modes and an illegal opcode reset is forced. The STOP modes are selected by setting the appropriate bits in SPMSC2. Table 20 summarizes the behavior of the MCU in each of the STOP1 and STOP4 modes.

10.8.5.1 STOP1 Mode

The STOP1 mode provides the lowest possible standby power consumption by causing the internal circuitry of the MCU to be powered down.

When the MCU is in STOP1 mode, all internal circuits that are powered from the voltage regulator are turned off. The voltage regulator is in a low-power standby state. STOP1 is exited by asserting either a reset or an interrupt function to the MCU.

Entering STOP1 mode automatically asserts LVD. STOP1 cannot be exited until the V_{DD} is greater than V_{LVDH} or $V_{LV/DL}$ rising (V_{DD} must rise above the LVI re-arm voltage).

Upon wake-up from STOP1 mode, the MCU will start up as from a power-on reset (POR) by taking the reset vector.

Note: *If there are any pending interrupts that have yet to be serviced, then the device will not go into the STOP1 mode. Be certain that all interrupt flags have been cleared before entry to STOP1 mode.*

10.8.5.2 STOP4 LVD enabled in STOP mode

The LVD system is capable of generating either an interrupt or a reset when the supply voltage drops below the LVD voltage. If the LVD is enabled by setting the LVDE and the LVDSE bits in SPMSC1 when the CPU executes a STOP instruction, then the voltage regulator remains active during STOP mode. If the user attempts to enter the STOP1 with the LVD enabled in STOP (LVDSE = 1), the MCU enters STOP4 instead.

Table 20. STOP mode behavior

Mode	STOP1	STOP4
LFO Oscillator, PWU	Always On and Clocking	
Free-Running Counter (FRC)	Always On and Optionally Counting	
Real-Time Interrupt (RTI)[1]	Always On if using LFO as Clock	
MFO Oscillator [2]	Optionally On	Optionally On
HFO Oscillator	Off	Off
CPU	Off	Standby
RAM	Off	Standby
Parameter Registers	On	On
FLASH	Off	Standby
TPM1 2-Chan Timer/PWM	Off	Off
Digital I/O	Disabled	Standby
Sensor Measurement Interface (SMI)	Off	Optionally On
Pressure P-cell	Off	Optionally On
Optional Acceleration g-cell	Off	Optionally On
Temperature Sensor (in ADC10)	Off	Optionally On [3]
Voltage Reference (in ADC10)	Off	Optionally On(3)
LFR Detector [4]	Periodically On	Periodically On
LFR Decoder	Optionally On	Optionally On
RF Controller, Data Buffer, Encoder	Optionally On	Optionally On

Table 20. STOP mode behavior*...continued*

Mode	STOP1	STOP4
RF Transmitter [5]	Optionally On	Optionally On
ADC10	Off	Optionally On(3)
Regulator	Off	On
I/O Pins	Hi-Z	States Held
Wake-up Methods	Interrupts, resets	Interrupts, resets
Computer Operating Properly (COP) watchdog	Off	Off

[1] The interrupt from RTI operates from all power modes, however the RTIF flag will not be set and the interrupt service routine will not execute if the RTI is configured and STOP1 mode entered. RTIF flag and the interrupt service routine execute if in Run mode or if STOP4 is entered.
[2] MFO oscillator started if the LFR detectors are periodically sampled, the LFR detectors detect an input signal; a pressure or acceleration reading is in progress or the RF state machine is sending data.
[3] Requires internal ADC10 clock to be enabled.
[4] Period of sampling set by MCU.
[5] RF data buffer may be set up to run while the CPU is in the STOP modes.

Specific to the tire pressure monitoring application the parameter registers and the LFO with wake-up timer are powered up at all times whenever voltage is applied to the supply pins. The LFR detector and MFO may be periodically powered up by the LFR decoder.

10.8.5.3 Active BDM enabled in STOP mode

If the ENBDM bit in BDCSCR is set, entry into the ACTIVE BACKGROUND DEBUG mode from RUN mode is enabled. The BDCSCR register is not memory mapped so it can only be accessed through the BDM interface by use of the BDM commands READ_STATUS and WRITE_CONTROL. If ENBDM is set when the CPU executes a STOP instruction, the system clocks to the BACKGROUND DEBUG logic remain active when the MCU enters STOP mode so BACKGROUND DEBUG communication is still possible. In addition, the voltage regulator does not enter its low-power standby state but maintains full internal regulation. If the user attempts to enter the STOP1 with ENBDM set, the MCU will instead enter this mode which is STOP4 with system clocks running.

Most BACKGROUND commands are not available in STOP mode. The memory-access-with-status commands do not allow memory access, but they report an error indicating that the MCU is in STOP mode. The BACKGROUND command can be used to wake the MCU from stop and enter ACTIVE BACKGROUND mode if the ENBDM bit is set. Once in BACKGROUND DEBUG mode, all BACKGROUND commands are available.

10.8.5.4 MCU on-chip peripheral modules in STOP modes

When the MCU enters any STOP mode, system clocks to the internal peripheral modules except the wake-up timer and LFR detectors/decoder are stopped. Even in the exception case (ENBDM = 1), where clocks are kept alive to the BACKGROUND debug logic, clocks to the peripheral systems are halted to reduce power consumption.

10.8.5.4.1 I/O pins

If the MCU is configured to go into STOP1 mode, the I/O pins are forced to their default reset state (Hi-Z) upon entry into stop. This means that the I/O input and output buffers are turned off and the pullup is disconnected.

10.8.5.4.2 Memory

All module interface registers are reset upon wake-up from STOP1 and the contents of RAM are not preserved. The MCU must be initialized as upon reset. The contents of the FLASH memory are non-volatile and are preserved in any of the STOP modes.

10.8.5.4.3 Parameter registers

The 64 bytes of parameter registers are kept active in all modes of operation as long as power is applied to the supply pins. The contents of the parameter registers behave like RAM and are unaffected by any reset.

10.8.5.4.4 LFO

The LFO remains active regardless of any mode of operation.

10.8.5.4.5 FRC

The Free-Running Counter can be enabled or halted. Once enabled and not halted, the FRC remains active regardless of any mode of operation.

10.8.5.4.6 MFO

The medium frequency oscillator (MFO) remains powered up when the MCU enters the STOP mode only when the SMI has been initiated to make a pressure or acceleration measurement; or when the RF transmitter's state machine is processing data.

10.8.5.4.7 HFO

The HFO is halted in all STOP modes.

10.8.5.4.8 PWU

The PWU remains active regardless of any mode of operation.

10.8.5.4.9 ADC10

The internal asynchronous ADC10 clock is always used as the conversion clock. The ADC10 can continue operation during STOP4 mode. Conversions can be initiated while the MCU is the STOP4 mode. All ADC10 module registers contain their reset values following exit from STOP1 mode. See Section 10.17.

10.8.5.4.10 LFR

When the LFR is enabled and the MCU enters STOP mode, the detectors in the LFR remain powered up depending on the states of the bits selecting the periodic sampling. See Section 10.15 for more details.

10.8.5.4.11 Band gap reference

The band gap reference should be enabled whenever the sensor measurement interface requires sensor or voltage measurements.

10.8.5.4.12 TPM1

When the MCU enters STOP mode, the clock to the TPM1 module stops and the module halts operation. If the MCU is configured to go into STOP1 mode, the TPM1 module is reset upon wake-up from STOP and must be re-initialized.

10.8.5.4.13 Voltage regulator

The voltage regulator enters a low-power standby state when the MCU enters any of the STOP modes except STOP4 (LVDSE = 1 or ENBDM = 1).

10.8.5.4.14 Temperature sensor

The temperature sensor is powered up on command from the MCU.

10.8.5.5 RFM module in STOP modes

The RFM's external crystal oscillator (XCO), bit rate generator, PLL, VCO, RF data buffer, data encoder, and RF output stage will remain powered up in STOP modes during a transmission, or if the SEND bit has been set and DIRECT mode has been enabled.

10.8.5.5.1 RF output

When the RFM finishes a transmission sequence the external crystal oscillator (XCO), bit rate generator, PLL, VCO, RF data buffer, data encoder, and RF output stage will remain powered up if the SEND bit is set.

10.8.5.6 P-cell in STOP modes

The P-cell is powered up only during a measurement if scheduled by the sensor measurement interface. Otherwise it is powered down.

10.8.5.7 Optional g-cell in STOP modes

The g-cell is powered up only during a measurement if scheduled by the sensor measurement interface. Otherwise it is powered down.

10.9 Memory

The overall memory map of the NTM88 resides on the MCU.

10.9.1 Memory map - parts delivered without firmware in flash

Table 21. Memory map for parts delivered without firmware in flash

Start Address	End Address	Type	Block description
$0000	$004F	Register	80 bytes direct page peripheral control registers for GPIO, KBI, IRQ, TPM, PWU, LF, ADC, SPI, SMI
$0050	$008F	Parameter	64 bytes Always-On parameter registers
$0090	$028F	RAM	512 bytes RAM
$0290	$07FF	Not mapped	1392 bytes not mapped
$0800	$17FF	SPI / Flash test access	4096 bytes Virtual addresses for SPI access to 4096 byte blocks of flash memory

Table 21. Memory map for parts delivered without firmware in flash*...continued*

Start Address	End Address	Type	Block description
$1800	$188F	Register	144 bytes high page peripheral control registers for interrupt, SIM, RTI, PMC, Flash, RF, FRC
$1890	$BFFF	Not mapped	42864 bytes not mapped
			Start of erase and re-program addresses supported by library IDE patches
$C000	$FBFF	Flash	15,360 bytes user program - erase and re-program with library IDE patches
			Intermediate end of erase and re-program addresses supported by library IDE patch; beginning of library protected sector
$FC00	$FD3F	Protected Flash	320 bytes user program - program one-time with library IDE patches; not erasable and not re-programmable after 1st use.
$FD40	$FD7D	Protected Flash	62 bytes coefficients and limits for manf./test - not erasable with library IDE patch
$FD7E	$FDA9	Protected Flash	44 bytes SMI coefficients for manf./test - not erasable with library IDE patch
$FDAA	$FDAA	Protected Flash	1 byte CodeF - not erasable with library IDE patch
$FDAB	$FDFA	Protected Flash	80 bytes trim coefficients; not erasable with library IDE patch
$FDFB	$FDFF	Protected Flash	5 bytes CodeH + unique ID - not erasable with library IDE patch
			Resumption of erase and re-program addresses supported by library IDE patch; end of library protected sector
$FE00	$FFAB	Flash	428 bytes user program - erase and re-program with library IDE patch
$FFAC	$FFAF	Flash	4 bytes CodeF + target ID - erase and re-program with library IDE patch
$FFB0	$FFBF	Flash	16 bytes flash key, protection and security coefficients; one-time programmable with library IDE patch
$FFCO	$FFDF	Flash	32 bytes user program - erase and re-program with library IDE patch
$FFE0	$FFFF	Flash	32 bytes ISR hardware vectors; erase and re-program with library IDE patch
			End of erase and re-program addresses supported by library IDE patch

10.10 Clock distribution

The various clock sources and their distribution are shown in Figure 12. All clock sources except the low frequency oscillator, LFO, can be turned off by software control in order to conserve power.

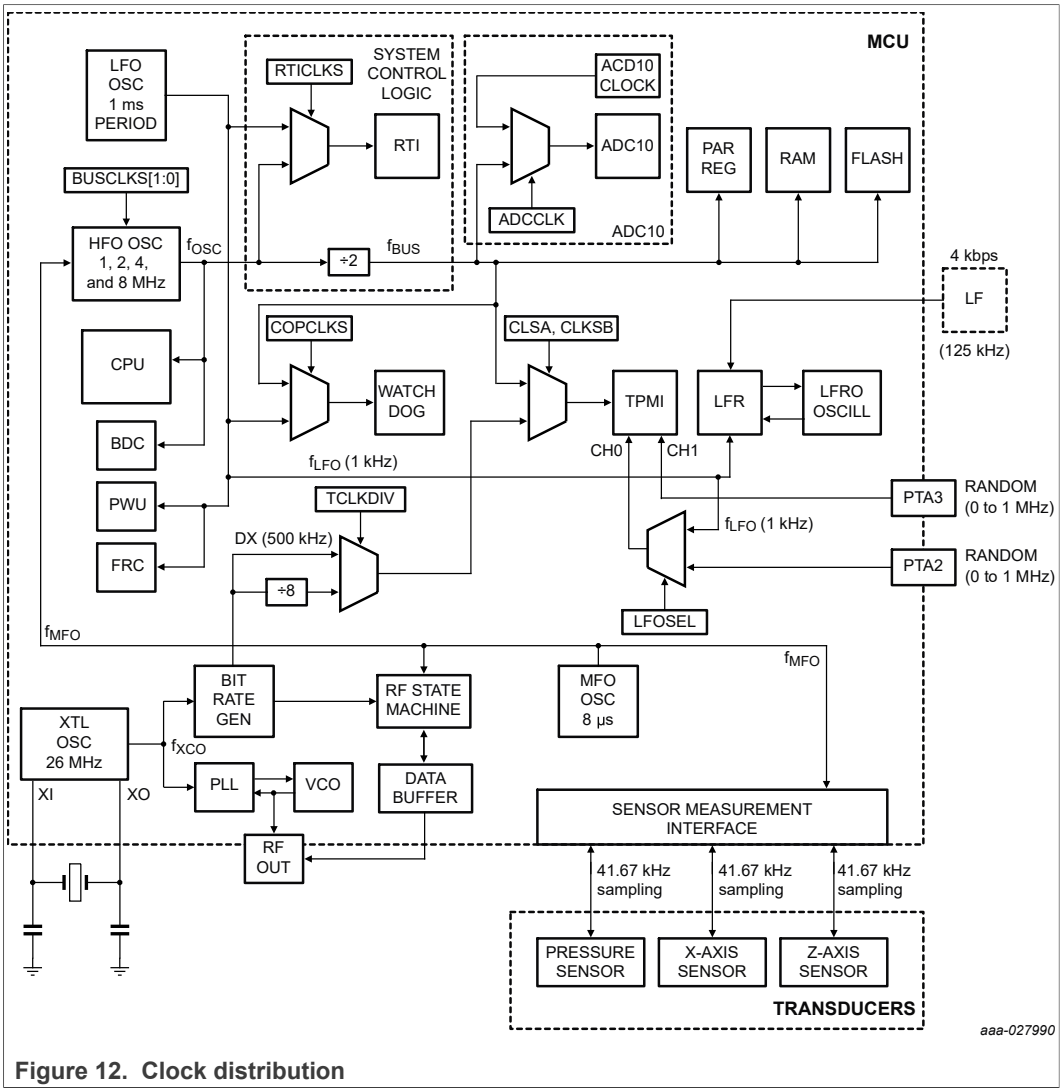


Figure 12. Clock distribution

10.11 Reset, interrupts and system configuration

This section discusses basic reset and interrupt mechanisms and the various sources of reset and interrupts in the NTM88. Some interrupt sources from peripheral modules are discussed in greater detail within other sections of this document. This section gathers basic information about all reset and interrupt sources in one place for easy reference. A few reset and interrupt sources, including the computer operating properly (COP) watchdog and real-time interrupt (RTI), are not part of on-chip peripheral systems, but are part of the system control logic.

10.11.1 Features

Reset and interrupt features include:

- Multiple sources of reset for flexible system configuration and reliable operation
- Reset status register (SIMRS) to indicate source of most recent reset
- Separate interrupt vectors for each module (reduces polling overhead)

10.11.2 MCU reset

Resetting the MCU provides a way to start processing from a known set of initial conditions. During reset, most control and status registers are forced to initial values and the program counter is loaded from the reset vector ($DFFE:$DFFF). On-chip peripheral modules are disabled and any I/O pins are initially configured as general-purpose high-impedance inputs with any pullup devices disabled. The I bit in the condition code register (CCR) is set to block maskable interrupts so the user program has a chance to initialize the stack pointer (SP) and system control settings. The SP is forced to $00FF at reset. The NTM88 has seven sources for reset:

- Power-on reset (POR)
- Low-voltage detect (LVD)
- Computer operating properly (COP) timer
- Periodic hardware reset (PRST)
- Illegal opcode detect
- Illegal address detect
- BACKGROUND DEBUG forced reset

Each of these sources has an associated bit in the system reset status register with the exception of the BACKGROUND DEBUG forced reset and the periodic hardware reset, PRST, that is indicated by the PRF bit in the PWUCS1 register.

10.11.3 Computer Operating Properly (COP) Watchdog

The COP watchdog is intended to force a system reset when the application software fails to execute as expected. To prevent a system reset from the COP timer (when it is enabled), application software must reset the COP timer periodically. If the application program gets lost and fails to reset the COP before it times out, a system reset is generated to force the system back to a known starting point. The COP watchdog is enabled by the COPE bit in SIMOPT1 register. The COP timer is reset by writing any value to the address of SIMRS. This write does not affect the data in the read-only SIMRS. Instead, the act of writing to this address is decoded and sends a reset signal to the COP timer.

The timeout period can be selected by the COPCLKS and the COPT[2:0] bits as shown in Table 22. The COPCLKS bit selects either the LFO or the CPU bus clock as the clocking source and the COPT[2:0] bits select the clock count required for a timeout. The tolerance of these timeout periods is dependent on the selected clock source (LFO or HFO).

Table 22. COP watchdog timeout period

COPCLKS	COPT			Clock Source	COP Overflow Count	COP Overflow Time (ms, nominal)
	2	1	0			
0	0	0	0	LFO	2^5	32
0	0	0	1	LFO	2^6	64
0	0	1	0	LFO	2^7	128
0	0	1	1	LFO	2^8	256
0	1	0	0	LFO	2^9	512
0	1	0	1	LFO	2^{10}	1024

Table 22. COP watchdog timeout period*...continued*

COPCLKS	COPT			Clock Source	COP Overflow Count	COP Overflow Time (ms, nominal)			
	2	1	0						
0	1	1	0	LFO	2^{11}	2048			
0	1	1	1	LFO	2^{11}	2048			
						BUSCLKS[1:0]			
						1:1 (0.5 MHz)	1:0 (1 MHz)	0:1 (2 MHz)	0:0 (4 MHz)
1	0	0	0	Bus Clock	2^{13}	16.384	8.192	4.096	2.048
1	0	0	1	Bus Clock	2^{14}	32.768	16.384	8.192	4.096
1	0	1	0	Bus Clock	2^{15}	65.536	32.768	16.384	8.192
1	0	1	1	Bus Clock	2^{16}	131.072	65.536	32.768	16.384
1	1	0	0	Bus Clock	2^{17}	262.144	131.072	65.536	32.768
1	1	0	1	Bus Clock	2^{18}	524.288	262.144	131.072	65.536
1	1	1	0	Bus Clock	2^{19}	1048.576	524.288	262.144	131.072
1	1	1	1	Bus Clock	2^{19}	1048.576	524.288	262.144	131.072

After any reset, the COP timer is enabled. This provides a reliable way to detect code that is not executing as intended. If the COP watchdog is not used in an application, it can be disabled by clearing the COPE bit in the write-once SIMOPT1 register. Even if the application will use the reset default settings in COPE, COPCLKS and COPT[2:0], the user should still write to write- once SIMOPT1 during reset initialization to lock in the settings. That way, they cannot be changed accidentally if the application program gets lost.

The write to SIMRS that services (clears) the COP timer should not be placed in an interrupt service routine (ISR) because the ISR could continue to be executed periodically even if the main application program fails. When the MCU is in ACTIVE BACKGROUND DEBUG mode, or either Stop1 or Stop4 modes, the COP timer is temporarily disabled. If enabled, the COP timer is reset at the time entering Stop1 and Stop4 modes, and will restart after 3 cycles of the selected clock source upon exiting; RTI may be used as a substitute.

10.12 General purpose I/O port pins

10.12.1 GPIO register descriptions

PTA[4:0] and PTB[1:0] pins are shared with on-chip peripheral functions. The peripheral modules have priority over the general purpose I/O so that when a peripheral is enabled, the general purpose I/O functions associated with the shared pins are disabled. After reset, the shared peripheral functions are disabled so that the pins are controlled as general purpose I/O.

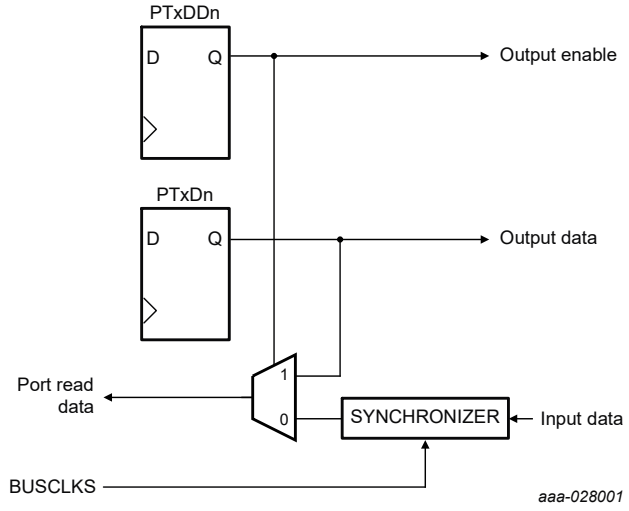


Figure 13. General purpose I/O block diagram

Reading and writing of general purpose I/O is performed through the port data registers PTxDn. The direction, either read of input or write of output, is controlled through the port data direction registers PTxDDn. When configured as input, the pull-up or pull-downs are controlled through a combination of port pull enable registers PTxPEn and the PTxDDn registers. Where x refers to the port A or B, and n refers to the port pin 0, 1, etc.

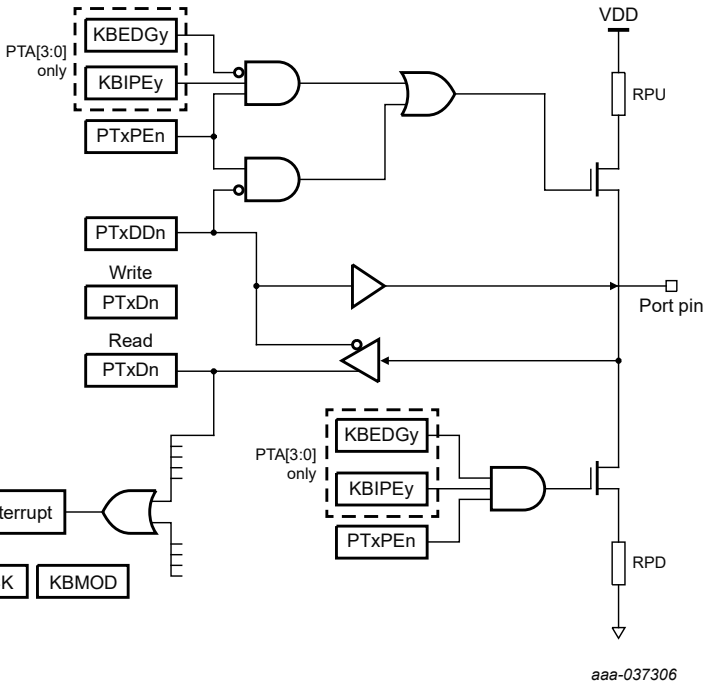


Figure 14. General purpose I/O logic

Port A [3:0] GPIOs support a keyboard interrupt peripheral function. Each keyboard interrupt pin can be programmed for edge or level or both sensitivity. The sensitivities can be programmed for falling edge / low level or rising edge / high level while in run mode, and falling edge / low level while in stop modes.

Table 23. Truth table for pullup and pulldown resistors

PTAPE[3:0] pull enable	PTADD[3:0] data direction	KBIPE[3:0] KBI pin enable	KBEDG[3:0] KBI edge select	Pullup	Pulldown
0	0	x	x	disabled	disabled
1	0	0	x	enabled	disabled
x	1	0	x	disabled	disabled
1	0	1	0	enabled	disabled
1	0	1	1	disabled	enabled

Table 23. Truth table for pullup and pulldown resistors

PTBPE[1:0] pull enable	PTBADD[1:0] data direction				
0	0			disabled	x
1	0			enabled	x
x	1			disabled	x

Port A 0 supports an external interrupt as a peripheral function. The PTA0 GPIO can be configured as an external Interrupt Request (IRQ), which when activated will force the CPU to exit a stop mode.

Port A 4 supports a background developer interface as a peripheral function. The PTA4 GPIO can be configured as the BDM serial data interface (BKGD) by an external host holding the PTA4 pin low prior to POR release.

10.12.1.1 General Purpose I/O

This section explains software controls related to general purpose input/output (I/O) and pin control. The NTM88 has seven general-purpose I/O pins which are comprised of a general use 5-bit port A and a 2-bit port B.

To avoid extra current drain from floating input pins, the user's application software must configure these pins so that they do not float (see Section 10.12.1.1.1 "Unused pin configuration").

Reading and writing of general purpose I/O is performed through the port data registers. The direction, either input or output, is controlled through the port data direction registers. The general purpose I/O port function for an individual pin is illustrated in the block diagram in Figure 13.

The data direction control bit (PTxDDn) determines whether the output buffer for the associated pin is enabled, and also controls the source for port data register reads. The input buffer for the associated pin is always enabled unless the pin is enabled as an analog function.

When a shared digital function is enabled for a pin, the output buffer is controlled by the shared function. However, the data direction register bit still controls the source for reads of the port data register.

When a shared analog function is enabled for a pin, both the input and output buffers are disabled. A value of 0 is read for any port data bit where the bit is an input (PTxDDn = 0) and the input buffer is disabled. In general, whenever a pin is shared with both an alternate digital function and an analog function, the analog function has priority such that if both the digital and analog functions are enabled, the analog function controls the pin.

It is a good programming practice to write to the port data register before changing the direction of a port pin to become an output. This ensures that the pin will not be driven momentarily with an old data value that happened to be in the port data register.

An internal pullup device can be enabled for each port pin by setting the corresponding bit in one of the pullup enable registers (PTxPEn). The pullup device is disabled if the pin is configured as an output by the general purpose I/O control logic or any shared peripheral function regardless of the state of the corresponding pullup enable register bit. The pullup device is also disabled if the pin is controlled by an analog function.

10.12.1.1.1 Unused pin configuration

Any general purpose I/O pins which are not used in the application must be properly configured to avoid a floating input that could cause excessive supply current, I_{DD}.

When the device comes out of the reset state the NXP supplied firmware will not configure any of the general purpose I/O pins.

Recommended configuration methods are:

1. Configure the general purpose I/O pin as an input (PTxDDn = 0) with the pin connected to the V_{DD} source; use a pullup resistor of 10-51 kΩ to assure sufficient noise immunity.
2. Configure the general purpose I/O pin as an input (PTxDDn = 0) with the internal pullup activated (PTxPEn = 1) and leave the pin disconnected.
3. Configure the general purpose I/O pin as an output (PTxDDn = 1) and drive the pin low (PTxDn = 0) and leave the pin disconnected.

In cases where GPIOs are directly connected to AV_{DD}, V_{DD}, AV_{SS}, V_{SS} or RV_{SS}, user application should configure the GPIO as an input with the internal pull-up disabled, in order to prevent software code faults from causing excessive supply current states should these pins become outputs.

10.12.1.1.2 Pin behavior in STOP modes

Pin behavior following execution of a STOP instruction depends on the STOP mode that is entered. An explanation of pin behavior for the various STOP modes follows:

- In STOP1 mode, all internal registers including general purpose I/O control and data registers are powered off. Each of the pins assumes its default reset state (input buffer, output buffer and internal pullup disabled). Upon exit from STOP1, all pins must be reconfigured the same as if the MCU had been reset.
- In STOP4 mode, all pin states are maintained because internal logic stays powered up. Upon recovery, all pin functions are the same as before entering STOP4.

10.12.1.2 Port A data register (PTAD)

Table 24. Port A data register (PTAD) (address $0000)

Bit	7	6	5	4	3	2	1	0
R W	reserved	reserved	reserved	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
Reset ($00)	0	0	0	0	0	0	0	0

Table 25. PTAD register field descriptions

Field	Description
4 PTAD[4:0]	PTAD[4:0] – For port A pins that are inputs, reads return the logic level on the pin. For port A pins that are configured as outputs, reads return the last value written to this register. Writes are latched into all bits of this register. For port A pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTAD to all 0s, but these 0s are not driven out the corresponding pins because reset also configures all port pins as high-impedance inputs with pullups disabled. Each bit 0 = pin inactive or connected to ground; Result of Reset Each bit 1 = pin active or connected to $V_{DD(A)}$ 0 0 0 0 0 = Result of Reset

10.12.1.3 Port A pin pull enable register (PTAPE)

Table 26. Port A pin pull enable register (PTAPE) (address $0001)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	0	PTAPE3	PTAPE2	PTAPE1	PTAPE0
W	—	—	—	—				
Reset ($00)	0	0	0	0	0	0	0	0

Table 27. PTAPE register field descriptions

Field	Description
3:0 PTAPE	PTAPE[3:0] – Each bit selects the internal pullup device is enabled for the associated PTA pin. For port A pins that are configured or default as output, these bits have no effect and the internal pullup devices are disabled. Each bit 0 = Internal pullup device disabled for port A bit n; Result of Reset Each bit 1 = Internal pullup device enabled for port A bit n. 0 0 0 0 = Result of Reset

10.12.1.4 Port A data direction register (PTADD)

Table 28. Port A data direction register (PTADD) (address $0003)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	1	PTADD3	PTADD2	PTADD1	PTADD0
W	—	—	—	—				
Reset ($00)	0	0	0	0	0	0	0	0

Table 29. PTADD register field descriptions

Field	Description
3:0 PTADD[3:0]	PTADD[3:0] - Each bit selects the direction of port A pins and what is read for PTADD reads. Each bit 0 = Input (output driver disabled) and reads return the pin value; Result of Reset Each bit 1 = Output driver enabled for port A bit n and PTADD reads return the contents of PTADDn. 0 0 0 0 = Result of Reset ***Note:*** *In GPIO mode, PTA4 operates as output-only.*

10.12.1.5 Port B data register (PTBD)

Table 30. Port B data register (PTBD) (address $0004)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PTBD1	PTBD0
W	—	—	—	—	—	—		
Reset ($00)	0	0	0	0	0	0	0	0

Table 31. PTBD register field descriptions

Field	Description
1:0 PTBD[1:0]	PTBD[1:0] – For port B pins that are inputs, reads return the logic level on the pin. For port B pins that are configured as outputs, reads return the last value written to this register. Writes are latched into all bits of this register. For port B pins that are configured as outputs, the logic level is driven out the corresponding MCU pin. Reset forces PTBD to all 0s, but these 0s are not driven out the corresponding pins because reset also configures all port pins as high-impedance inputs with pullups disabled. Each bit 0 = pin inactive or connected to ground; Result of Reset Each bit 1 = pin active or connected to $V_{DD(A)}$ 0 0 = Result of Reset

10.12.1.6 Port B pin pull enable register (PTBE)

Table 32. Port B pin pull enable register (PTBE) (address $0005)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PTBPE1	PTBPE0
W	—	—	—	—	—	—		
Reset ($00)	0	0	0	0	0	0	0	0

Table 33. PTBE register field descriptions

Field	Description
1:0 PTBPE[1:0]	PTBPE[1:0] – Each bit selects the internal pullup device is enabled for the associated PTB pin. For port B pins that are configured as outputs, these bits have no effect and the internal pullup devices are disabled. Each bit 0 = Internal pullup device disabled for port B bit n; Result of Reset Each bit 1 = Internal pullup device enabled for port B bit n. 0 0 = Result of Reset

10.12.1.7 Port B data direction register (PTBDD)

Table 34. Port B data direction (PTBDD) (address $0007)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PTBDD1	PTBDD0
W	—	—	—	—	—	—		
Reset ($00)	0	0	0	0	0	0	0	0

Table 35. PTBDD register field descriptions

Field	Description
1:0 PTBDD[1:0]	PTBDD[1:0] - Each bit selects the direction of port B pins and what is read for PTBDD reads. Each bit 0 = Input (output driver disabled) and reads return the pin value; Result of Reset Each bit 1 = Output driver enabled for port B bit n and PTBDD reads return the contents of PTBDDn. 0 0 = Result of Reset

10.12.2 External wake-up functions

10.12.2.1 KBI status and control register (KBISC)

Note:

Prior to enabling the keyboard by setting the KBIE to 1, this status byte, as a first step, must be read to avoid an immediate assertion of the interrupt.

In addition, the keyboard interrupt KBF results immediately:

- *if a port pin PTA[3:0] is at a logic 1 state and*
- *the user subsequently enables the keyboard by setting the corresponding KBIE[3:0] to 1 and*
- *the user sets the edge to rising/high by setting to 1 the corresponding KBIES[3:0]*

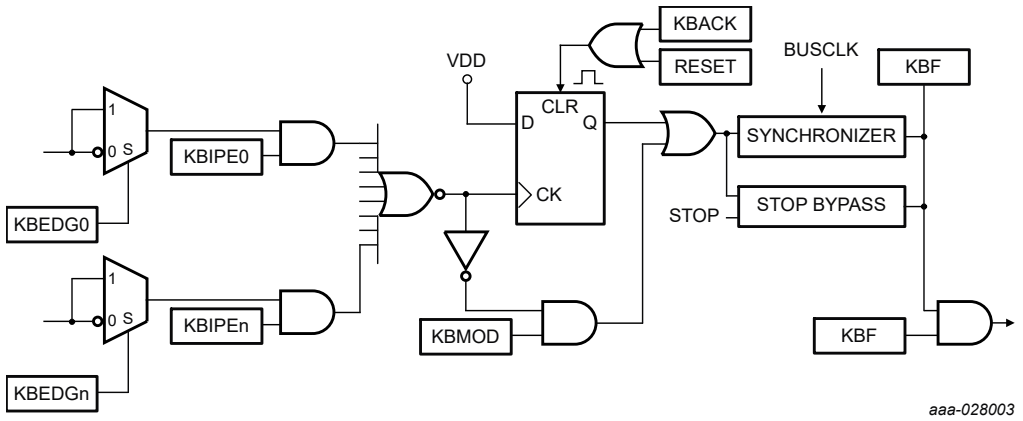


Figure 15. KBI block diagram

Table 36. KBI status and control register (KBISC) (address $000C)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	0	KBF	0	KBIE	KBIMOD
W	—	—	—	—	—	KBACK		
Reset	0	0	0	0	0	0	0	0
POR ($00)	0	0	0	0	0	0	0	0

Table 37. KBISC register field descriptions

Field	Description
3 KBF	KBF - The read-only KBF bit indicates when a keyboard interrupt is detected. Writes have no effect on KBF. 0 = No keyboard interrupt detected; Result of power-on reset. Existing state will remain after all other types of reset. 1 = Keyboard interrupt detected.
2 KBACK	KBACK - The write-only KBACK bit is part of the flag clearing mechanism. KBACK always reads as 0. 0 = Read result; Write no effect; Result of Reset 1 = Write 1 to clear KBF for Keyboard interrupt acknowledge.
1 KBIE	KBIE - Keyboard Interrupt Enable — KBIE determines whether a keyboard interrupt is requested. 0 = Keyboard interrupt request not enabled; Result of Reset 1 = Keyboard interrupt request enabled.
0 KBIMOD	KBIMOD - Keyboard Detection Mode — KBMOD (along with the KBEDG bits) controls the detection mode of the keyboard interrupt pins. 0 = Keyboard detects edges only; Result of Reset 1 = Keyboard detects both edges and levels.

10.12.2.2 Keyboard interrupt pin enable register (KBIPE)

Table 38. Keyboard interrupt pin enable register (KBIPE) (address $000D)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	0	KBIPE3	KBIPE2	KBIPE1	KBIPE0
W	—	—	—	—				
Reset ($00)	0	0	0	0	0	0	0	0

Table 39. KBIPE register field descriptions

Field	Description
3:0 KBIPE[3:0]	KBIPE[3:0] – The 4 bits KBIPE[3:0] selects corresponding keyboard interrupt pin from Port A GPIOs. 0 = Pin not enabled as keyboard interrupt; Result of Reset 1 = Pin enabled as keyboard interrupt.

10.12.2.3 Keyboard interrupt edge select register (KBIES)

Table 40. Keyboard interrupt edge select register (KBIES) (address $000E)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	0	KBEDG3	KBEDG2	KBEDG1	KBEDG0
W	—	—	—	—				
Reset ($00)	0	0	0	0	0	0	0	0

Table 41. KBIES register field descriptions

Field	Description
3:0 KBED GE[3:0]	KBEDGE[3:0] – The 4 bits KBEDGE[3:0] selects the edge/low level or rising edge/high level function of the corresponding pin. 0 = Falling edge/low level, available in all modes; Result of Reset 1 = Rising edge/high level, only available while in Run mode.

10.12.2.4 Ext. interrupt status and control register (IRQSC)

Note: *Prior to enable of the IRQ by setting to 1 the IRQIE, this status byte must be read as a first step to avoid an immediate assertion of the interrupt. Also, the Interrupt IRQF will immediately result:*

- *if the port pin PTA0 is at a logic 1 state and*
- *the user subsequently enables the Interrupt by setting to 1 the IRQPE and the user sets the edge to rising/high by setting to 1 the IRQEDG*

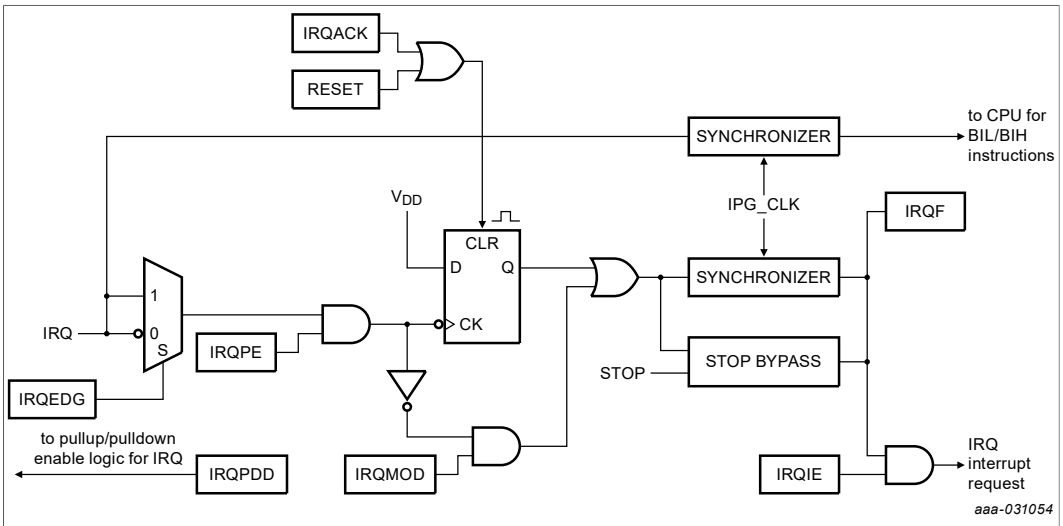


Figure 16. External interrupt logic

Table 42. Ext. interrupt status and control register (IRQSC) (address $000F)

Bit	7	6	5	4	3	2	1	0
R	0	IRQPDD	IRQEDG	IRQPE	IRQF	0	IRQIE	IRQMOD
W	—				—	IRQACK		
Reset	0	0	0	0	U	0	0	0
POR ($00)	0	0	0	0	0	0	0	0

Table 43. IRQSC register field descriptions

Field	Description
6 IRQPDD	IRQPDD — IRQ Pull Device Disable Bit The IRQPDD bit is used to disable the on-chip pullup/pulldown device on the IRQ pin. This allows users to have an external device if required for their application. 0 = On-chip pullup/pulldown device is enabled; Result of Reset 1 = On-chip pullup/pulldown device is disabled

Table 43. IRQSC register field descriptions*...continued*

Field	Description
5 IRQEDG	IRQEDG – The IRQEDG bit selects the edge/low level or rising edge/high level function of the PTA0 pin. 0 = Falling edge/low level, available in all modes; Result of Reset 1 = Rising edge/high level, only available while in Run mode.
4 IRQPE	IRQPE – The IRQPE bit enables the external PTA0 pin to function as the IRQ source. 0 = PTA0 not selected as the IRQ source; Result of Reset 1 = PTA0 selected as the IRQ source.
3 IRQF	IRQF – IRQ pending Flag The read-only IRQF bit indicates when a wake-up interrupt has been generated by the external IRQ. This bit is cleared by writing a one to the IRQACK bit. Writing a zero to this bit has no effect. 0 = external interrupt not generated or was previously acknowledged; Result of power-on reset. Existing state will remain after all other types of reset. 1 = external interrupt generated.
2 IRQACK	IRQACK – IRQ Acknowledge The write-only IRQACK bit clears the IRQF bit if written with a one. Writing a zero to the IRQACK bit has no effect on the IRQF bit. Reading the IRQACK bit returns a zero. Reset has no effect on this bit. 0 = Read result; Write no effect; Result of Reset 1 = Write 1 to clear IRQF for IRQ interrupt acknowledge
1 IRQIE	IRQIE – IRQ Interrupt Enable The IRQIE bit enables or disables the external IRQ interrupt function 0 = IRQ interrupt disabled; Result of Reset 1 = IRQ interrupt enabled
0 IRQMOD	IRQMOD – Keyboard Detection Mode IRQMOD (along with the IRQEDG bits) controls the detection mode of the keyboard interrupt pins. 0 = IRQ detects on falling or rising edges only; Result of Reset 1 = IRQ detects both edges and levels.

10.13 Timer pulse-width module

The timer pulse-width module (TPM1) is a two channel timer system that supports traditional input capture, output compare, or edge-aligned PWM on each channel. All the features and functions of the TPM1 are as described in the MC9S08RC16 product specification. The user has the option to connect the two timer channels to the PTB[1:0] pins for interface to external circuits.

The TPM1 has the following features:

- May be configured for buffered, center-aligned pulse-width modulation (CPWM) on all channels
- Clock sources independently selectable
- Selectable clock sources (device dependent): bus clock, fixed system clock
- Clock prescaler taps for divide by 1, 2, 4, 8, 16, 32, 64, or 128
- 16-bit free-running or up/down (CPWM) count operation
- 16-bit modulus register to control counter range
- Timer system enable
- One interrupt per channel plus a terminal count interrupt

- Channel features:
  - Each channel may be input capture, output compare, or buffered edge-aligned PWM
  - Rising-edge, falling-edge, or any-edge input capture trigger
  - Set, clear, or toggle output compare action
  - Selectable polarity on PWM outputs

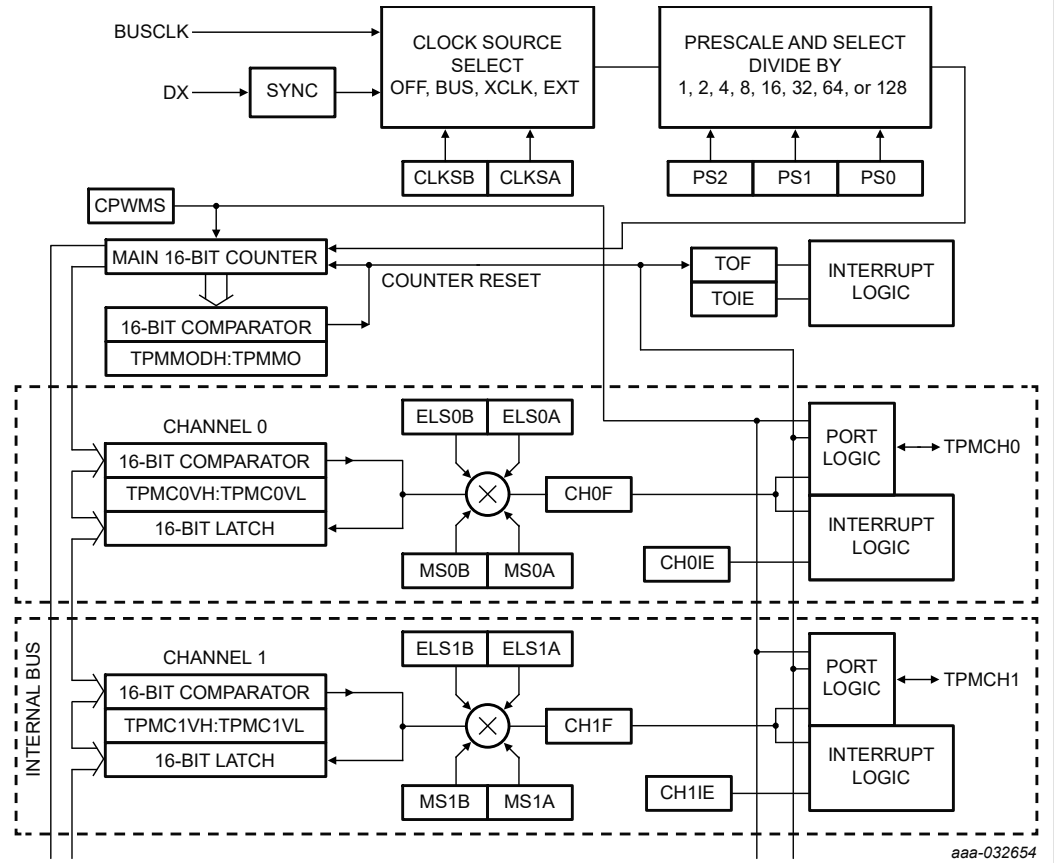


Figure 17. Timer pulse-width block diagram

10.13.1 TPM1 configuration information

The device provides one two-channel timer/pulse-width modulator (TPM1).

An easy way to measure the low frequency oscillator (LFO) is to connect the LFO directly to TPM1 channel 0. The LFOSEL bit in the SOPTZ determines whether TPM1CH0 is connected to PTAZ or the LFO.

TPM1 clock source selection for the TPM1 is shown in the following table.

Table 44. TPM1 clock source selection

CLKSB	CLKSA	Clock Source
0	0	No source; TPM1 disabled
0	1	BUSCLK
1	0	unused
1	1	Internal DX pin

10.13.1.1 Block diagram

Figure 17 shows the structure of a TPM1.

The central component of the TPM1 is the 16-bit counter that can operate as a free-running counter, a modulo counter, or an up- /down-counter when the TPM1 is configured for center-aligned PWM. The TPM1 counter (when operating in normal up-counting mode) provides the timing reference for the input capture, output compare, and edge-aligned PWM functions. The timer counter modulo registers, TPMMODH:TPMMODL, control the modulo value of the counter. (The values 0x0000 or 0xFFFF effectively make the counter free running.) Software can read the counter value at any time without affecting the counting sequence. Any write to either byte of the TPMCNT counter resets the counter regardless of the data value written.

All TPM1 channels are programmable independently as input capture, output compare, or buffered edge-aligned PWM channels.

10.13.2 External signal description

When any pin associated with the timer is configured as a timer input, a passive pullup can be enabled. After reset, the TPM1 modules are disabled and all pins default to general-purpose inputs with the passive pullups disabled.

Each TPM1 channel is associated with an I/O pin on the MCU. The function of this pin depends on the configuration of the channel. In some cases, no pin function is needed so the pin reverts to being controlled by general-purpose I/O controls. When a timer has control of a port pin, the port data and data direction registers do not affect the related pin(s). See Section 7 "Pinning information" for additional information about shared pin functions.

10.13.3 TPM register descriptions

10.13.3.1 Timer status and control register (TPMSC)

Table 45. Timer status and control register (TPMSC) (address $0010)

Bit	7	6	5	4	3	2	1	0
R	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
W	—							
Reset ($00)	0	0	0	0	0	0	0	0

Table 46. TPMSC register field descriptions

Field	Description
7 TOF	TOF – Timer Overflow Flag This read-only TOF bit is set when the TPM1 counter changes to 0000 after reaching the modulo value programmed in the TPM1 counter modulo registers. When the TPM1 is configured for CPWM, TOF is set after the counter has reached the value in the modulo register, at the transition to the next lower count value. Clear TOF by reading the TPM1 status and control register when TOF is set and then writing a 0 to TOF. If another TPM1 overflow occurs before the clearing sequence is complete, the sequence is reset so TOF would remain set after the clear sequence was completed for the earlier TOF. Writing a 1 to TOF has no effect. 0 = TPM1 counter has not reached modulo value or overflow; Result of power-on reset. Existing state will remain after all other types of reset. 1 = TPM1 counter has overflowed
6 TOIE	TOIE – Timer Overflow Interrupt Enable This read/write bit enables TPM1 overflow interrupts. If TOIE is set, an interrupt is generated when TOF equals 1. 0 = TOF interrupts inhibited (use software polling); Result of Reset 1 = TOF interrupts enabled
5 CPWMS	CPWMS – Center-aligned PWM Select This read/write bit selects CPWM operating mode. Reset clears this bit so the TPM1 operates in up-counting mode for input capture, output compare, and edge-aligned PWM functions. Setting CPWMS reconfigures the TPM1 to operate in up-/down-counting mode for CPWM functions. 0 = All TPM channels operate as input capture, output compare, or edge-aligned PWM mode as selected by the MSnB:MSnA control bits in each channel's status and control register; Result of Reset 1 = All TPM channels operate in center-aligned PWM mode
4:3 CLKS[B:A]	CLKS[B:A] – Clock Source Select The 2-bits CLKS[B:A] are used to disable the TPM1 system or select one of three clock sources to drive the counter prescaler. The internal DX source is synchronized to the bus clock by an on-chip synchronization circuit. 0 0 = No source selected, TPM disabled; Result of Reset 0 1 = Bus clock selected 1 0 = undefined, TPM enabled but not clocking 1 1 = Internal Dx clock from RF module selected, approx. 500 kHz
[2:0] PS[2:0]	PS[2:0] – Prescale Divisor Selection The 3-bits PS[2:0] selects one of eight divisors for the TPM1 clock input. This prescaler is located after any clock source synchronization or clock source selection, so it affects whatever clock source is selected to drive the TPM1 system. 0 0 0 = divide by 1; Result of Reset 0 0 1 = divide by 2 0 1 0 = divide by 4 0 1 1 = divide by 8 1 0 0 = divide by 16 1 0 1 = divide by 32 1 1 0 = divide by 64 1 1 1 = divide by 128

10.13.3.2 Timer counter high and low registers (TPMCNTH/L)

Table 47. Timer counter high register (TPMCNTH) (address $0011)

Bit	7	6	5	4	3	2	1	0
R	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
W								
Reset ($00)	0	0	0	0	0	0	0	0

Table 48. Timer counter low register (TPMCNTL) (address $0012)

Bit	7	6	5	4	3	2	1	0
R	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
W								
Reset ($00)	0	0	0	0	0	0	0	0

Table 49. TPMCNTH/L register field descriptions

Field	Description
15:0	The two read-only TPMCNT[15:0] counter registers contain the high and low bytes of the value in the TPM1 counter. Reading either byte (TPM1CNTH or TPM1CNTL) latches the contents of both bytes into a buffer where they remain latched until the other byte is read. This allows coherent 16-bit reads in either order. The coherency mechanism is automatically restarted by an MCU reset, a write of any value to TPM1CNTH or TPM1CNTL, or any write to the timer status/control register (TPM1SC). Reset clears the TPM1 counter registers.

10.13.3.3 Timer modulo high and low registers (TPMMODH/L)

Table 50. Timer modulo high register (TPMMODH) (address $0013)

Bit	7	6	5	4	3	2	1	0
R	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
W								
Reset ($00)	0	0	0	0	0	0	0	0

Table 51. Timer modulo low register (TPMMODL) (address $0014)

Bit	7	6	5	4	3	2	1	0
R	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
W								
Reset ($00)	0	0	0	0	0	0	0	0

Table 52. TPMMODH/L register field descriptions

Field	Description
15:0	The read/write TPMMOD[15:0] modulo registers contain the modulo value for the TPM1 counter. After the TPM1 counter reaches the modulo value, the TPM1 counter resumes counting from 0000 at the next clock (CPWMS = 0) or starts counting down (CPWMS = 1), and the overflow flag (TOF) becomes set. Writing to TPM1MODH or TPM1MODL inhibits TOF and overflow interrupts until the other byte is written. Reset results in a free-running timer counter (i.e. modulo disabled). $0000 = Result of Reset

10.13.3.4 Timer channel 0/1 status and control registers (TPMCySC)

Where y = Channel 0 or Channel 1.

Table 53. Timer channel 0 status and control register (TPMC0SC) (address $0015)

Bit	7	6	5	4	3	2	1	0
R	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
W	—						—	—
Reset ($00)	0	0	0	0	0	0	0	0

Table 54. Timer channel 1 status and control register (TPMC1SC) (address $0018)

Bit	7	6	5	4	3	2	1	0
R	CH1F	CH1IE	MS1	MS1A	ELS1B	ELS1A	0	0
W	—						—	—
Reset ($00)	0	0	0	0	0	0	0	0

Table 55. TPMCySC register field descriptions

Field	Description
7 CH0/1F	CHyF – Channel 0/1 Flag When channel n is configured for input capture, this read-only CHyF bit is set when an active edge occurs on the channel 0/1 pin. When channel 0/1 is an output compare or edge-aligned PWM channel, CHyF is set when the value in the TPM1 counter registers matches the value in the TPM1 channel 0/1 value registers. This flag is seldom used with center-aligned PWMs because it is set every time the counter matches the channel value register, which corresponds to both edges of the active duty cycle period. A corresponding interrupt is requested when CHyF is set and interrupts are enabled (CHyIE = 1). Clear CHyF by reading TPM1CySC while CHyF is set and then writing a 0 to CHyF. If another interrupt request occurs before the clearing sequence is complete, the sequence is reset so CHyF would remain set after the clear sequence was completed for the earlier CHyF. This is done so a CHyF interrupt request cannot be lost by clearing a previous CHyF. Writing a 1 to CHyF has no effect. 0 = No input capture or output compare event occurred on channel 0; Result of power-on reset. 1 = Input capture or output compare event occurred on channel 0; Result of other reset types.
6 CH0/1IE	CHyiE – Channel 0/1 Interrupt Enable This read/write bit enables interrupts from channel 0/1. 0 = Channel 0/1 interrupt requests disabled (use software polling); Result of Reset 1 = Channel 0/1 interrupt requests enabled

Table 55. TPMCySC register field descriptions*...continued*

Field	Description
5:4 MS0.1[B;a]	MSy[B:A] – Channel 0/1 Mode Select When CPWMS = 0, MSyB = 1 configures TPM1 channel 0/1 for edge-aligned PWM mode. When CPWMS = 0 and MSyB = 0, MSyA configures TPM1 channel 0/1 for input capture mode or output compare mode.
3:2 ELS0/1[B:A]	ELSy[B:A] – Channel 0/1 Edge/Level Select Depending on the operating mode for the timer channel as set by CPWMS:MSyB:MSyA and shown below, these bits select the polarity of the input edge that triggers an input capture event, select the level that will be driven in response to an output compare match, or select the polarity of the PWM output. Setting ELSyB:ELSyA to 0:0 configures the related timer pin as a general-purpose I/O pin unrelated to any timer channel functions. This function is typically used to temporarily disable an input capture channel or to make the timer pin available as a general-purpose I/O pin when the associated timer channel is set up as a software timer that does not require the use of a pin.

Table 56. Timer channel operating mode settings

CPWMS	MSy[B:A]	ELSy[B:A]	Mode
x	x	0 0	Pin not used for TPM1 channel; use as an external clock for the TPM1 or revert to general-purpose I/O; Result of Reset
0	0 0	0 1	Input capture rising edge
0	0 0	1 0	Input capture falling edge
0	0 0	1 1	Input capture rising or falling edges
0	0 1	0 0	Output compare software monitor
0	0 1	0 1	Output compare toggle output on compare match
0	0 1	1 0	Output compare clear output on compare match
0	0 1	1 1	Output compare set output on compare match
0	1 x	1 0	Edge-aligned PWM clear output on compare match
0	1 x	x 1	Edge-aligned PWM set output on compare match
1	x x	1 0	Center-aligned PWM clear output on compare match
1	x x	x 1	Center-aligned PWM set output on compare match

10.13.3.5 Timer channel 0/1 value registers (TPMCyVH/L)

Where y = Channel 0 or Channel 1.

Table 57. Timer channel 0 value register (TPMC0VH) (addresses $0016)

Bit	7	6	5	4	3	2	1	0
R W	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Reset ($00)	0	0	0	0	0	0	0	0

Table 58. Timer channel 0 value register (TPMC0VL) (addresses $0017)

Bit	7	6	5	4	3	2	1	0
R W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset ($00)	0	0	0	0	0	0	0	0

Table 59. Timer channel 1 value register (TPMC1VH) (addresses $0019)

Bit	15	14	13	12	11	10	9	8
R W	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Reset ($00)	0	0	0	0	0	0	0	0

Table 60. Timer channel 1 value register (TPMC1VL) (addresses $001A)

Bit	7	6	5	4	3	2	1	0
R W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset ($00)	0	0	0	0	0	0	0	0

Table 61. TPMCyVH/L register field descriptions

Field	Description
15:0 TPMCyV[15:0]	The TPMCyV[15:0] read/write registers contain the captured TPM1 counter value of the input capture function or the output compare value for the output compare or PWM functions. The channel value registers are cleared by reset. In input capture mode, reading either byte (TPM1CyVH or TPM1CyVL) latches the contents of both bytes into a buffer where they remain latched until the other byte is read. This latching mechanism also resets (becomes unlatched) when the TPM1CySC register is written. In output compare or PWM modes, writing to either byte (TPM1CyVH or TPM1CyVL) latches the value into a buffer. When both bytes have been written, they are transferred as a coherent 16-bit value into the timer channel value registers. This latching mechanism may be manually reset by writing to the TPM1CySC register. This latching mechanism allows coherent 16-bit writes in either order, which is friendly to various compiler implementations. $0000 = Result of Reset

10.14 Periodic wake-up timer module

The periodic wake-up timer (PWU) generates a periodic interrupt to wake up the MCU from any of the STOP modes. It also has an optional periodic reset to restart the MCU. It is driven by the LFO oscillator in the RTI module which generates a clock at a nominal one millisecond interval. The LFO and the wake-up timer are always active and cannot be powered off by any software control. The control bits are set so that there is either a periodic wake-up, a periodic reset, or both a wake-up interrupt and a periodic reset. No combination of control bits will disable both the wake-up interrupt and the periodic

reset. In addition, there is no hardware control that can mask a wake-up interrupt once it is generated by the PWU.

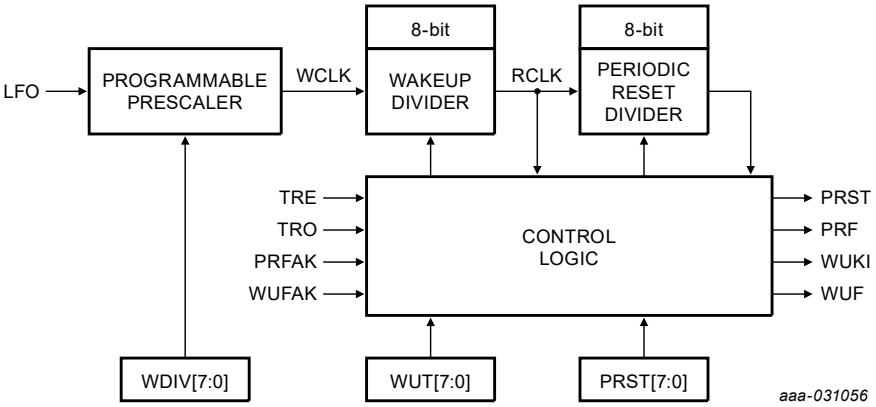


Figure 18. Periodic wake-up timer block diagram

10.14.1 PWU timer register descriptions

10.14.1.1 Periodic wake-up status and control register (PWUSR)

Table 62. Periodic wake-up status and control register (PWUSR) (address $001B)

Bit	7	6	5	4	3	2	1	0
R	WUF	0	PSEL	PRF	0	0	0	0
W	—	WUFACK		—	PRFACK	—	—	—
Reset	U	0	0	U	0	0	0	0
POR ($00)	0	0	0	0	0	0	0	0

Table 63. PWUSR register field descriptions

Field	Description
7 WUF	WUF – Wake-up Interrupt Flag The read-only WUF bit indicates when a wake-up interrupt has been generated by the PWU. This bit is cleared by writing a one to the WUFAK bit. Writing a zero to this bit has no effect. 0 = Wake-up interrupt not generated or was previously acknowledged; Result of power-on reset. Existing state remains after periodic reset. 1 = Wake-up interrupt generated.
6 WUFACK	WUFACK – Wake-up Interrupt Acknowledge The write-only WUFAK bit clears the WUF bit if written with a one. Writing a zero to the WUFAK bit has no effect on the WUF bit. Reading the WUFAK bit returns a zero. Reset has no effect on this bit. 0 = Read result; Write no effect; Result of Reset 1 = Write 1 to clear WUF for Wake-up interrupt acknowledge.
5 PSEL	PSEL – Page Select The PSEL read/write bit selects whether the CSTAT[7:0] register represents the RCLK or PRT counters. This bit is cleared by a power-on reset that is not created by an exit from the STOP mode, but is unaffected by other resets. 0 = CSTAT[7:0] represent the RCLK counter status; Result of Reset 1 = CSTAT[7:0] represent the PRT counter status

Table 63. PWUSR register field descriptions*...continued*

Field	Description
4 PRF	PRF – Periodic Reset Flag The read-only PRF bit indicates when a periodic reset has been generated by the PWU. MCU writes to this bit have no effect. This bit is cleared by writing a one to the PRFAK bit. 0 = Periodic reset not generated or previously acknowledged; Result of power-on reset. Existing state remains after periodic reset. 1 = Periodic reset generated.
3 PRFACK	PRFACK – PRF Interrupt Acknowledge The PRFAK bit clears the PRF bit if written with a one. Writing a zero to the PRFAK bit has no effect on the PRF bit. Reading the PRFAK bit returns a zero. 0 = Read result; Write no effect; Result of Reset 1 = Write 1 to clear PRF for Periodic Reset interrupt acknowledge.

10.14.1.2 Periodic wake-up divider register (PWUDIV)

Table 64. Periodic wake-up divider register (PWUDIV) (address $001C)

Bit	7	6	5	4	3	2	1	0
R	WDIV7	WDIV6	WDIV5	WDIV4	WDIV3	WDIV2	WDIV1	WDIV0
W								
Reset ($1F)	0	0	0	1	1	1	1	1

Table 65. PWUDIV register field descriptions

Field	Description
[7:0] WDIV	The WDIV[7:0] bits select a divider for the incoming LFO clock to generate the wake-up clock. The operating range of WDIV[7:0] is $00 up to $FF. Reading WDIV[7:0] provides the value written. This results in a wake-up clock with periods from 0.504 seconds up to 4.584 seconds, when the LFO is 1 kHz. The user can use this divider to fine-tune the wake-up time based on the variation in the LFO frequency. The conversion from the decimal value of the WDIV[7:0] bits to the wake-up clock time is given as described in the following equation. Power-on-reset forces WDIV[7:0] to a value of $1F (decimal 31), and results in WCLK of 1 second, assuming LFO is typical 1 kHz. $$\mathrm{WCLK} = \frac{(504 + 16 \times \mathrm{WDIV}[7:0])}{f_{\mathrm{LFO}}}$$ Where: f_{LFO} = LFO frequency in Hz, ~1 kHz typical

10.14.1.3 Periodic wake-up interrupt register (PWUCS0)

Table 66. Periodic wake-up interrupt register (PWUCS0) (address $001D)

Bit	7	6	5	4	3	2	1	0
R	WUT7	WUT6	WUT5	WUT4	WUT3	WUT2	WUT1	WUT0
W								
Reset ($FF)	1	1	1	1	1	1	1	1

Table 67. PWUCS0 register field descriptions

Field	Description
[7:0] WUT	The WUT[7:0] bits select the number of wake-up clocks until the next wake-up interrupt is generated. Wake-up interrupt time RCLK = Wake-up clock time WCLK x WUT[7:0] The WUT[7:0] gives a range of wake-up interrupt times from 1 to 255 x wake-up clocks. Depending on the value of the bits for the WDIV[7:0] this time interval can nominally be from 0.504 s to 1168.92 s in 0.504 s steps. Whenever the WUT[7:0] bits are changed, the timeout period is restarted. Writing the same data to the WUT[7:0] bits has no effect. Writing zeros to all of the WUT[7:0] bits forces the wake-up divider to a value of $FF and disables the wake-up interrupt. However, writing all zeros to the WUT[7:0] bits is inhibited if all of the PRST[7:0] bits are already cleared to zero. This prevents disabling both the periodic wake-up and the periodic reset at the same time. The WUT[7:0] bits are preset to a value of $FF (decimal 255) by any resets. $FF = Result of power on or periodic wake-up unit reset.

10.14.1.4 Periodic wake-up reset register (PWUCS1)

Table 68. Periodic wake-up reset register (PWUCS1) (address $001E)

Bit	7	6	5	4	3	2	1	0
R	PRST7	PRST6	PRST5	PRST4	PRST3	PRT2	PRST1	PRST0
W								
Reset ($FF)	1	1	1	1	1	1	1	1

Table 69. PWUCS1 register field descriptions

Field	Description
[7:0] PRST	The PRST[7:0] bits select the number of wake-up interrupts until the next periodic reset is generated. Periodic reset time PRT = Wake-up interrupt time RCLK x PRST[7:0] The PRST[7:0] gives a range of periodic reset times from 1 to 255 x wake-up interrupts. Depending on the value of the bits for the WDIV[7:0] and WUT[7:0] this time interval can nominally be from 0.504 s to 4967.91 minutes with steps from 0.504 s to 1168.92 s. Whenever the PRST[7:0] bits are changed the timeout period is restarted. Writing the same data to the PRST[7:0] bits has no effect. Writing zeros to all of the PRST[7:0] bits forces the periodic reset to be disabled if at least one of the WUT[7:0] bits is set to a one. This assures that there will be at least a wake-up interrupt. However, writing all zeros to the PRST[7:0] bits is inhibited if all of the WUT[7:0] bits are already cleared to zero. This prevents disabling both the periodic wake-up and the periodic reset at the same time. The PRST[7:0] bits are preset to a value of $FF (decimal 255) by any resets. $FF = Result of power on or periodic wake-up unit reset.

10.14.1.5 Periodic wake-up counter register (PWUS)

Table 70. Periodic wake-up counter register (PWUS) (address $001F)

Bit	7	6	5	4	3	2	1	0
R	CSTAT7	CSTAT6	CSTAT5	CSTAT4	CSTAT3	CSTAT2	CSTAT1	CSTAT0
W	—	—	—	—	—	—	—	—
Reset ($00)	1	1	1	1	1	1	1	1

Table 71. PWUS register field descriptions

Field	Description
[7:0] CSTAT	The CSTAT[7:0] read-only bits show the status of the counter selected by the PSEL bit. The effect of any reset on these bits depends on how the reset affects the selected counter. Reading these counters immediately after a WUF or PRF generated flag will return zero contents. $00 = Result of power on or periodic wake-up unit reset. ***Note:*** *Due to a coincident alignment of the LFO clock source for the PWU and the PWUS register, an inadvertent read of the PWUS may result in corruption of the PWUDIV, PWUCS0, and PWUCS1 registers. Users are advised to write the PWUDIV, PWUCS0, and PWUCS1 registers just prior to entering a Stop mode, and avoid reading the PWUS register at that time. If a corruption might be detected during a Run mode cycle, users should re-write the desired settings for the PWUDIC, PWUCS0, and WPUCS1 registers prior to entering a Stop mode.*

10.15 Low frequency (LF) receiver module

The low-frequency receiver (LFR) is a very low-power, low-frequency, receiver system for short-range communication in TPMS. The module allows an external coil to be connected to two dedicated differential input pins. In TPMS systems a single coil may be oriented for optimal coupling between the receiver in the tire or wheel and a transmitter coil on the vehicle body or chassis.

This LFR system minimizes power consumption by allowing flexibility in choosing the ratio of on to off times and by turning off power to blocks of circuitry until they are needed during signal reception and protocol recognition. In addition, this LFR system can autonomously listen for valid LF signals, check for protocol and ID information so the main MCU can remain in a very low power standby mode until valid message data has been received.

The LFR can be configured for various message protocols and telegrams to allow it to be used in a broad range of applications. The message preamble must be a series of Manchester coded bits at the nominal 3.906 kbit/s data rate. A synchronization pattern is used to mark the boundary between the preamble and the beginning of Manchester encoded information in the message body. The synchronization pattern is a non-Manchester specific TPMS pattern. Messages can optionally include none, an 8-bit or a 16- bit ID value. Messages may contain any number of data bytes with the end-of-message indicated by detecting an illegal Manchester bit at a data byte boundary.

It is not intended that LFR may be actively receiving/decoding LF signals while physical parameter measurements are being made; or during the time that the RFM may be actively powered up and/or transmitting RF data. The resulting interactions will degrade the accuracy of the LF detection.

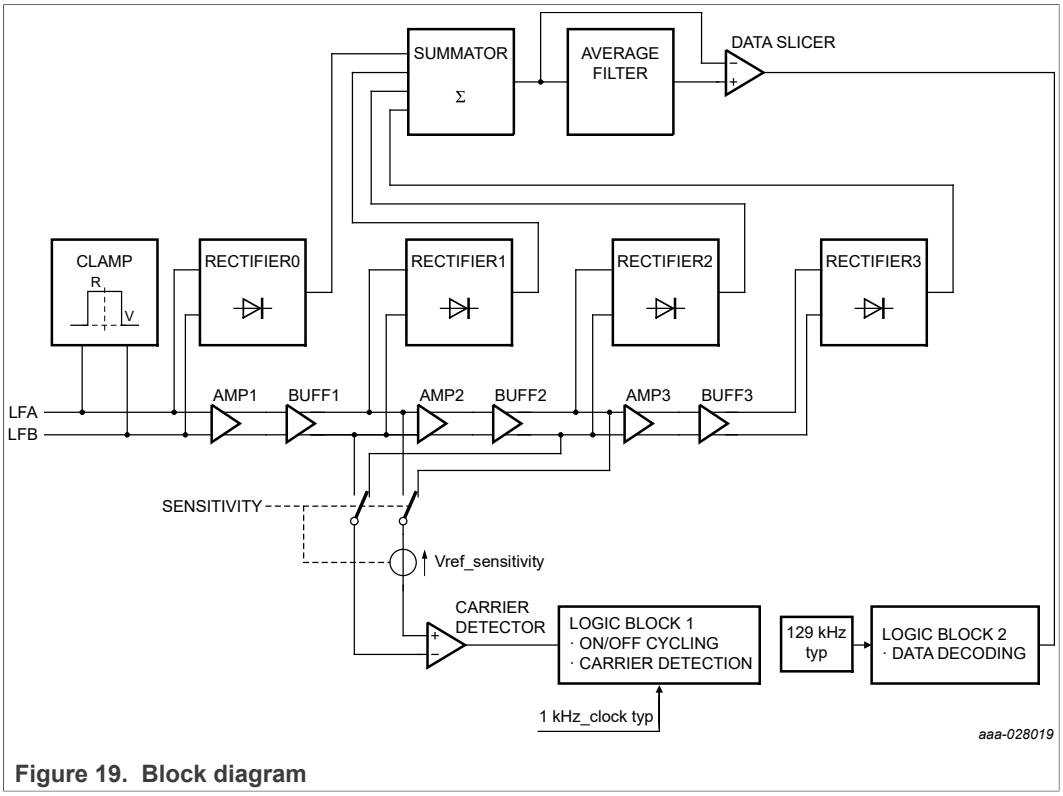


Figure 19. Block diagram

10.15.1 Features

Major features of the LFR module include:

- Differential input LF detector (two dedicated pins):
  - Selectable sensitivity (two levels: Low Sens (LS) and High Sens (HS)).
  - Thresholds trimmed at the factory with trim setting saved in nonvolatile memory.
  - LFR has a reference oscillator (LFRO) trimmed at the factory with trim setting saved in nonvolatile memory.
  - Selectable signal sampling time interval and on-time.
  - Sample interval and on times controlled by LFR state machine or directly by the MCU.
- Configurable receive mode:
  - Simple LF carrier detection/Telegram decode. (CARMOD)
- Configurable message protocol (telegram structure):
  - Various SYNC decoding (SYNC[1:0])
    6-bit time SYNC requirements
    7.5-bit time SYNC requirements
    9-bit time SYNC requirements
  - Optional ID (ID[1:0])
    8-bit or 16-bit ID
    On or off
  - 0-n bytes of message data. End-of-data marked by loss of Manchester at a byte boundary.
- Optional continuous monitoring and decode of the LF detector.

- Selectable MCU interrupt when a received data byte is ready in an LFR buffer, when a Manchester error is detected in the frame, when an ID is received or when a valid carrier has been detected.

10.15.2 Modes of operation

The LFR is a peripheral module on an MCU. After being configured by application software, the LFR can operate autonomously to detect and verify incoming LF messages. When a valid message or carrier pulse is received and verified the LFR can wake the MCU from standby modes to read received data or act upon a carrier detection.

The primary modes of operation for the LFR are:

- Disabled. Everything off and drawing minimal leakage current. LFR register contents will be retained.
- Carrier detect/listen. Minimum circuitry enabled to detect any incoming LF signal, check it for the appropriate signal level, frequency, and duration.
- TPMS protocol verification.
- Data reception.

10.15.3 Power management

In addition to using low power circuit design techniques, the LFR module provides system-level features to minimize system energy requirements. In an MCU that includes the LFR module, all MCU circuitry except a very low current 1 kHz oscillator (LFO) and minimum regulator circuitry can be disabled. After a reset, the MCU would initialize the LFR module and then enter a very low power standby mode (depending upon the MCU, this could be lower than 1 µA for the MCU portion). The LFR module includes everything it needs to periodically listen for LF messages, perform Manchester decoding, verify the message telegram, and assemble incoming data into 8-bit bytes. The LFR does not wake the MCU unless a valid message is being received and a data byte is ready to be read.

The LFR cycles between an off state, where everything is disabled, and an on state, where it listens for a carrier signal. The on time is controlled by LFONTM[3:0] control bits in the LFCTL2 register. The time between the start of each sample on time is controlled by LFSTM[3:0] control bits in the LFCTL2 register. Even lower duty cycles can be achieved by using the MCU to wake once per second and maintain a software counter to delay for an arbitrarily long time before enabling the LFR to perform a series of carrier detect cycles.

Within the LFR, circuits remain disabled until they are needed. When the LFR is listening for a carrier signal, only a 1 kHz clock source, a portion of the input amplifier and a periodic auto-zero are running. After a carrier signal is detected, with high enough amplitude, frequency, and duration the LFRO oscillator is enabled so the LFR can begin to decode the incoming information.

The LFR module has a power up settling time of 2-LFO period before any active operations. In the ON/OFF cycle, those 2 ms are hidden in the sampling time during the off time.

10.15.4 Input amplifier

The LFR module receives LF modulated signals through a dedicated differential pair of inputs which is connected to an external coil. The enable control (LFEN) allows the user to enable the LF input depending on the application requirements. The SENS[1:0] bits

in the LFCTL1 register allows the user to select one of two input sensitivity thresholds which determines the signal level required before the input carrier will be detected. The sensitivity setting is used during carrier detection but does not affect reception after the carrier has been detected. When the CARMOD bit is cleared, after a carrier with sufficient amplitude, frequency, and duration has been detected the output stage of the amplifier is turned on to allow data reception.

10.15.5 LFR data mode states

The modes of operation the LFR state machine will sequence as shown in Figure 20.

10.15.6 Carrier detect

Carrier detection includes a check for a certain number of edges on a signal that is greater than the input sensitivity threshold. During the check for carrier edges, only the 1 kHz low frequency oscillator (LFO) clock source is running so power consumption remains very low.

During carrier detection the incoming signal is amplified and passed through a sensitivity threshold comparator. The SENS[1:0] bits in the LFCTL1 register selects two levels of sensitivity and determines the signal amplitude that is needed to allow edges to be seen at the output of the sensitivity threshold comparator. When a carrier is above this threshold, a block is powered on and validates the carrier. This frequency, and duration check function can be disabled by clearing the VALEN bit. If VALEN is set, the block checks for the carrier duration and the carrier frequency. The time needed to validate a carrier is programmed by the LFCDTM register. The carrier frequency should be 125 kHz. If the signal above the threshold is not within the frequency range or not present during enough time, then the carrier will not be validated and the validation block will turn off.

If no carrier signal is validated within the on time of the LFR, the state machine returns to the off state and the alternating cycle of on time and off time continues. Carrier edge counts start at zero when a new on time begins.

In the data mode (CARMOD = 0), if the required number of carrier edges are detected before the end of the ON time, the LFR will remain ON to complete the reception of a message telegram.

In the carrier detect mode (CARMOD = 1) there is no need to enable other LFR circuitry to evaluate any other message components after the required number of carrier edges are detected. One or several consecutive carriers can be validated by this process before the LFCDF flag is set. The LFCC control bits are used to program the number of consecutive ON times where a complete carrier validation is needed before interrupting the MCU. In this case, the LFCDF flag is set and, provided the LFCDIE interrupt enable is also set, an interrupt is issued to wake the MCU. In carrier detect mode, the LFCDIE control bit should always be set because the intended purpose of the carrier detect mode is to wake the MCU when a carrier is detected. When LFCDF is set, the LFR waits until it is cleared before it continues the alternating cycle of on time and off time, starting with an off time.

In data mode, when a carrier is detected the averaging filter is powered on and the LFR continues to the next state to look for the rest of a message telegram; and the LFR module will search for valid SYNC word (with length programmed through the SYNC bits in the LFCTL3 register depending on preamble type). If the external LF field is not a TPMS frame, a timeout will turn off the LFR module. This timeout can be program through TIMOUT bit the LFCTL4 register.

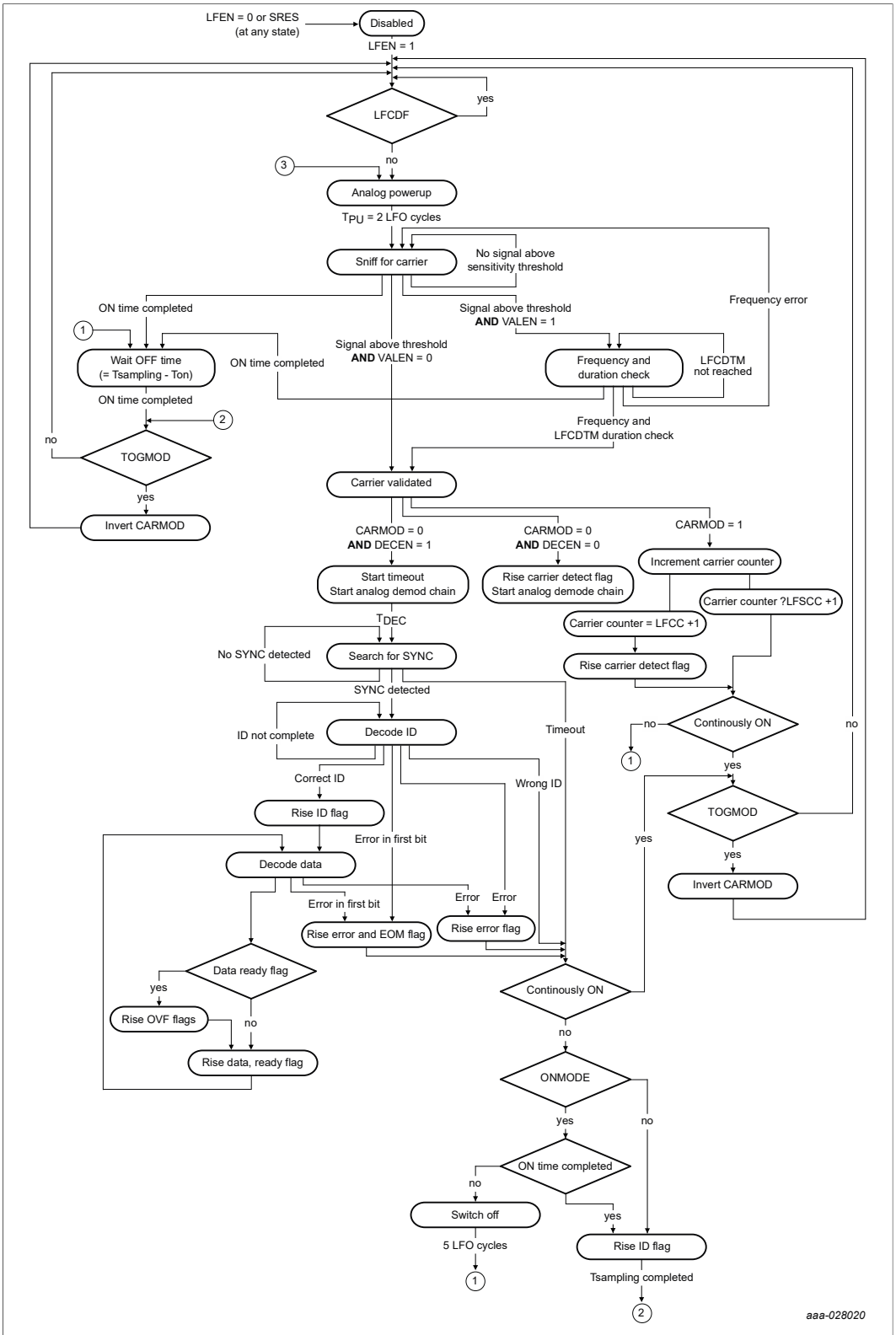


Figure 20. NTM88 LFR state machine diagram

10.15.7 Auto-zero sequence

An auto-zero sequence is performed periodically on the input amplifier to cancel offset errors. During reception of the SYNC pattern and body of the message, auto-zero operations are synchronized to data edges of the incoming signal to avoid interfering with normal reception. During the auto-zero sequence, the input amplifier is temporarily disconnected from the external coil and connected to ground. The auto-zero sequence takes roughly 64 µs. It is performed at each LFO period in carrier mode and on one over four decoded data edges in data mode.

When the DECEN bit is cleared, the auto-zero sequence is performed at each LFO period. During the 64 µs of the auto-zero sequence, the receiver is holding the state "0" or "1" previously decoded. Since the LFR receiver is not active during this time, the possible data-rate that the analog can detect is at least limited by this duration.

10.15.8 Data recovery

Rectified signals from the amplifier output are connected to the input of an averaging filter and data slicer. The slicer therefore compares the rectified signal with its own average value to decode the data. When a carrier is present, the slicer output voltage rises and when the carrier stops the slicer output voltage falls. The output of this comparator provides a binary digital signal that indicates whether the carrier is present or not. This digital signal is connected to the data clock recovery circuit, the SYNC detect circuit, and the Manchester decoder circuit.

The Manchester decoder uses the digital output of the data slicer to detect the logic level of each incoming data bit and to synchronize the decoder state machine. The LFPOL polarity bit in the LFCTRLA register selects the expected encoding of the Manchester data bit.

If a strong signal (above roughly 100 mV p-p differential) is entered into the LFR, the input impedance will switch instantaneously to a lower programmed value (the LOWQ[1:0] bits in the LFCTRLC) and be maintained during the current data packet if the DEQEN bit is set. At the next ON time, the default high input impedance will be set again. The strong signal detection and the automatic impedance change can be disabled by clearing the DEQEN bit.

10.15.9 Data clock recovery and synchronization

Data clock recovery and synchronization takes place during the SYNC portion of an incoming message. The preamble must be modulated Manchester data. The type of required SYNC pattern determines the allowed preamble type depending on the SYNC[1:0] control bits.

The design data rate is 3.906 kbit/s which gives a bit time equivalent to about 32 cycles of the LF carrier frequency. In a Manchester encoded bit time, the carrier should be present for either the first half or the second half of the bit time depending on whether the bit is a logic zero or a logic one.

The LFRO clock source is 32 times the target data rate. The LFRO is used for decoding data and also sequencing auto-zero operations.

10.15.10 Manchester decode

When the LFPOL bit is clear, a logic one bit is defined as no LF carrier present for the first half of the bit time; and a logic zero bit is defined as LF carrier present for the first

half of the bit time as shown in Figure 21. Another way to say this from the point of view of the data slicer output is that a logic zero bit has a falling edge at the middle of the bit time and a logic one bit has a rising edge at the middle of the bit time. The data slicer threshold is dynamically adjusted to the midpoint between the carrier-present and no-carrier levels at the summing node for the rectified output of the LF input amplifier.

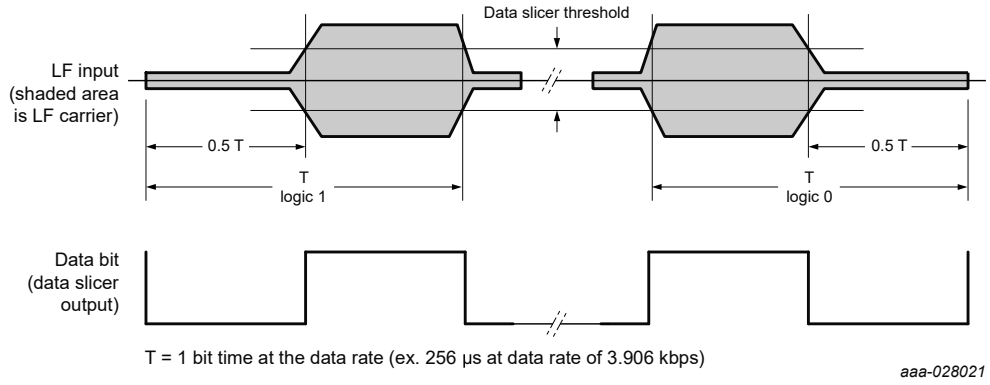


Figure 21. Manchester encoded datagram for LFPOL = 0

When the LFPOL bit is set, a logic one bit is defined as LF carrier present for the first half of the bit time; and a logic zero bit is defined as no LF carrier present for the first half of the bit time as shown in Figure 21.

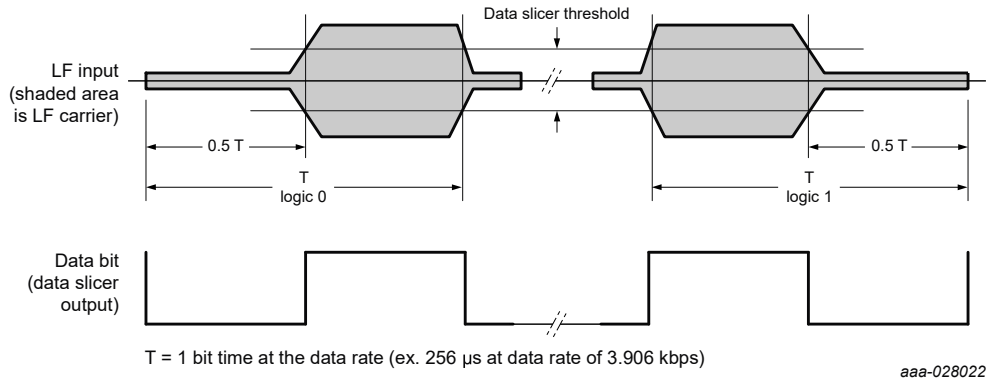


Figure 22. Manchester encoded datagram for LFPOL = 1

10.15.11 Duty cycle for data mode

The definition of the duty cycle for the Manchester encoded data depends on the relative rise and fall times of the incoming LF carrier as shown in Figure 23.

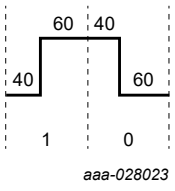


Figure 23. Definition of duty cycle of 40 %

Regarding the SYNC pattern which is non-Manchester coded, the duty cycle is applied on all falling edges with the same proportion as a 1T Manchester symbol, as shown in Figure 24.

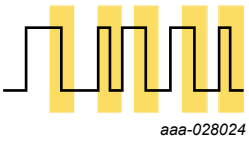


Figure 24. Impact of duty cycle on SYNC pattern

10.15.12 Input signal envelope

The combination of the external LF antenna and any external components as shown in Figure 25 should not significantly filter the envelope of the LF carrier as shown in Figure 26. Excessive filtering will cause the received message error rate (MER) to increase.

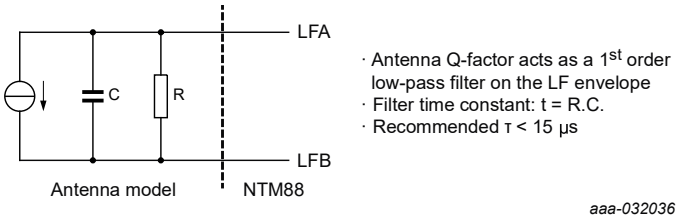


Figure 25. Antenna Q-factor equivalent model for the LF envelope

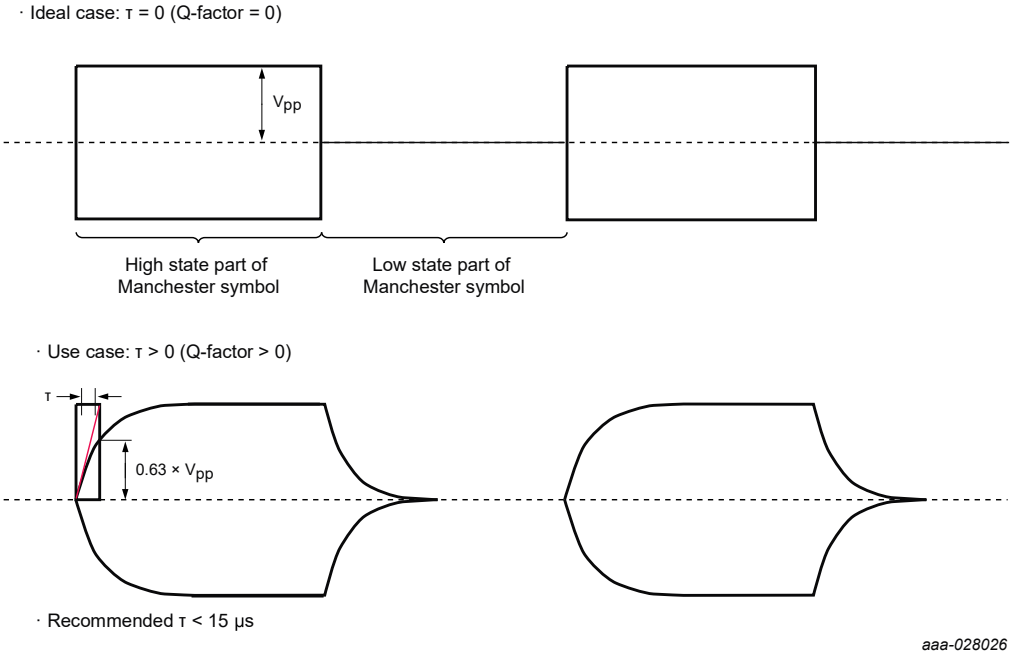


Figure 26. LF envelope filtering

10.15.13 Telegram verification

The LFR has control bits to allow flexibility in the telegram format and protocol to allow the LFR to adapt to various systems. The LFR can operate in a normal data receive mode where it receives complete telegrams, or in a carrier detect mode where it only

checks for a carrier. In the carrier detect mode, as soon as a carrier is detected, the LFCDF flag is set. If LFCDIE is also set, an interrupt request is sent to wake the MCU

The format of the complete Manchester encoded datagram is comprised of a Manchester data preamble (series of Manchester 1s or 0s), a synchronization period, an optional ID, and zero to n data bytes.

The synchronization period can be used for synchronizing the beginning of the data packet. The SYNC pattern that follows the preamble can be either a 6-, 7.5- or 9 bit-time non-Manchester pattern as shown in Figure 27.

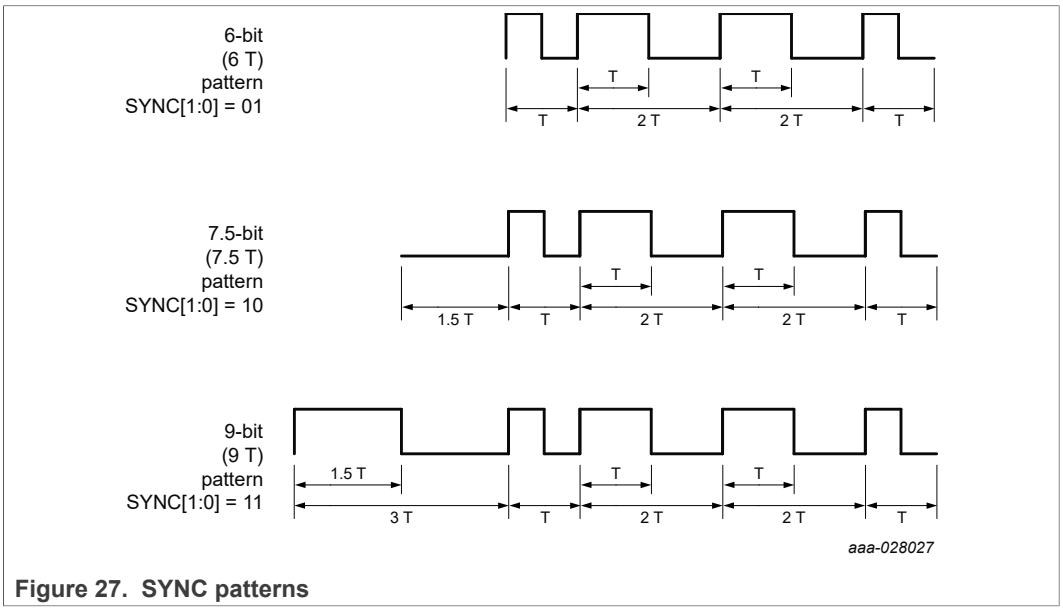


Figure 27. SYNC patterns

These patterns would normally not appear anywhere in the Manchester encoded portion of a message so there is no possibility that the LFR could accidentally synchronize to a message that was already in progress when the LFR started listening for a message. These patterns are also complex enough so that it is very unlikely that noise or interference could be mistaken for these SYNC patterns. In the data mode and after the detection of a valid carrier, the LFR will decode the data stream waiting for the SYNC word. Should this carrier not be an accepted TPMS type, no SYNC will be received and the LFR module will stay in data receive mode forever. A timeout counter is therefore started after a carrier detection and will stop the receiver if reaching the programmed value selected by the TIMOUT[1:0] bits in the LFCTL4 register. This timeout counter is clocked by the internal LFRO clock.

The LFR can be configured to have an optional 0, 8-bit, or 16-bit ID after the SYNC pattern. If the ID value matches the received ID, the message is accepted. The ID value can be used to identify a specific receiver, a message type, or some other identifier as defined by application software.

Any number of data bytes can be included after the ID. The LFR begins to assemble data bytes from the incoming signal as soon as the ID check is complete. If the first bit-time after the last bit of the ID does not conform to Manchester coding requirements, the LFR considers the message complete and terminates the LFR operation without setting the data ready flag (LFDRF). If data follows the ID, it is serially received and when 8 bits have been received the LFR copies this byte into the LFDATA register and sets the LFDRF flag. If the LFDRIE interrupt enable is also set (and it should be), an interrupt request is sent to wake the MCU so it can read the data and process it according to the

instructions in the application program. Additional bytes are received until a bit time that is not Manchester encoded is found. If a non-Manchester bit time is found, the LFERF bit will be set and indicates a Manchester coding error. If this happens on the first bit of the next byte of the message the LFEOMF bit will also be set.

The preamble is a period of Manchester bits before the SYNC pattern as shown in Figure 28. The SYNC pattern will only be matched for the bit times specified by the SYNC[1:0] control bits. Depending on the expected SYNC pattern the allowed preambles is as described for the SYNC[1:0] bits in the LFCTL3 register.

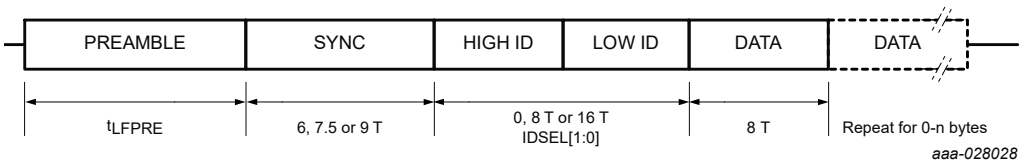


Figure 28. Telegram format (carrier preamble)

10.15.14 Error detection and handling

When the DECEN bit is set, LFR messages are monitored for data rate or SYNC errors, incorrect message ID, and Manchester coding errors. When an error is detected the LFR goes back to sniff mode until the end of ON time completion, if ONMODE is set; or turns off until the start of the next scheduled sampling interval, if ONMODE is cleared. Because the MCU uses more power than the LFR module, it is desirable to keep the MCU in low power standby modes as much as possible. Therefore, the handling of these errors will be performed by the LFR and not require additional software processing by the MCU.

When the DECEN bit is clear, there is no monitoring on data. The MCU needs to poll the state of the LFDO bit and create its own decoding scheme within software on the detected signal. To be able to start the polling only when data are received, the carrier detection flag is enabled in data mode when DECEN = 0. During data reception, the auto-zero sequence is performed at each LFO period. The MCU needs also to determine the end of the telegram and turn off the LFR (LFEN = 0) during two LFO cycles before any other operations.

10.15.15 Continuous ON mode

In the Continuously ON mode, the LFR module will remain on continuously while the LFEN bit is set. The Continuously ON mode is controlled by setting the LFSTM[3:0] bits.

In the Continuously ON mode, if a signal is successfully processed by the digital, the LFR module will stop and restart automatically. The gap is 2-3 LFO periods. Also if TOGMOD bit is set, the LFR module will stop after the ON time cycle and re- start automatically, after having changed the CARMOD bit.

10.15.16 Initialization information

When power is applied to the MCU, the LFR must be initialized and configured before it can begin to receive LF messages. Several systems in the LFR require factory trimming to ensure operation within specified limits. After these trim values are written, they remain constant until the next MCU reset.

The application program must set up control bits and registers to configure the LFR to determine the structure of the message telegram, the input sensitivity, and other LFR

options. It is good practice to clear the flags in the LFS register before enabling interrupt sources in order to avoid any immediate interrupt requests.

10.15.17 LF receiver module register descriptions

10.15.17.1 LF control 1 register (LFCTL1)

Table 72. LF control 1 register (LFCTL1) (address $0020)

Bit	7	6	5	4	3	2	1	0
R	LFEN	0	CARMOD	0	IDSEL1	IDSEL2	SENS1	SENS0
W		SRES		—				
Reset	U	U	U	U	U	U	U	U
POR ($)	0	0	0	0	0	0	0	0
LFR Soft reset	0	U	0	0	0	0	0	0

Table 73. LFCTL1 register field descriptions

Field	Description
7 LFEN	LFEN – LF Block Enable This read-write control bit is used to enable or disable the LF receiver. Once this bit is set the LFR will go through a power-up sequence that starts on the next rising edge of the LFO clock. The first complete cycle of the LFO is used to power up the LFR circuits. Following this startup time the auto-zero sequence is performed for 64 µs and then the LFR is ready to receive signals. 0 = LF receiver in standby; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = LF receiver active
6 SRES	SRES- Soft Reset of LF Block This read/write bit controls the soft reset of the LFR. The bit is self-reset and always reads as a logical zero. 0 = Reset completed 1 = Start a soft reset.
5 CARMOD	CARMOD – Carrier Mode This read/write control bit selects the basic operating mode for the LFR. 0 = Data receive mode; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = Carrier detect mode - wake the MCU when a carrier signal is detected if LFCDIE is set.
3:2 IDSEL[1:0]	IDSEL[1:0] – Wake-up ID Selection The two bits IDSEL[1:0] selects the existence and length of the wake-up ID. Reset clears these bits. 0 0 = No ID expected; Result of power on or LFR reset. Existing state remains after all other reset types. 0 1 = 8-bit ID based on the contents of the LFIDL register 1 0 = 16-bit ID based on the contents of the LFIDH and LFIDL registers 1 1 = 8-bit ID matches the contents of either the LFIDH or LFIDL registers

Table 73. LFCTL1 register field descriptions*...continued*

Field	Description
1:0 SENS[1:0]	SENS[1:0] – Sensitivity Selection The two bits SENS[1:0] select the sensitivity thresholds for the LFR input. These thresholds apply to the detection portion of a message. If the input level is below the SNODET_x level, no signal will be detected. If the level is above SDET_x, the signal will be detected. Sensitivity settings are only used in the carrier detect path and do not affect reception of the message body. 0 0 = Very Low sensitivity (S_{DET_VL}; S_{NODET_VL}); Result of power on or LFR reset. Existing state remains after all other reset types. 0 1 = Low sensitivity (S_{DET_L}; S_{NODET_L}) 1 0 = High sensitivity (S_{DET_H}; S_{NODET_H}) 1 1 = Very High sensitivity (S_{DET_VH}; S_{NODET_VH})

10.15.17.2 LF control 2 register (LFCTL2)

Table 74. LF control 2 register (LFCTL2) (address $0021)

Bit	7	6	5	4	3	2	1	0
R	LFSTM3	LFSTM2	LFSTM1	LFSTM0	LFONTM3	LFONTM2	LFONTM1	LFONTM0
W								
Reset	U	U	U	U	U	U	U	U
POR ($60)	0	1	1	0	0	0	0	0
LFR Soft Reset ($60)	0	1	1	0	0	0	0	0

Table 75. LFCTL2 register field descriptions

Field	Description
7:4 LFSTM[3:0]	LFSTM[3:0] – LF Sampling Time Interval Selection The four bits LFSTM[3:0] select the length of time between when the LFR input detector is turned on as set by the LFONTM bits in LFCTL2 register. The initial sampling interval starts with the LFO clock following a write to these bits. 0 1 1 0 = Result of power on or LFR reset. Existing state remains after all other reset types. See Table 76 for all states.
3:0 LFONTM[3:0]	LFONTM[3:0] – LF Sampling On Time Selection The four bits LFONTM[3:0] select the length of time that the LFR input detector is turned on at the beginning of each sampling interval set by the LFSTM bits. This ON time is the net sampling time with any initialization time (maximum of 2 ms) included in the OFF time prior to the sample ON time. If a signal is successfully detected, the length of time the detector remains ON depends on the operating mode. In carrier detect mode (CARMOD = 1) the detector will be turned off early if the evaluation of the carrier signal is completed before the end of the scheduled ON time. In data receive mode (CARMOD = 0) the detector will remain ON until the end of the message, an error is detected or timeout occurrence. The LFONTM selected time must be less than the LFSTM selected time, otherwise the Continuously ON mode is present. 0 0 0 0 = Result of power on or LFR reset. Existing state remains after all other reset types. See Table 77 for all states.

Table 76. LF sampling time interval selection

LFSTM[3:0]	Clock Cycles	~ Time ms
0 0 0 0	Continuous ON	
0 0 0 1	16	16
0 0 1 0	32	32
0 0 1 1	64	64
0 1 0 0	128	128
0 1 0 1	256	256
0 1 1 0	512	512
0 1 1 1	1024	1024
1 0 0 0	2048	2048
1 0 0 1	4096	4096
1 0 1 0 — 1 1 1 1	Continuous ON	

Table 77. LF sampling on time selection

LFONTM[3:0]	Clock Cycles	~ Time ms
0 0 0 0	1	1
0 0 0 1	2	2
0 0 1 0	4	4
0 0 1 1	8	8
0 1 0 0	16	16
0 1 0 1	32	32
0 1 1 0	64	64
0 1 1 1	128	128
1 0 0 0	256	256
1 0 0 1	512	512
1 0 1 0 — 1 1 1 1	1024	1024

10.15.17.3 LF control 3 register (LFCTL3)

Table 78. LF control 3 register (LFCTL3) (address $0022)

Bit	7	6	5	4	3	2	1	0
R	LFDO	TOGMOD	SYNC1	SYNC0	LFCDTM3	LFCDTM2	LFCDTM1	LFCDTM0
W	—							
Reset	U	U	U	U	U	U	U	U
POR ($32)	0	0	1	1	0	0	1	0
LFR Soft Reset	U	0	1	1	0	0	1	0

Table 79. LFCTL3 register field descriptions

Field	Description
7 LFDO	LFDO – LF Detector Output This read-only bit follows the bit slicer output signal that goes high during the presence of a carrier. It may change at any time. 0 = LF detector output low (no signal above threshold); Result of power-on reset. Existing state remains after all other types of reset. 1 = LF detector output high (received signal above threshold)
6 TOGMOD	TOGMOD – LFR Mode Toggle This read/write bit enables the toggling of the CARMOD bit at each new LFON sequence. Reset clears this bit. Therefore the reception chain will alternately look for a carrier frame or for a data frame. 0 = CARMOD bit does not change and determines detector mode; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = CARMOD bit will be toggled every LFON detection sequence, starting by CARMOD selection.
5:4 SYNC[1:0]	SYNC[1:0] – LF Synchronization Patter Selection The two bits SYNC[1:0] selects the type of SYNC pattern. Reset presets these bits to the 11 (9T SYNC) option. Compatible with preamble consisting of minimum 2 ms Manchester data to allow for proper averaging filter operation. 0 0 = For factory test purposes, not intended for use in any application. 0 1 = 6T SYNC pattern 1 0 = 7.5T SYNC pattern 1 1 = 9T SYNC pattern; Result of power on or LFR reset. Existing state remains after all other reset types.
3:0 LFCDTM[3:0]	LFCDTM[3:0] – LF Carrier Detect Time The 4 bits LFCDTM[3:0] select the length of time which the LFR input detector must detect a carrier before validating it. In carrier mode (CARMOD = 1), if the carrier is active for at least the time selected by the LFCDTM[3:0] bits and the LFCC counter value is reached, the LFCDF flag in the LFS register will be set; and if the LFCDIE control bit is also set, the MCU will be interrupted (wake-up). In the data receive mode (CARMOD = 0) the LFCDTM[3:0] bits select the length of time which the LFR input detector must detect a carrier before the effective receive chain is powered on. Once the carrier has been validated the LFCDTM[3:0] bits ignored during the decode of the rest of the data. 0 0 1 0 = Result of power on or LFR reset. Existing state remains after all other reset types. See Table 80 for additional states.

Table 80. LF carrier and data detect states

	Carrier detect		Data detect	
LFCDTM[3:0]	Clock Cycles	~ Time µs	Clock Cycles	~ Time µs
0 0 0 0	8	64	8	64
0 0 0 1	16	128	8	64
0 0 1 0	32	256	8	64
0 0 1 1	64	512	8	64
0 1 0 0	128	1024	8	64
0 1 0 1	256	2048	8	64
0 1 1 0	512	4096	8	64
0 1 1 1	1024	8192	8	64
1 0 0 0	8	64	8	64

Table 80. LF carrier and data detect states*...continued*

	Carrier detect		Data detect	
LFCDTM[3:0]	**Clock Cycles**	**~ Time µs**	**Clock Cycles**	**~ Time µs**
1 0 0 1	16	128	16	128
1 0 1 0	32	256	32	256
1 0 1 1	64	512	64	512
1 1 0 0	128	1024	128	1024
1 1 0 1	256	2048	256	2048
1 1 1 0	512	4096	512	4096
1 1 1 1	1024	8192	1024	8192

10.15.17.4 LF control 4 register (LFCTL4)

Table 81. LF control 4 register (LFCTL4) (address $0023)

Bit	7	6	5	4	3	2	1	0
R	LFDRIE	LFERIE	LFCDIE	LFIDIE	DCEN	VALEN	TIMOUT1	TIMOUT0
W								
Reset	U	U	U	U	U	U	U	U
POR ($0F)	0	0	0	0	1	1	1	1
LFR Soft Reset ($0F)	0	0	0	0	1	1	1	1

Table 82. LFCTL4 register field descriptions

Field	Description
7 LFDRIE	LFDRIE – LFR Data Register Full Interrupt Enable This read/write bit enables interrupts to be requested when the LFR data register is full. 0 = LFDRF interrupts disabled. Use software polling; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = LFR Data Register Full interrupts enabled. If LFDRIE = 1, then interrupt is requested when LFDRF = 1.
6 LFERIE	LFERIE – LFR Error Interrupt Enable This read/write bit enables interrupts to be requested when the LFR detects an error in reception of a non-Manchester encoded bit time following the SYNC time, or if when a sampling error is detected, or when the ID is not matched. 0 = LFERF interrupts disabled. Use software polling; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = LFERF interrupts are enabled. If LFERIE is set, then an interrupt is requested when LFERF = 1.
5 LFCDIE	LFCDIE - LFR Carrier Detect Interrupt Enable This read/write bit enables interrupts to be requested when the LFCD flag rises. 0 = LFCDF interrupts disabled. Use software polling; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = LFCDF interrupts are enabled. If LFCDIE is set, then an interrupt is requested when LFCDF = 1.

Table 82. LFCTL4 register field descriptions*...continued*

Field	Description
4 LFIDIE	LFIDIE – LFR ID Detect Interrupt Enable This read/write bit enables interrupts to be requested when the LFR detects a match to the wake-up ID code selected in the LFIDH:L registers. 0 = LFIDF interrupts disabled; Use software polling; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = LFIDF interrupts are enabled. If LFIDIE is set, then an interrupt is requested when LFIDF = 1.
3 DCEN	DCEN – LF Digital Decode Enable This read/write bit enables the data processing by the digital decoder. When disabled, the frame format (Manchester, data-rate, SYNC, data) is not checked. There is no more error flag assertion (data, error, ID, etc.). The MCU should then poll the LFDO bit to extract from the analog detector the bit stream. 0 = Digital decoder is disabled. 1 = Digital decoder is enabled; Result of power on or LFR reset. Existing state remains after all other reset types.
2 VALEN	VALEN – LF Carrier Validation Enable This read/write bit enables the carrier validation process. 0 = Carrier Validation disabled. 1 = Carrier Validation enabled; Result of power on or LFR reset. Existing state remains after all other reset types.
1:0 TIMOUT[1:0]	TIMOUT[1:0] – Synchronization Time out Selection The two bits TIMOUT[1:0] select the period of time that the LFR will search for a SYNC pattern in the data mode. If the SYNC pattern is not detected the LFR will be turned off after this delay time. These time intervals are clocked by the internal LFRO clock. 0 0 = SYNC word is continuously searched — no timeout. 0 1 = SYNC search time set to nominal 8 milliseconds. 1 0 = SYNC search time set to nominal 24 milliseconds. 1 1 = SYNC search time set to nominal 48 milliseconds; Result of power on or LFR reset. Existing state remains after all other reset types.

10.15.17.5 LF receiver status register (LFS)

Table 83. LF receiver status register (LFS) (address $0024)

Bit	7	6	5	4	3	2	1	0
R	LFDRF	LFERF	LFCDF	LFIDF	LFOVF	LFEOMF	LPSM	0
W	—	—	—	—	—	—		LFIAK
Reset	U	U	U	U	U	U	U	U
POR ($02)	0	0	0	0	0	0	1	0
LFR Soft Reset ($02)	0	0	0	0	0	0	1	0

Table 84. LFS register field descriptions

Field	Description
7 LFDRF	LFDRF – LF Data Ready Flag This read-only status flag is set when a complete byte of data has been received by the LFR. An interrupt is sent to the MCU if the LFDRIE bit is set. Clear LFDRF by writing a one to the LFIAK bit or reading the LFDATA register. 0 = No new data in LFDATA register; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = A new byte of data has been received and can be read from the LFDATA register.
6 LFERF	LFERF – LF Receive Error Flag In data receive mode, this read-only status flag is set when a non-standard bit time is detected in the Manchester data mode, when an incorrect ID is received or when a sampling error has been detected. Any received data bits before the error occurs are placed in the data buffer. In carrier detect mode, this read-only status flag is not used and remains clear. An interrupt is sent to the MCU if the LFERIE bit is set. Clear LFERF by writing a one to the LFIAK bit. 0 = Normal operation; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = Error detected in the Manchester data mode, when an incorrect ID is received or when a sampling error has been detected.
5 LFCDF	LFCDF – LF Carrier Pulse Detect Flag In carrier detect mode, this read-only status flag is set when the number of consecutive carrier validations set by the LFCC bits in is reached. Note that the LFCC function is not working if TOGMOD = 1 or if CARMOD = 1. Clear LFCDF by writing a one to the LFIAK bit. 0 = Normal operation; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = Carrier detection has occurred.
4 LFIDF	LFIDF – LF ID Detect Flag In data receive mode, this read-only status flag is set when the received ID matches the stored value. This interrupt can be generated even if no data bits follow the ID. An interrupt is sent to the MCU if the LFIDIE bit is set. Clear LFIDF by writing a one to the LFIAK bit. 0 = Normal operation; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = wake-up ID has been detected.
3 LFOVF	LFOVF – LF Receive Data Overflow Flag In data receive mode, this read-only status flag is set when a complete byte of data has been received and written into the LFDATA register, but the previously received byte was not read from LFDATA register yet. This indicates that the MCU has lost the previously received data byte. In carrier detect mode, this read-only status flag is not used and remains cleared. No separate interrupt is generated by this specific flag bit because the LFDRF flag would serve that purpose. Clear LFOVF by writing a one to the LFIAK bit. 0 = Normal operation; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = Previous data over-written before MCU read it.
2 LFEOMF	LFEOMF – LF Receive Data End of Message Flag In data receive mode, this read-only status flag is set when a complete byte of data has been received and written into the LFDATA register and an end-of-message Manchester encoding error occurs. In carrier detect mode, this read-only status flag is not used and remains clear. No interrupt is generated by this flag bit because the LFERF flag would serve that purpose. Clear LFEOMF by writing a one to the LFIAK bit. 0 = No EOM detected; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = EOM detected.
1 LPSM	LPSM = Low Power Sniff Mode This bit used to activate the low power consumption during SNIFF mode. It saves approximately 1 µA with a trade-off of an additional 300 µs in transition from carrier to data mode. 0 = Low time transition from carrier to data mode 1 = Low consumption during sniff mode; Result of power on or LFR reset. Existing state remains after all other reset types.

Table 84. LFS register field descriptions*...continued*

Field	Description
0 LFIAK	LFIAK – LF Interrupt Acknowledge Control Writing a one to the LFIAK bit clears the LFDRF, LFERF, LFCDF, LFIDF, LFOVF and LFEOMF flag bits. When a one is written to the LFIAK, it is automatically cleared at the next positive edge of the MCU bus clock. Then, reading the LFIAK bit is allowed but will always return zero. Writing a zero the LFIAK bit has no effect. 0 = No effect; Result of power-on reset. Existing state remains after all other reset types. 1 = Write 1 to clear the LFDRF, LFERF, LFCDF, LFIDF, LFOVF and LFEOMF flag bits.

10.15.17.6 LF received data register (LFDATA)

Table 85. LF received data register (LFDATA) (address $0025)

Bit	7	6	5	4	3	2	1	0
R	LFRXD7	LFRXD6	LFRXD5	LFRXD4	LFRXD3	LFRXD2	LFRXD1	LFRXD0
W	—	—	—	—	—	—	—	—
Reset ($)	U	U	U	U	U	U	U	U
POR ($00)	0	0	0	0	0	0	0	0
LFR soft reset ($00)	0	0	0	0	0	0	0	0

Table 86. LFDATA register field descriptions

Field	Description
7:0 LFDATA	The LFDATA[7:0] is a read-only register that contains the most recent received data value. As data is serially received by the LFR, it is assembled into 8-bit values. When a new complete 8-bit value is received, it is moved into the LFDATA register, over-writing any previous value, and the LFDRF data ready flag is set to indicate a value is available for the MCU to read. If a previous value was ready but was not read out of the LFDATA register before a new data byte is ready, the LFOVF overflow flag is also set to indicate this overflow condition. Writes to LFDATA have no meaning or effect. $00 = Result of power on or LFR reset. Existing state remains after all other reset types.

10.15.17.7 LF receiver ID registers (LFID)

Table 87. LF receiver ID register (LFID) (address $0026)

Bit	7	6	5	4	3	2	1	0
R	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
W								
Reset	U	U	U	U	U	U	U	U
POR ($00)	0	0	0	0	0	0	0	0
LFR soft reset ($00)	0	0	0	0	0	0	0	0

Table 88. LF receiver ID register (LFID) (address $0027)

Bit	7	6	5	4	3	2	1	0
R W	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
Reset	U	U	U	U	U	U	U	U
POR ($00)	0	0	0	0	0	0	0	0
LFR soft reset ($00)	0	0	0	0	0	0	0	0

Table 89. LFID register field descriptions

Field	Description
7:0 ID	These two 8-bit read/write registers hold one of two ID values for LF messages. The type of ID checking can be selected or disabled by using the IDSEL[1:0] bits in the LFCTL1 register. When ID checking is enabled (LFIDIE = 1), the ID value received through the LFR must match the contents of the LFIDH and/or LFIDL registers depending on the IDSEL bits in order to generate the MCU wake-up and set the LFIDF flag. If the error detection interrupts are not enabled, LFERIE = 0, an ID mismatch will be ignored and the MCU will remain in standby mode to minimize power consumption. If the error detection interrupts are enabled, LFERIE = 1, an ID mismatch will trigger an interrupt to wake up the MCU, and set the set the LFERF flag. $0000 = Result of power on or LFR reset. Existing state remains after all other reset types.

10.15.17.8 LF receiver control E register (LFCTRLE)

Table 90. LF receiver control E register (LFCTRLE) (address $0028)

Bit	7	6	5	4	3	2	1	0
R W	reserved	reserved	reserved	reserved	TRIMEE	AZSC2	AZSC1	AZSC0
Reset	U	U	U	U	U	U	U	U
POR ($01)	0	0	0	0	0	0	0	1
LFR soft reset ($01)	0	0	0	0	0	0	0	1

Table 91. LFCTRLE register field descriptions

Field	Description
3 TRIMEE	TRIMEE – TRIM Edition Enable, controls write access to the TRIM1 and TRIM2 registers. 0 = TRIM registers are read-only mode; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = TRIM registers are read/write access mode.

Table 91. LFCTRLE register field descriptions*...continued*

Field	Description
2:0 AZSC[2:0]	AZSC[2:0] – Log Amp Auto-zero Sequencer Control The 3 bits AZSC[2:0] control the AZ and trim within the LOGAMP. x 0 0 = Nominal AZ sequence - recommended setting x 0 1 = Short amp output release, max delay with Rectifiers x 1 0 = Short amp output release, max delay with Amp input x 1 1 = All short, max delay with end of AZ 0 x x = Nominal sensitivity trim - recommended setting 1 x x =Sensitivities shifted by - 4 trim steps 0 0 1 = Result of power on or LFR reset. Existing state remains after all other reset types.

10.15.17.9 LF receiver control D register (LFCTRLD)

Table 92. LF receiver control D register (LFCTRLD) (address $0029)

Bit	7	6	5	4	3	2	1	0
R	AVFOF1	AVFOF0	DEQS	AZDC1	AZDC0	ONMODE	CH125K1	CH125K0
W			—					
Reset	U	U	U	U	U	U	U	U
POR ($01)	0	0	0	0	0	0	0	1
LFR soft reset	0	0	U	0	0	0	0	1

Table 93. LFCTRLD register field descriptions

Field	Description
7:6 AVFOF[1:0]	AVFOF[1:0] – Auto Zero Release Delay The two bits AVFOF[1:0] control the delay between falling edge of the SUM d_az_en input and falling edge of internal AZ control line. 0 0 = No delay; Result of power on or LFR reset. Existing state remains after all other reset types. 0 1 = No delay 1 0 = One-half of 125 kHz clock period delay - recommended setting 1 1 = One and one-half of 125 kHz clock periods delay
5 DEQS	DEQS – DeQing Status Flag This read-only status bit allows the reading of the effective activation of the DeQing System. 0 = DeQing system not activated; Result of power-on reset. Existing state remains after all other reset types. 1 = DeQing system activated
4:3 AZDC[1:0]	AZDC[1:0] – Auto Zero Triggering Control In data receive mode, the two bits AZDC[1:0] control the triggering of AZ sequence with respect to both LFCPTAZ value (ref. LFCTRLB register) and the state of the demodulation input data state. 0 0 = AZ starts after LFCPTAZ numbers of input data edges; Result of power on or LFR reset. Existing state remains after all other reset types. 0 1 = Z starts randomly adding -1, 0 or 1 to LFCPTAZ value between each AZ. 1 0 = AZ starts after LFCPTAZ numbers of input data edges and when the input data (d_data) state is 0. 1 1 = AZ starts after LFCPTAZ numbers of input data edges and when the input data (d_data) state is 1 - recommended setting.

Table 93. LFCTRLD register field descriptions*...continued*

Field	Description
2 ONMODE	ONMODE – On Behavior Mode Control This read/write bit selects how an error will affect the ON time. 0 = Any error will stop the ON time; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = If remaining ON time, the LFR will go back to sniff mode at any error - recommended setting.
1:0 CHK125[1:0]	CHK125[1:0] – Carrier Check The two bits CHK125[1:0] control the CARVAL carrier validation frequency checking method. 0 0 = the carrier is validated on n x (2 x 32 µs packets), where n is depending on the LFCDTM value – recommended setting for Low Sensitivity mode 0 1 = the carrier is validated on n x (8 x 8 µs packets), where n is depending on the LFCDTM value – recommended setting for High Sensitivity mode; Result of power on or LFR reset. Existing state remains after all other reset types. 1 0 = same performance as 0 1 1 1 = same performance as 0 0

10.15.17.10 LF receiver control C register (LFCTRLC)

Table 94. LF receiver control C register (LFCTRLC) (address $002A)

Bit	7	6	5	4	3	2	1	0
R **W**	AMPGAIN1	AMPGAIN0	FINSEL1	FINSEL0	AZEN	LOWQ1	LOWQ0	DEQEN
Reset ($)	U	U	U	U	U	U	U	U
POR ($C8)	1	1	0	0	1	0	0	0
LFR soft reset ($C8)	1	1	0	0	1	0	0	0

Table 95. LFCTRLC register field descriptions

Field	Description
7:6 AMPG AIN[1:0]	AMPGAIN[1:0] – Third stage Amplifier Gain Control These bits control the third amplifier gain. 0 0 = Gain of ~2.2 - recommended setting 0 1 = Gain of ~3.8 1 0 = Gain of ~4.9 1 1 = Gain of ~6.6; Result of power on or LFR reset. Existing state remains after all other reset types.
5:4 FINSEL[1:0]	FINSEL[1:0] – Final stage Amplifier Gain Control These bits select the final stage of the LOGAMP. 0 0 = Continuous time biasing - Fixed Gain 6; Result of power on or LFR reset. Existing state remains after all other reset types. 0 1 = Continuous time biasing - Programmable Gain - recommended setting 1 0 = Fourth rectifier disabled 1 1 = Fourth rectifier disabled

Table 95. LFCTRLC register field descriptions*...continued*

Field	Description
3 AZEN	AZEN – Data Auto Zero Enable This bit allows the AZ sequence during data frame. 0 = AZ during data disabled; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = AZ during data enabled - recommended setting
2:1 LOWQ[1:0]	LOWQ[1:0] – DeQing Resistor Control These bits select the resistor added in parallel to the input network. 0 0 = ~4 kΩ; Result of power on or LFR reset. Existing state remains after all other reset types. 0 1 = ~2 kΩ 1 0 = ~1 kΩ 1 1 = ~500 ohm
0 DEQEN	DEQEN – DeQing Enable Control The bit controls the DeQing system. 0 = DeQing disabled; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = DeQing enabled.

10.15.17.11 LF receiver control B register (LFCTRLB)

Table 96. LF receiver control B register (LFCTRLB) (address $002B)

Bit	7	6	5	4	3	2	1	0
R W	HYST1	HYST0	LFFAF	LFCAF	LFPOL	LFCPTAZ2	LFCTPAZ1	LFCTPAZ0
Reset	U	U	U	U	U	U	U	U
POR ($C4)	1	1	0	0	0	1	0	0
LFR soft reset ($C4)	1	1	0	0	0	1	0	0

Table 97. LFCTRLB register field descriptions

Field	Description
7:6 HYST[1:0]	HYST[1:0] – Slicer Hysteresis Control The two bits HYST[1:0] control the data slicer hysteresis 0 0 = 20 mV 0 1 = 40 mV 1 0 = 50 mV 1 1 = 30 mV – recommended setting; Result of power on or LFR reset. Existing state remains after all other reset types.
5:4 LFFAF: LFCAF	LFFAF:LFCAF – Average filter bi-phase filter control The two bits LFFAF:LFCAF activate bi-phase filtering and control offset value 0 0 = Standard low pass filtering activated - recommended setting; Result of power on or LFR reset. Existing state remains after all other reset types. 0 1 = Standard low pass filtering activated 1 0 = Bi-phase filtering activated - Low offset from input signal low level 1 1 = Bi-phase filtering activated - High offset from input signal low level

Table 97. LFCTRLB register field descriptions*...continued*

Field	Description
3 LFPOL	LFPOL – LF Manchester Polarity Selection This read/write bit selects the polarity of the transition in the middle of the bit time. The LFPOL is not used in Carrier mode. 0 = Zero is falling edge in middle of a bit time, one is a rising edge in the middle of bit time; Result of power on or LFR reset. Existing state remains after all other reset types. 1 = Zero is rising edge in middle of a bit time, one is a falling edge in the middle of bit time.
2:0 LFCPT AZ[2:0]	LFCPTAZ[2:0] – LF Auto Zero Counter Selection Applications to set these bits to 1 1 0 for proper LF operation. These bits tune the minimum number of data edges between two auto-zero requests during a data frame. 1 0 0 = Result of power on or LFR reset. Existing state remains after all other reset types.

10.15.17.12 LF receiver control A register (LFCTRLA)

Table 98. LF receiver control A register (LFCTRLA) (address $002C)

Bit	7	6	5	4	3	2	1	0
R	reserved	reserved	reserved	reserved	LFCC3	LFCC2	LFCC1	LFCC0
W								
Reset	U	U	U	U	U	U	U	U
POR ($00)	0	0	0	0	0	0	0	0
LFR soft reset ($00)	0	0	0	0	0	0	0	0

Table 99. LFCTRLA register field descriptions

Field	Description
3:0 LFCC[3:0]	LFCC[3:0] – LF Successive Carrier Validations Counter Results The value of the LFCC[3:0] bits define how many times the carrier detect sample ON time detected an LF carrier signal before the LFCDF flag bit set. The flag will be risen when the number of ON samples with a detected carrier greater than the LFCDTM[3:0] reaches the value of the LFCC[3:0] bits plus one. The internal count of detected carrier pulses will increment the count as long as they are consecutive samples. When a sample is encountered without any detected carrier the count will be reset. The LFCC register is considered reset in data mode. The first carrier validation will lead to startup of the receiver chain. This feature allows the user to define a number of consecutive carrier detections are required before the flag is risen; and is useful in detecting long duration carrier pulses. This counter is disabled if CARMOD = 0 or TOGMOD = 1. 0 0 0 0 = Result of power on or LFR reset. Existing state remains after all other reset types.

10.16 Radio frequency (RF) transmitter module

It is not intended that the RFM may be actively powered up and/or transmitting RF data while physical parameter measurements are being made; or during the time that the LFR may be actively receiving/decoding LF signals. The resulting interactions will degrade the performance of the RF output spectrum. The RF module (RFM) consists of an external crystal-driven oscillator, VCO, fractal-n PLL and RF output amplifier (PA) for an antenna. It also contains a small state machine controller, random time generator and hardware data buffer for automated output or direct control from the MCU.

The RF transmitter module has the following features:

- Interframe
  - Interframe generated by a combination of BASE, INCR & RAND parameters
  - Incorporates an 7bit LFSR PRBS (random) generator
  - Used to randomize the time interval between repeated TX sequences
- Frac-N MASH3 Modulator
  - 4-bit integer (Fixed) and 13-bit fractional range
- VCO Calibration Machine
  - Automatic Calibration with every power up sequence
  - Possible to override of calibration result by writing directly to sub-band register
- Power Up/Dn state machine
  - Automatic power up/down sequencer
  - Manual control available
- Register Map
  - Fully accessible read/write 8-bit wide register map
  - Protected register space for non-user functionality
- TX Buffer
  - 32 byte / 256 bit Transmit buffer
  - End of transmission Interrupt
  - Encodes and assembles data stream for transmission (Manchester, Bi-Phase, NRZ)

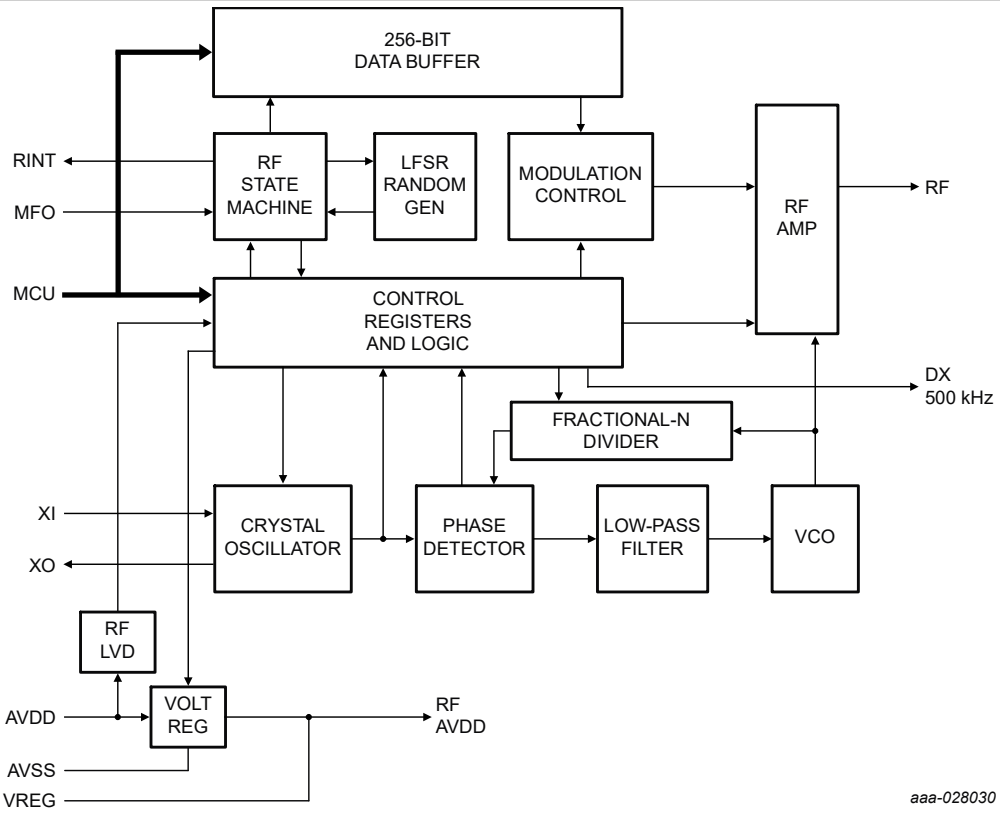


Figure 29. Radio frequency module block diagram

10.16.1 RF data modes

There are two modes of operation in using the RF output in either the data buffer mode or MCU direct mode.

10.16.1.1 RF data buffer mode

In the RF data buffer mode the transmissions are sent by dedicated logic hardware while the MCU can be put into a low power mode until the transmission is completed. This RF state machine is clocked by the MFO which is enabled when the SEND bit is set and when any of the LFR, SMI or MCU are operating.

The RF data buffer consists of a dedicated RFM state machine and a 256-bit data buffer. The RF data buffer is loaded with whatever data pattern the user software creates. The number of data bits to be sent is selected by the FRM[7:0] control bits. The control logic is triggered by the SEND control bit when it is time to transmit the data which is sent to the RF stage after being encoded as either Manchester, Bi-Phase or NRZ data according to the method selected by the CODE[1:0] bits.

Before the data can be transmitted the RFM control logic enables the external crystal oscillator and phase-locked-loop to initialize before the RF output stage can begin transmission.

The external crystal connected to the X0 and XI pins provides the carrier frequency as well as the data rate clock needed for the data rates associated with the OOK or FSK modulation. Therefore, the tolerance on the data rate will depend on the characteristics of the external crystal.

Once the data buffer is emptied the data transfer stops; the RF output stage is turned off; and the SEND control bit is cleared and an interrupt of the MCU may be generated to wake it from the STOP1 mode. The user can test that the transmission has completed by reading back the state of the SEND control bit or the RFIF status bit.

There is also the option to send the same data frame from 1 to 16 times with interlaced time intervals when the RF transmitter PA output stage is off. If multiple frames of data are to be transmitted within a datagram the spacing before the first frame and between subsequent frames can be controlled by the RFM state machine in several ways:

1. Use of a programmable timer (random, base time, time adder).
2. No time delays.

In addition, the RFM crystal oscillator, VCO and PLL can be turned off during any interframe timing by use of the IFPD bit.

When using the data buffer mode the user’s software should not change any bits in the RFM registers after the SEND has been set and the transmission is still in progress. Changing RFM register contents during a transmission can lead to data faults or errors.

10.16.1.2 MCU direct mode

When the CODE[1:0] bits are both set the encoding is controlled directly by the MCU where the data to the RF output depends on the state of the DATA bit and the selected modulation scheme. In this mode the user software must control the RF output stage to power up (using the SEND control bit), WAIT for the RF output stage to stabilize (monitor the RCTS status bit) and clock the DATA to the RF output stage. In this mode the data rate and its stability will depend on the internal HFO oscillator.

Any transfers of data from the MCU will use the DATA bit which will be reflected as modulated data on the RF pin once the RF output stage is set up to transmit. The maximum data rate in this mode will depend on the complexity of the user software and the MCU clock rate.

The POL bit in this case simply inverts the state of the DATA bit before it drives the RF output stage.

The accuracy of the data rate in the MCU direct mode is directly dependent on the HFO accuracy.

10.16.2 RF output buffer data frame

When using the RF data buffer mode each frame of data is sent as 2 to 256 bits per frame with a possible two trailing bits for an end-of-message, EOM, as shown in Figure 30. The actual data being transmitted in a given data frame and any combinations of data frames into a single datagram is dependent on the user software.

The number of frames sent in a given datagram can be from 1 to 16 based on the FNUM[3:0] bits in the RFCR3. The 256-bit buffer is divided into two pages of 128 bits as selected by the RPAGE bit in the RFCR2.

The data buffer is unloaded to the RF output starting with the least significant bit (RFD0) in the least significant byte (RFB0) up through the most significant bit (RFD127) in the most significant byte (RFB15). This is often referred to as "little-endian" data ordering.

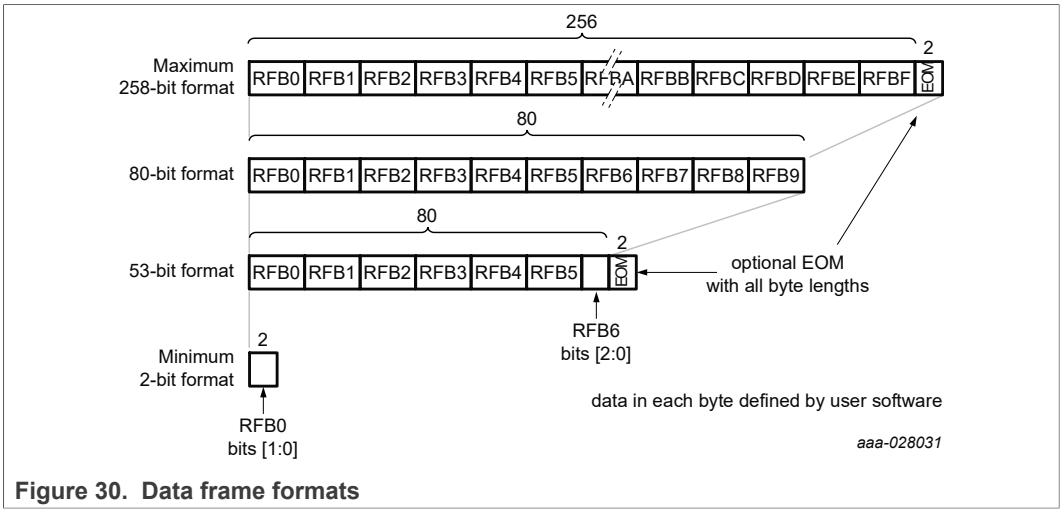


Figure 30. Data frame formats

10.16.2.1 Data buffer length

The number of bits sent in a given transmission frame is selected by the FRM[7:0] control bits encoded as a direct binary number plus one. This gives a range of 2 through 256 bits. Data written to data buffer bits above the highest bit number will be ignored. Transmission always begins with the data written in the RFB0 location. When the requested number of bits have been transmitted an interrupt to the MCU can be generated if the RFIE bit is set.

10.16.2.2 End of Message (EOM)

If the EOM control bit is set, then at the end of the data frame there will be carrier for a period of two bit times at level high for the OOK modulation modes or f_{DATA1} for the FSK modulation modes. Following the EOM period there will be no carrier for either

the OOK or FSK modes. If the EOM control bit is clear, no EOM period is added to the transmission.

10.16.3 RF transmission randomization

When there are two or more different transmitters, the clock rates of each may drift into synchronism with each other; and there is the possibility of RF data collisions and the loss of data from both transmitters. In order to reduce possible RF data collisions each transmission will contain from 1 to 16 frames of data. Each frame may be spaced at after the initially timed transmission start time and between any two data frames.

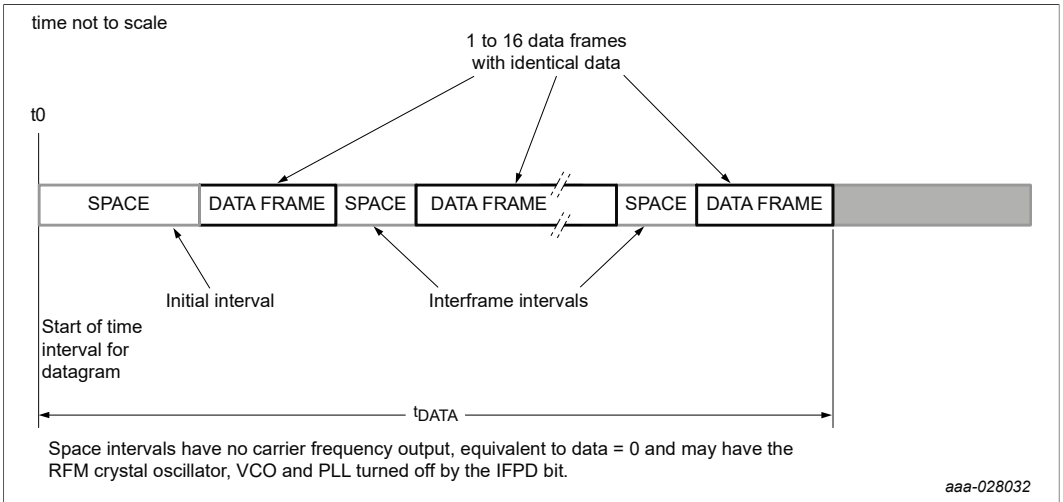


Figure 31. Radio frequency transmission randomization

The generation of the initial and interframe time intervals can be done with a combination of a programmable counter, a pseudo-random interval generator and a frame counter. The initial time interval can be done by adjusting the start time using the MCU or using this interval timing generator.

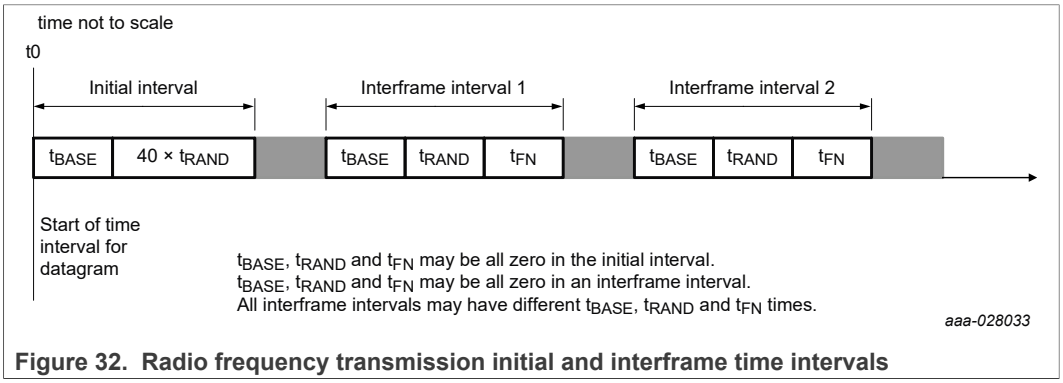


Figure 32. Radio frequency transmission initial and interframe time intervals

10.16.3.1 Initial time interval

When generating an initial time interval, the MCU loads the RFM interval generator variables and enters the STOP1 mode. When the initial time interval ends, the data in the RFM data buffer is automatically sent and the MCU will wake at the end of the transmission. The initial time interval is made up of two components:

$t_{INIT} = t_{BASE} + (40 \times t_{RAND})$

Where:

t_{INIT} = Total time interval before first frame is transmitted in ms

t_{BASE} = Base time in ms; ≤5 ms not recommended

t_{RAND} = Pseudo-random time in ms based on a Galois 7-bit LFSR

10.16.3.2 Interframe time intervals

When generating an interframe time interval the MCU loads the RFM interval generator variables and then goes to the STOP1 mode. When the interframe time interval ends the data in the RFM data buffer is automatically sent and the MCU will wake at the end of the transmission. The interframe time interval is made up of three components:

$t_{IFRM} = t_{BASE} + t_{RAND} + t_{FN}$

Where:

t_{IFRM} = Total time interval between each transmitted frame in ms

t_{BASE} = Base time in ms; ≤5 ms not recommended

t_{RAND} = Time adder in ms for frame number

t_{FN} = Pseudo-random time in ms based on a Galois 7-bit LFSR

10.16.3.3 Base time interval

The base time interval, t_{BASE}, is used in the initial time interval and in datagram transmissions with two or more frames. The programmable frame space interval is based on a simple 8-bit, count-down timer as described by the RFBT[7:0] control bits in the RFCR4 register. This time interval is forced to zero when the RFBT[7:0] are all clear. The range of the base time must be set to 0 or between 5 and 255 ms using a clock generated from the MFO divided by 125.

10.16.3.4 Pseudo-random time interval

The pseudo-random time interval, t_{RAND}, is used both in the initial and the interframe time intervals if the LFSR[6;0] bits are set to something other than all zeros. When the ISPC bit is set the pseudo-random initial time interval before the first data frame will be 40 times the value of t_{RAND}. When the LFSR[6:0] bits are used the t_{RAND} time will vary based on a pseudo-random generated binary number using a Galois linear feedback shift register (LFSR) implemented using the primitive polynomial for a 7-stage register.

This LFSR creates a sequence of 127 binary numbers including $01 through $3F which are each repeated only once in each sequence of 127 clocks of the shift register. The LFSR is initialized to $40 during power up of the device. When a random interval is to be determined the contents of the LFSR are sampled as the “random number” for calculating the required interval time. Following the use of the random interval the LFSR is clocked once to advance it to the next pseudo-random number. The range of the pseudo-random time is 1 to 127 ms using a clock generated from the MFO divided by 125. The current value of the LFSR can be changed and/or read by the LFSR[6:0] bits in the RFCR5 register.

Note: *The LFSR bits in RFCR5 are the seed and not the current LFSR random number, which is not accessible.*

The range of the pseudo-random time is 1 to 127 ms using a clock generated from the MFO divided by 125. The current value of the LFSR can be changed and/or read by the LFSR[6:0] bits in the RFCR5 register.

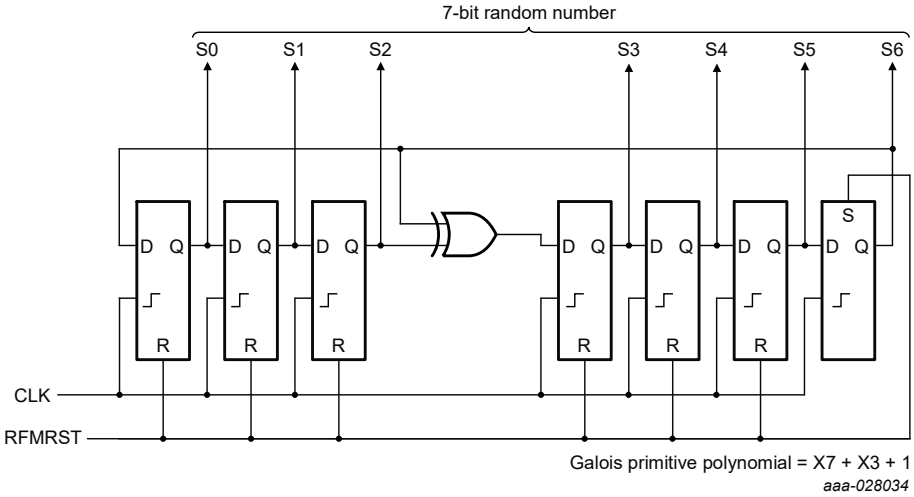


Figure 33. LFSR implementation

A value of all zeros in the LFSR will remain unchanged with every clock input and cannot be used as a starting "seed."

The resulting range of times for the initial and interframe pseudo-random time will be as given in Table 100 for both the design center and the variation resulting from the tolerance of the MFO clock.

Table 100. Randomization interval times

Time Interval	Randomization Number	Ideal Time Interval (ms)	Time Interval Including MFO Tolerance (ms)	
			Minimum	Maximum
Initial	1	40	37.2	42.8
	127	5080	4347.2	5434.6
Interframe	1	1	0.93	1.07
	127	127	118.1	135.9

10.16.3.5 Frame number time

The frame number time, t_{FN}, is only used between frames and is based on a selectable time from 0 to 63 ms and the number of the frame that was just transmitted as given in Table 101. If the frame number time is not used, the value of the selected time should be set to zero. The maximum number of frames is defined by the FNUM[3:0] control bits.

The range of the frame number time is a multiple of 0 to 63 ms using a clock generated from the MFO divided by 125. The value of this time multiple can be changed by the RFFT[5:0] bits in the RFCR6 register. The BPS[7:0] control bits are set to $34 by the RFMRST signal which results in a default data rate of 9434 bits/sec.

Table 101. Frame number interval times

Value of FNUM[3:0]	Number of frames	Frame interval where time added	Nominal frame number time interval added (ms)	
			Minimum	Maximum
0	1	None	n/a	n/a
1	2	1 - 2	1	63
2	3	2 - 3	2	126
3	4	3 - 4	3	189
4	5	4 - 5	4	252
5	6	5 - 6	5	315
6	7	6 - 7	6	378
7	8	7 - 8	7	441
8	9	8 - 9	8	504
9	10	9 - 10	9	567
10	11	10 - 11	10	630
11	12	11 - 12	11	693
12	13	12 - 13	12	756
13	14	13 - 14	13	819
14	15	14 - 15	14	882
15	16	15 - 16	15	945

10.16.4 RFM in STOP1 mode

The entire RF transmitter digital section can remain powered up, if enabled by the RFEN bit (see Section 10.13 "Timer pulse-width module"), when the MCU goes into the STOP1 mode.

10.16.5 Data encoding

The CODE[1:0] control bits select either Manchester, Bi-Phase, NRZ or MCU direct data encoding of each data bit being transferred from the RF data buffer to the RF output stage. Further, the polarity of the selected encoding method can be inverted using the POL control bit.

10.16.5.1 Manchester encoding

When the CODE[1:0] bits are both clear the data is Manchester encoded format, with data transmitted as a transition in voltage occurring in the middle of the bit time. The polarity of this transition is selected by the POL bit. When the POL bit is cleared, then a logical LOW is defined as an increase in signal in the middle of a bit time and a logical HIGH is defined as a decrease in signal in the middle of a bit time as shown in Figure 34. When the POL bit is set, then a logical LOW is defined as a decrease in signal in the middle of a bit time and a logical HIGH is defined as an increase in signal in the middle of a bit time as shown in Figure 35. Since there is always a transition in the middle of the bit time there must also be a transition at the start of a bit time if consecutive "1" or "0" data are present.

10.16.5.2 Bi-Phase encoding

When the CODE[1:0] bits are 0:1 then the data is Bi-Phase encoded format, with data transmitted as the presence or absence of a transition in signal in the middle of the bit time. The polarity of this transition is selected by the POL bit. Unlike Manchester coding there is always a signal transition at the boundaries of each bit time. When the POL bit is cleared, then a logical HIGH is defined as no change in signal in the middle of a bit time and a logical LOW is defined as a change in the signal in the middle of a bit time as shown in Figure 36. When the POL bit is set, then a logical HIGH is defined as a change in signal in the middle of a bit time and a logical LOW is defined as no change in the signal in the middle of a bit time as shown in Figure 37. Since there is always a transition at the ends of the bit time consecutive bits of the same state may have two signal states (high or low) during the middle of the bit time.

10.16.5.3 NRZ encoding

When the CODE[1:0] bits are 1:0 then the data is NRZ encoded format, with data transmitted as either a high or low for the complete bit time. The polarity of this state is selected by the POL bit. The Manchester and Bi-Phase encoding can actually be created using NRZ encoding running at twice the desired data rate.

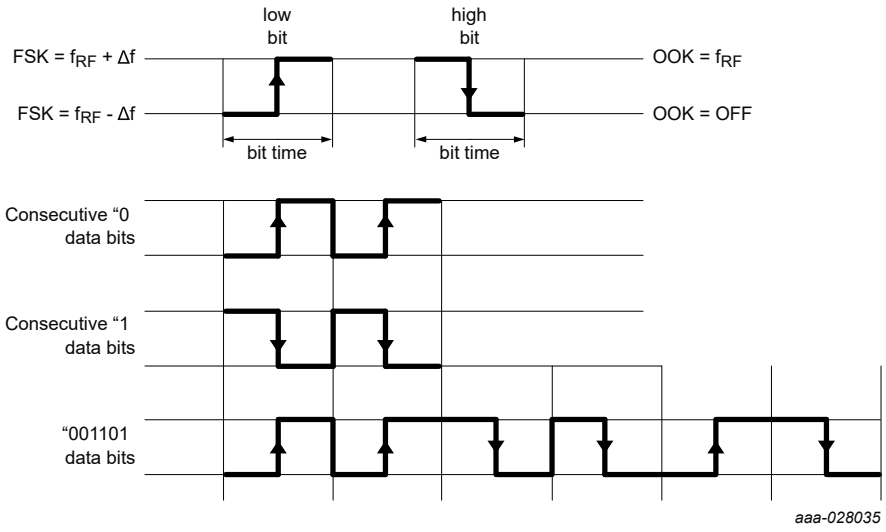


Figure 34. Manchester data bit encoding (POL = 0)

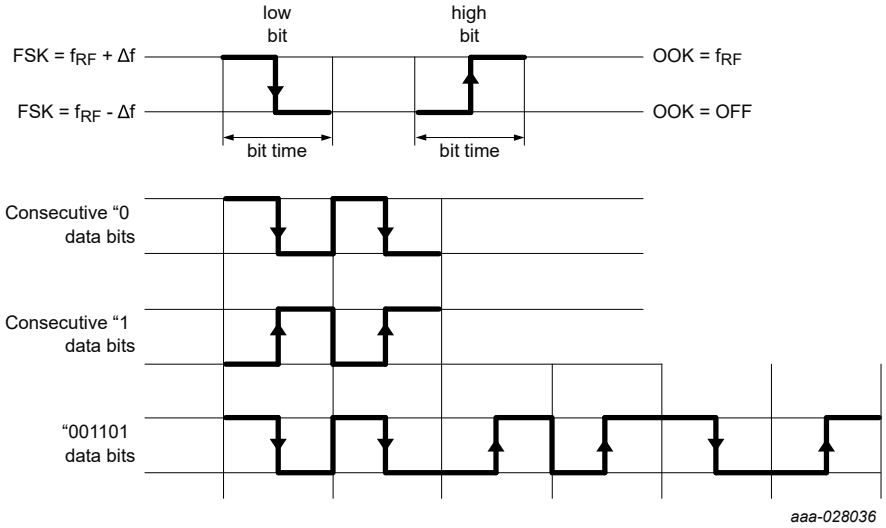


Figure 35. Manchester data bit encoding (POL = 1)

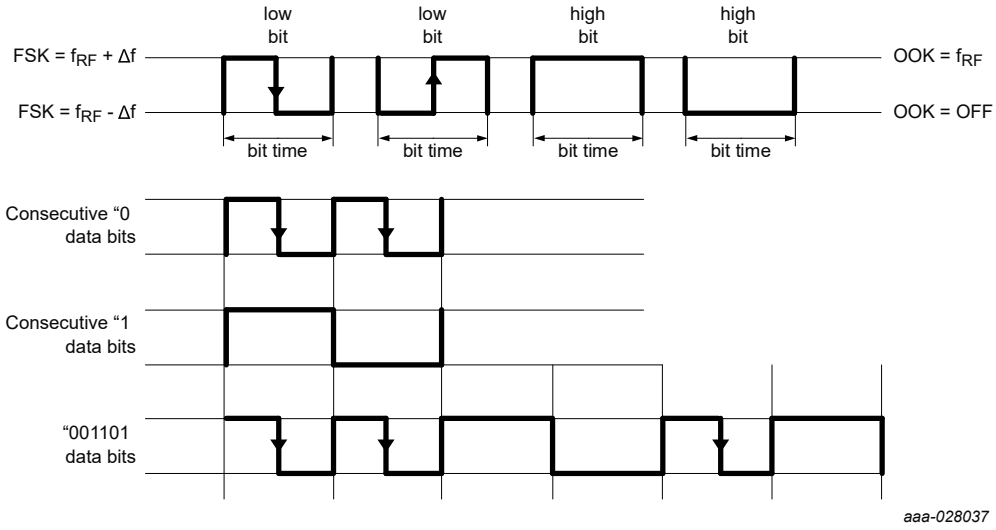


Figure 36. Bi-Phase data bit encoding (POL = 0)

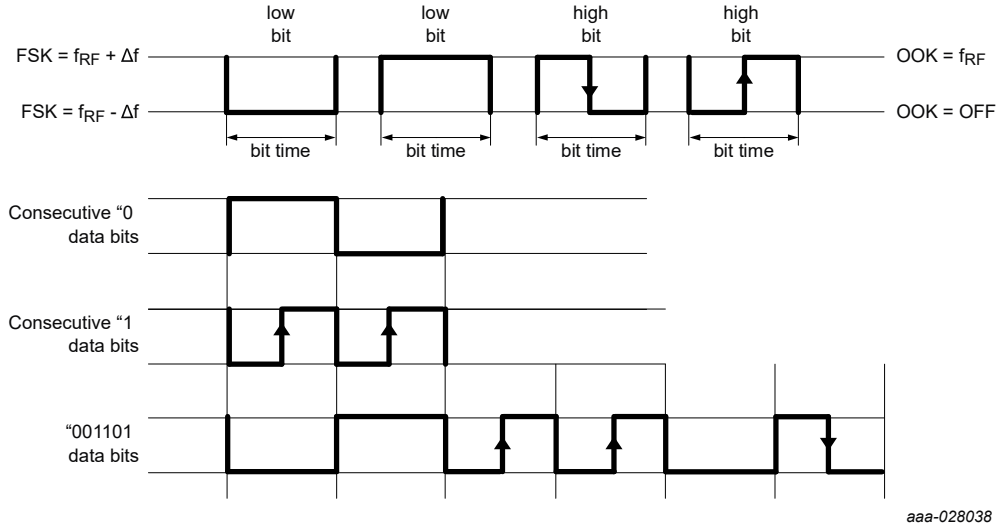


Figure 37. Bi-Phase data bit encoding (POL = 1)

10.16.6 RF output stage

The RF output stage consists of a PLL, control logic and an output RF amplifier. Data is sent to the RF output stage from either the RF data buffer or the DATA bit in the RFCR3 depending on the selected mode of operation as described in Section 10.16.1 "RF data modes".

The RF output stage is enabled by the state of the SEND control bit. The PLL in the RF output stage will signal back via the RCTS status bit when the PLL is locked and ready to transmit.

10.16.6.1 Modulation method

The modulation control bit, MOD, described in Table 119 and Table 123, sets the modulation of the RF signal will be either amplitude shift keying (OOK) or frequency shift keying (FSK) with several options for the frequency shift.

When operating in the FSK mode the internal, fractional-n PLL divider will be used to create the two carrier frequencies for data zero and data one. This method is more effective and robust than "pulling" the external crystal in order to shift the carrier frequency.

10.16.6.2 Carrier frequency

The carrier frequency is established mainly by the external crystal used, but a centering of the fractional-n PLL provides more precise control. If the CF control bit is clear the PLL will be configured for a carrier center frequency of the 315 MHz. If the CF control bit is set the PLL will be configured for a carrier center frequency of the 434 MHz.

10.16.6.3 RF power output

The maximum power output from the RF pin can be adjusted to one of 21 levels using the PWR[4:0] bits.

10.16.6.4 Transmission error

Any transmission will be aborted if one of the following occurs:

1. The RCTS signal does not become active within the t_{LOCK} time.
2. The PLL falls out of lock after once being set and the SEND bit is still active.
3. The XCO monitor output falls.

If either of these cases occurs the RF output will be turned off; the SEND control bit will be cleared; and the transmission error status flag, RFEF, will be set. The RFEF bit triggers an interrupt of the MCU if the RFIEN is set. The RFEF bit is cleared by writing a logical one to the RFIAK bit.

10.16.6.5 Supply voltage check during RF transmission

A separate low voltage detector can be enabled during the RF transmission and a status bit checked for low voltage drops due to a weak battery during the higher transmission currents. This RF LVD can be enabled by setting the RFLVDEN bit and the resulting status is reported on the RFVF bit. The RFVF bit can be cleared by writing a logical one to the RFIAK bit if the supply voltage has risen above the detect threshold. Further, if the voltage falls far enough for the VCO and PLL to fall out-of-lock, then the RF output will be turned off and the transmission will be terminated.

10.16.6.6 RF Reset (RFMRST)

The RF state machine, crystal oscillator, PLL and VCO can be reset to the initial off state by the RFMRST signal generated by one of the following methods:

1. Internal RFM power-on reset (RFPOR).
2. Writing a one to the RFMRST bit in the RFCR7.

Any of these reset methods will not alter any data stored in the data buffer.

10.16.7 RF interrupt

The RFM will interrupt the MCU when the SEND bit is cleared at the end of a data buffer transmission. This interrupt occurs at the end of a programmed set of frames. If number of frame count FNUM[3:0] is set to zero, then only one frame is sent and the interrupt occurs at the end of that first frame transmitted. If the number of the frame count is greater than zero, then the interrupt will be generated depending on the state of the IFID bit.

The interrupt will also create a flag bit, RFIF, which can be cleared by writing a logical one to the RFIAK bit. The interrupt can be enabled/disabled by the RFIEN bit.

10.16.8 Datagram transmission times

In order to comply with FCC requirements in the US market the periodically transmitted datagram must be less than 1 second in length and be separated by an off time that is at least 10 seconds or at least 30 times longer than the transmission time, whichever is longer. The user software must adhere to this ruling for products intended for the US market.

10.16.9 Pre-charge function

Pre-charge function can be enabled before a transmission to reduce the slew rate of current consumption during power up sequence, avoiding current surges which might otherwise result in LVD activation or reduce battery life. If the Pre-Charge function is enabled by setting AREGPC = 1, then when the analog regulator is activated, the pre-charge function starts after the digital regulator voltage is established. As long as AREGOK bit is 0, the pre-charge function is ON and the radio transmitter is OFF. When the voltage across the external capacitance reaches the threshold voltage 1.5 V, then AREGOK bit is set to 1, the pre-charge circuitry is automatically disabled and the power up sequence continues as usual.

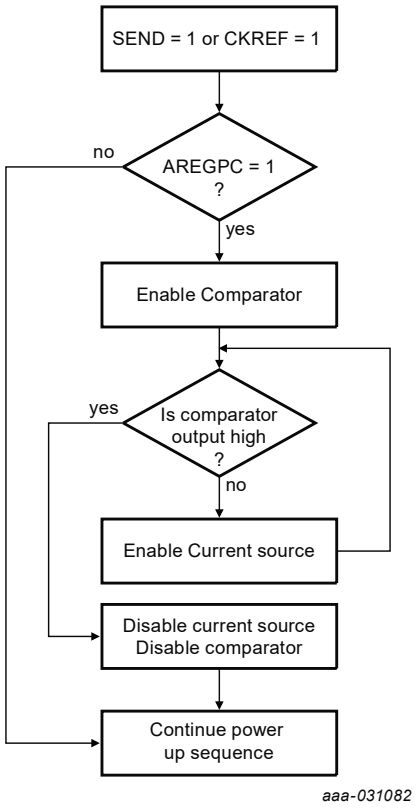


Figure 38. Pre-charge diagram

10.16.10 VCO calibration machine

The RFM incorporates a VCO calibration machine which works in conjunction with the VCO. The calibration machine selects the optimal VCO sub-band with respect to a predefined reference voltage applied to the VCO.

- Calibration supports max band VCO sub-bands. Max band corresponds to the band where the VCO frequency is maximum.
- A successive approximation algorithm is used to calculate the optimum sub-band.
- F_c, the Center Frequency (AFREQ+BFREQ)/2 is used as the reference frequency for the VCO calibration in FSK mode (MOD = 1).
- BFREQ is used as the reference frequency for the VCO calibration in OOK mode (MOD = 0).
- Calibration occurs every time the VCO is enabled.

- The calibration takes approximately 5 µs.

The state machine of the calibration is shown in Figure 39.

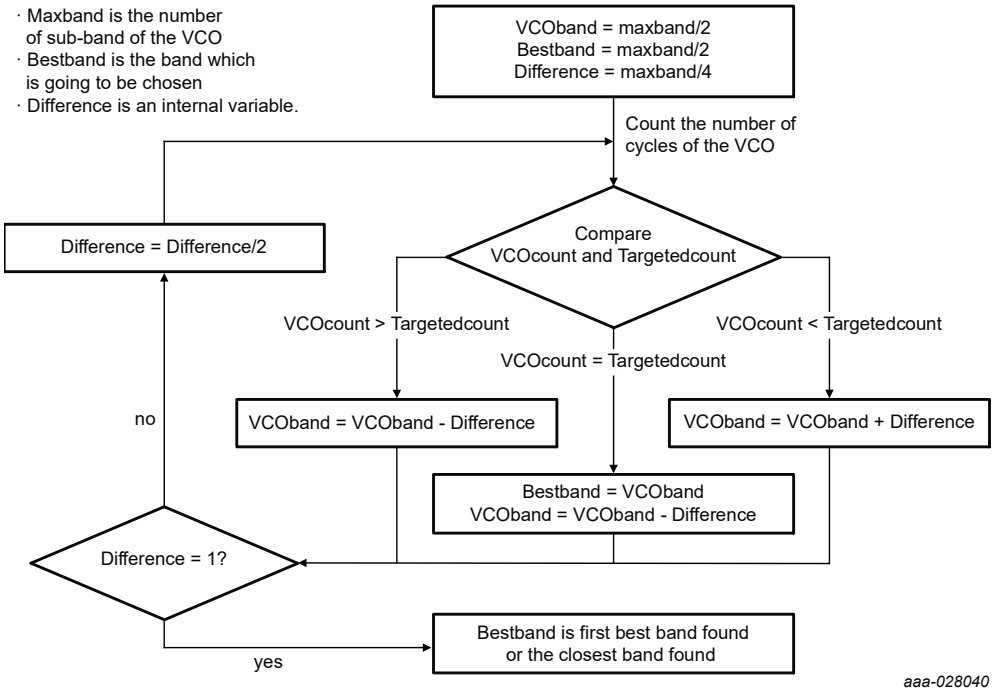


Figure 39. VCO calibration state machine

10.16.11 RFM register descriptions

10.16.11.1 RFM control 0 register (RFCR0)

Table 102. RFM control 0 register (RFCR0) (address $1830)

Bit	7	6	5	4	3	2	1	0
R	BPS7	BPS6	BPS5	BPS4	BPS3	BPS2	BPS1	BPS0
W								
Reset ($34)	0	0	1	1	0	1	0	0

Table 103. RFCR0 register field descriptions

Field	Description
7:0 BPS[7:0]	The BPS[7:0] bits select a divider for the incoming external crystal clock to generate the baud rate. The operating range of BPS[7:0] is $00 up to $FF. Reading BPS[7:0] provides the value written. This results in a baud rate from 1953 up to 500,000 bits per second, when the external crystal clock is 26 MHz. The conversion from the decimal value of the BPS[7:0] bits to the baud rate is given as described in the following equation. Power-on-reset forces BPS[7:0] to a value of $34 (decimal 52), and results in baud rate of 9434 bits per second, assuming external crystal clock is typical 26 MHz. $$\text{Baud rate} = \frac{f_{XTAL}}{52 \times (BPS[7:0] + 1)} = \frac{5 \times 10^5}{BPS[7:0] + 1}$$ Where: f_{XTAL} = External crystal frequency in Hz = 26 MHz $34 = Result of Reset

Table 104. Data rate option examples

Data Rate		BPS[7:0] Decimal Value		Data Rate		BPS[7:0] Decimal Value
Target	Nominal	f_{XTAL} = 26 MHz		Target	Nominal	f_{XTAL} = 26 MHz
2000 bit/s	2000.0	249		4800 bit/s	4807.7	103
2400 bit/s	2403.8	207		5000 bit/s	5000.0	99
4000 bit/s	4000.0	124		9600 bit/s	9615.4	51
4500 bit/s	4504.5	110		19200 bit/s	19230.8	25

10.16.11.2 RFM control 1 register (RFCR1)

Table 105. RFM control 1 register (RFCR1) (address $1831)

Bit	7	6	5	4	3	2	1	0
R	FRM7	FRM6	FRM5	FRM4	FRM3	FRM2	FRM1	FRM0
W								
Reset ($00)	0	0	0	0	0	0	0	0

Table 106. RFCR1 register field descriptions

Field	Description
7:0 FRM[7:0]	The 8 bits FRM[7:0] select the number of bits in each datagram. The number of bits is determined by the binary value of the FRM[7:0] bits plus one. This makes the range of bits from 2 to 256. A value of $00 for the FRM[7:0] control bits will result in no frames being sent. The FRM[7:0] control bits are cleared by RFMRST signal. $00 = Result of Reset

10.16.11.3 RFM control 2 register (RFCR2)

Table 107. RFM control 2 register (RFCR2) (address $1832)

Bit	7	6	5	4	3	2	1	0
R	SEND status	reserved	reserved	PWR4	PWR3	PWR2	PWR1	PWR0
W	SEND							
Reset ($00)	0	0	0	0	0	0	0	0

Table 108. RFCR2 register field descriptions

Field	Description
7 SEND	SEND – Start Transmit Control The SEND control bit starts the transmission of data held in the RFM data buffer according to the bit length specified by the FRM[7:0] bits. The SEND control bit is automatically cleared when the data buffer transmission has ended or by the RFMRST signal. A transmission can be prematurely interrupted by writing a logical zero to the SEND bit. 0 = Data transmission ended or transmission not in progress; Result of Reset 1 = Start data transmission or transmission in progress.
4:0 PWR[4:0]	PWR[4:0] – RF Transmit Power Selection The PWR[4:0] control bits select the optimum power output of the RF power amplifier. These power output levels assume optimal matching network to the RF pin. The PWR[4:0] control bits are cleared by an RFM reset. This setting targets -10 dBm typical power output. The PWR control bits scale the typical output power level from -1.5 dBm to 8 dBm in steps of 0.5 dB and fixes the low power level mode to -10 dBm. 00000 set output power level to –10 dBm (Result of reset) 00001 set output power level to –1.5 dBm 00010 set output power level to –1.0 dBm 00011 set output power level to –0.5 dBm 00100 set output power level to 0.0 dBm 00101 set output power level to 0.5 dBm 00110 set output power level to 1.0 dBm 00111 set output power level to 1.5 dBm 01000 set output power level to 2.0 dBm 01001 set output power level to 2.5 dBm 01010 set output power level to 3.0 dBm 01011 set output power level to 3.5 dBm 01100 set output power level to 4.0 dBm 01101 set output power level to 4.5 dBm 01110 set output power level to 5.0 dBm 01111 set output power level to 5.5 dBm 10000 set output power level to 6.0 dBm 10001 set output power level to 6.5 dBm 10010 set output power level to 7.0 dBm 10011 set output power level to 7.5 dBm 10100 set output power level to 8.0 dBm Codes greater than 10100 are reserved for test purposes and should not be used.

10.16.11.4 RFM control 3 register (RFCR3)

Table 109. RFM control 3 register (RFCR3) (address $1833)

Bit	7	6	5	4	3	2	1	0
R	DATA	IFPD	ISPC	IFID	FNUM3	FNUM2	FNUM1	FNUM0
W								
Reset ($00)	0	0	0	0	0	0	0	0

Table 110. RFCR3 register field descriptions

Field	Description
7 DATA	DATA – Data State The DATA bit determines the output state of the RF power amplifier when the RFM is in the MCU direct control mode (CODE[1:0] = 11) 0 = RF output state low; Result of Reset 1 = RF output state high.
6 IFPD	IFPD – Interframe Power Down Control The IFPD control bit selects whether the XCO and associated analog blocks are powered down during interframe timing caused by the RFM. The IFPD control bit is cleared by the RFMRST signal. The restart of these functions will start 1 ms before the end of the timing interval if another frame is to be transmitted. 0 = The XCO remains powered up as long as the SEND bit is set; Result of Reset 1 = The XCO is powered down during RFM controlled interframe timing events.
5 ISPC	ISPC – Interframe Random Space Control When the ISPC bit is set the initial time delay before the first frame will be enabled. This bit is cleared by an RFM reset. 0 = No initial time interval; Result of Reset 1 = Initial time interval enabled.
4 IFID	IFID – Interframe Interrupt Delay Control The IFID control bit selects how the RFIF bit will be managed. The IFID control bit is cleared by the RFMRST signal. 0 = The RFIF bit is set and the MCU interrupted if the RFIEN bit is set, after the last frame transmitted; Result of Reset 1 = The RFIF bit is set and the MCU interrupted if the RFIEN bit is set, only after the last frame plus an additional interframe message is transmitted.
3:0 FNUM[3:0]	FNUM[3:0] – Number of Frames The 4 bits FNUM[3:0] bits set the number of frames transmitted in each RF datagram. The frames will be randomly spaced apart as described in bits 6, 5 and 4. These bits are cleared by an RFM reset. The number of frames transmitted is the binary number plus one. 0 0 0 0 = Result of Reset

10.16.11.5 RFM control 4 register (RFCR4)

Table 111. RFM control 4 register (RFCR4) (address $1834)

Bit	7	6	5	4	3	2	1	0
R	RFBT7	RFBT6	RFBT5	RFBT4	RFBT3	RFBT2	RFBT1	RFBT0
W								
Reset ($80)	1	0	0	0	0	0	0	0

Table 112. RFCR4 register field descriptions

Field	Description
7:0 RFBT[7:0]	RFBT[7:0] – The 8 bits RFBT[7:0] select the interframe timing between multiple frames of transmission. The base time value is equal to a nominal one millisecond for each count of the RFBT[7:0] bits. The RFBT[7:0] control bits are cleared by the RFMRST signal and must be set to either 0 or between 5 and 255. $80 = Result of Reset

10.16.11.6 RFM control 5 register (RFCR5)

Table 113. RFM control 5 register (RFCR5) (address $1835)

Bit	7	6	5	4	3	2	1	0
R W	BOOST	LFSR6	LFSR5	LFSR4	LFSR3	LFSR2	LFSR1	LFSR0
Reset ($00)	0	0	0	0	0	0	0	0

Table 114. RFCR5 register field descriptions

Field	Description
7 BOOST	BOOST – VCO Power Consumption Boost Control This bit controls the VCO power consumption in order to decrease the phase noise required by the Japanese regulation. The BOOST control bit is cleared by the RFMRST signal. 0 = The VCO runs at its lower power consumption level (higher phase noise); Result of Reset 1 = The VCO runs at its higher power consumption level (lower phase noise).
6:0 LFSR[6:0]	LFSR[6:0] – Pseudo-Random timer The 7 bits LFSR[6:0] select the current seed value of the LFSR when enabling pseudo-random timing intervals when any of the LFSR[6:0] bits are set. The value written to this register is loaded into the actual LFSR when the SEND bit is set. The time value is equal to a nominal one millisecond for each count of the resulting LFSR[6:0] bits. A value of $00 placed in the LFSR causes the LFSR to stay at the $00 state on each clocking of the LFSR. To cause the LFSR to cycle through its pseudo-random number sequence requires that any value other than $00 be written to the LFSR[6:0] bits. 0 0 0 0 0 0 0 = Result of Reset

10.16.11.7 RFM control 6 register (RFCR6)

Table 115. RFM control 6 register (RFCR6) (address $1836)

Bit	7	6	5	4	3	2	1	0
R W	VCO_ GAIN1	VCO_ GAIN0	RFFT5	RFFT4	RFFT3	RFFT2	RFFT1	RFFT0
Reset ($00)	1	0	0	0	0	0	0	0

Table 116. RFCR6 register field descriptions

Field	Description
7:6 VCO_ GAIN[1:0]	VCO_GAIN[1:0] – VCO Gain Selection The two bits VCO_GAIN[1:0] bit is set and the VCO_GAIN[0] bit is cleared by the RFMRST signal. VCO_GAIN does not normally need to be adjusted by the end user. 1 0 = Result of Reset

Table 116. RFCR6 register field descriptions*...continued*

Field	Description
5:0 RFFT[5:0]	RFFT[5:0] – Frame Number Timer The 6 bits RFFT[5:0] select the interframe timing between multiple frames of transmission. The time value is equal to a nominal one millisecond for each count of the RFFT[5:0] bits multiplied by the frame number of the last transmitted frame. The RFFT[5:0] control bits are cleared by the RFMRST signal. 0 0 0 0 0 0 = Result of Reset

10.16.11.8 RFM control 7 register (RFCR7)

Table 117. RFM control 7 register (RFCR7) (address $1837)

Bit	7	6	5	4	3	2	1	0
R	RFIF	RFEF	RFVF	0	RFIEN	RFLVDEN	RCTS	0
W	—	—	—	RFIAK			—	RFMRST
Reset ($00)	0	0	0	0	0	0	0	0

Table 118. RFCR7 register field descriptions

Field	Description
7 RFIF	RFIF – RF Interrupt Flag The read-only RFIF status bit indicates if the RF transmission has ended properly when using the data buffer mode and the SEND bit has been cleared. Writes to this bit will be ignored. The RFIF status bit is cleared by writing a logical one to the RFIAK bit or the RFMRST bit. RFMRST signal clears this bit. 0 = RF transmission in progress or not in the data buffer mode; Result of Reset 1 = RF transmission completed in the data buffer mode.
6 RFEF	RFEF – RF Transmission Error Flag The read-only RFEF status bit indicates if there was an error in the current or prior RF transmission. Writes to this bit will be ignored. The RFEF status bit is cleared by writing a logical one to the RFIAK bit or the RFMRST bit. RFMRST signal clears this bit. 0 = No RF transmission error occurred; Result of Reset 1 = RF transmission error occurred.
5 RFVF	RFVF – RF LVD Trigger Flag When the RF LVD is enabled and the supply voltage falls below the threshold, the read-only RFVF flag will be set if the RFLVDEN bit is set. Writes to this bit will be ignored. The RFVF status bit is cleared by writing a logical one to the RFIAK bit or the RFMRST bit. RFMRST signal clears this bit 0 = Voltage is and has been above RF LVD rising threshold or the RF LVD is disabled; Result of Reset 1 = Voltage has dropped below the RF LVD falling threshold since last reset of this bit.
4 RFIAK	RFIAK – RF Interrupt Acknowledge Control Writing a one to the RFIAK bit clears the RFIF, RFEF and RFVF flag bits. Writing a zero to the RFIAK bit has no effect on the RFIF, RFEF and RFVF flag bits. The RFMRST signal has no effect on this bit. 0 = No effect; Result of Reset 1 = Write 1 to clear the RFIF, RFEF, and RFVF bits.
3 RFIEN	RFIEN – RF Interrupt Enable Control The RFIEN bit enables the RFIF, the RFEF and the RFVF bits to generate an interrupt to the MCU. The RFMRST signal clears this bit. 0 = RF interrupts disabled; Result of Reset 1 = RF interrupts enabled.

Table 118. RFCR7 register field descriptions*...continued*

Field	Description
2 RFLVDEN	RFLVDEN – RF Low Voltage Detect Enable Control When the RFLVDEN bit is set, the RF LVD circuit will be enabled, and the RF LVD events are routed to the RF LVD Trigger Flag. This bit is cleared by the RFMRST signal. 0 = RF LVD disabled; Result of Reset 1 = RF LVD enabled.
1 RCTS	RCTS – RF Clear To Send Status Flag When the RCTS bit is set the RF XCO, VCO and PLL have started and locked and the RFM is ready to send data. This bit is cleared by the RFMRST signal. 0 = RFM not ready to send; Result of Reset 1 = RFM ready to send.
0 RFMRST	RFMRST – RFM Reset Control Writing a one to the RFMRST bit will completely reset the RFM and its registers. This bit is not affected by a reset of the MCU. This bit will always read as a zero. 0 = No effect; Result of Reset 1 = Reset RFM.

10.16.11.9 RFM phase lock loop control registers 0 through 3 (PLLCR0 / PLLCR1 / PLLCR2 / PLLCR3)

Table 119. RFM phase lock loop control register 0 (PLLCR0) (address $1838)

Bit	15	14	13	12	11	10	9	8
R W	AFREQ12	AFREQ11	AFREQ10	AFREQ9	AFREQ8	AFREQ7	AFREQ6	AFREQ5
Reset	0	0	0	0	0	0	0	0

Table 120. RFM phase lock loop control register 1 (PLCCR1) (address $1839)

Bit	7	6	5	4	3	2	1	0
R W	AFREQ4	AFREQ3	AFREQ2	AFREQ1	AFREQ0	POL	CODE1	CODE0
Reset	0	0	0	0	0	0	0	0

Table 121. RFM phase lock loop control register 2 (PLCCR2) (address $183A)

Bit	15	14	13	12	11	10	9	8
R W	BFREQ12	BFREQ11	BFREQ10	BFREQ9	BFREQ8	BFREQ7	BFREQ6	BFREQ5
Reset	0	0	0	0	0	0	0	0

Table 122. RFM phase lock loop control register 3 (PLCCR3) (address $183B)

Bit	7	6	5	4	3	2	1	0
R W	BFREQ4	BFREQ3	BFREQ2	BFREQ1	BFREQ0	CF	MOD	CKREF
Reset	0	0	0	0	0	0	0	0

Table 123. PLLCR0 / PLLCR1 / PLLCR2 / PLLCR3 register field descriptions

Field	Description
15:3 AFREQ[12:0]	The 13 bits AFREQ[12:0] control the FSK carriers for transmitting 0's. The AFREQ[12:0] control bits are cleared by the RFMRST signal. $$\text{FSK 0 carrier} = f_{XTAL} \times \left\{ 12 + (4 \times CF) + \frac{AFREQ[12:0]}{8192} \right\}$$ Where: f_{XTAL} = External crystal frequency in Hz = typical 26 MHz CF = state of the CF carrier select control bit. AFREQ[12:0] = decimal value, 1 LSB of AFREQ = 3.17 kHz $0000 = Result of Reset
2 POL	POL – Data Polarity The POL control bit selects the polarity of the data encoding selected by the CODE[1:0] bits. The POL control bit is cleared by the RFMRST signal. 0 = NRZ and MCU direct DATA bit data non-inverted and Manchester encoding polarity; Result of Reset 1 = all types of encoding polarity are inverted.
1:0 CODE[1:0]	CODE[1:0] – Data Encoding and Source The two bits CODE[1:0] control bits select the type of data encoding and source of data for the RF output. The CODE[1:0] control bits are cleared by the RFMRST signal. 0 0 = Manchester encoded data from the RFM data buffer; Result of Reset 0 1 = Bi-Phase encoded data from the RFM data buffer. 1 0 = NRZ encoded data from the RFM data buffer. 1 1 =MCU direct mode with RF output driven by the state of the DATA bit.
15:3 BFREQ[12:0]	The 13 bits BFREQ[12:0] control the OOK carrier for transmitting 1's, where lack of carrier defines OOK 0's. $$\text{FSK 1 carrier} = f_{XTAL} \times \left\{ 12 + (4 \times CF) + \frac{BFREQ[12:0]}{8192} \right\}$$ Where: f_{XTAL} = External crystal frequency in Hz = typical 26 MHz CF = state of the CF carrier select control bit. BFREQ[12:0] = decimal value, 1 LSB of BFREQ = 3.17 kHz $0000 = Result of Reset
2 CF	CF – Carrier Frequency Control The CF control bit selects the optimal VCO setup and correct divider for the 500 kHz reference clock to the MCU on DX based on the external crystals required for the desired carrier frequency. The CF control bit is cleared by the RFMRST signal. 0 = Configured for 315 MHz, 12.1154 PLL divider using a 26.000 MHz external crystal; Result of Reset 1 = Configured for 434 MHz, 16.6923 PLL divider using a 26.000 MHz external crystal.

Table 123. PLLCR0 / PLLCR1 / PLLCR2 / PLLCR3 register field descriptions*...continued*

Field	Description
1 MOD	MOD – RF Modulation Control The MOD control bit selects the method of modulating the RF. The MOD control bit is cleared by the RFMRST signal. 0 = Configured for OOK; Result of Reset 1 = Configured for FSK.
0 CKREF	CKREF – Generated Clock Reference Control Generates the DX signal to the TPM1 module for determining the other clock frequencies: 0 = DX signal not generated; Result of Reset 1 = DX 500 kHz signal connected to the TPM1 module.

10.16.11.10 RFM transmit data 0 through 31 registers (RFTX0 : RFTX31)

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD7	RFTXD6	RFTXD5	RFTXD4	RFTXD3	RFTXD2	RFTXD1	RFTXD0
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD15	RFTXD14	RFTXD13	RFTXD12	RFTXD11	RFTXD10	RFTXD9	RFTXD8
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD23	RFTXD22	RFTXD21	RFTXD20	RFTXD19	RFTXD18	RFTXD17	RFTXD16
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD31	RFTXD30	RFTXD29	RFTXD28	RFTXD27	RFTXD26	RFTXD25	RFTXD24
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD39	RFTXD38	RFTXD37	RFTXD36	RFTXD35	RFTXD34	RFTXD33	RFTXD32
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD47	RFTXD46	RFTXD45	RFTXD44	RFTXD43	RFTXD42	RFTXD41	RFTXD40
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD55	RFTXD54	RFTXD53	RFTXD52	RFTXD51	RFTXD50	RFTXD49	RFTXD48
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD63	RFTXD62	RFTXD61	RFTXD60	RFTXD59	RFTXD58	RFTXD57	RFTXD56
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD71	RFTXD70	RFTXD69	RFTXD68	RFTXD67	RFTXD66	RFTXD65	RFTXD64
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD79	RFTXD78	RFTXD77	RFTXD76	RFTXD75	RFTXD74	RFTXD73	RFTXD72
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD87	RFTXD86	RFTXD85	RFTXD84	RFTXD83	RFTXD82	RFTXD81	RFTXD80
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD95	RFTXD94	RFTXD93	RFTXD92	RFTXD91	RFTXD90	RFTXD89	RFTXD88
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD103	RFTXD102	RFTXD101	RFTXD100	RFTXD99	RFTXD98	RFTXD97	RFTXD96
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD111	RFTXD110	RFTXD109	RFTXD108	RFTXD107	RFTXD106	RFTXD105	RFTXD104
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD119	RFTXD118	RFTXD117	RFTXD116	RFTXD115	RFTXD114	RFTXD113	RFTXD112
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD127	RFTXD126	RFTXD125	RFTXD124	RFTXD123	RFTXD122	RFTXD121	RFTXD120
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD135	RFTXD134	RFTXD133	RFTXD132	RFTXD131	RFTXD130	RFTXD129	RFTXD128
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD143	RFTXD142	RFTXD141	RFTXD140	RFTXD139	RFTXD138	RFTXD139	RFTXD136
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD151	RFTXD150	RFTXD149	RFTXD148	RFTXD147	RFTXD146	RFTXD145	RFTXD14
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD159	RFTXD158	RFTXD157	RFTXD156	RFTXD155	RFTXD154	RFTXD153	RFTXD152
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD167	RFTXD166	RFTXD165	RFTXD164	RFTXD163	RFTXD162	RFTXD161	RFTXD160
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD175	RFTXD174	RFTXD173	RFTXD172	RFTXD171	RFTXD170	RFTXD169	RFTXD168
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD183	RFTXD182	RFTXD181	RFTXD180	RFTXD179	RFTXD178	RFTXD177	RFTXD176
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD191	RFTXD190	RFTXD189	RFTXD188	RFTXD187	RFTXD186	RFTXD185	RFTXD184
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD199	RFTXD198	RFTXD197	RFTXD196	RFTXD195	RFTXD194	RFTXD193	RFTXD192
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R	RFTXD207	RFTXD206	RFTXD205	RFTXD204	RFTXD203	RFTXD202	RFTXD201	RFTXD200
W								
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD215	RFTXD214	RFTXD213	RFTXD212	RFTXD211	RFTXD210	RFTXD209	RFTXD208
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD223	RFTXD222	RFTXD221	RFTXD220	RFTXD219	RFTXD218	RFTXD217	RFTXD216
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD231	RFTXD230	RFTXD229	RFTXD228	RFTXD227	RFTXD226	RFTXD225	RFTXD224
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD239	RFTXD238	RFTXD237	RFTXD236	RFTXD235	RFTXD234	RFTXD233	RFTXD232
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD247	RFTXD246	RFTXD245	RFTXD244	RFTXD243	RFTXD242	RFTXD241	RFTXD240
Reset	x	x	x	x	x	x	x	x

Table 124. RFM transmit data 0 through 31 registers (RFTX0 : RFTX31) (addresses $183C : $185B)

Bit	7	6	5	4	3	2	1	0
R W	RFTXD255	RFTXD254	RFTXD253	RFTXD252	RFTXD251	RFTXD250	RFTXD249	RFTXD248
Reset	x	x	x	x	x	x	x	x

Table 125. RFTX0 : RFTX31 register field descriptions

Field	Description
0:255 RFTXD[0:255]	The RFTX0 through RFTX31 registers contain 256 read/write bits for the RFM to use when outputting data as described in Table 124. These bits are unaffected by any reset. The data buffer is unloaded to the RF output starting with the least significant bit (RFTXD0) in the least significant byte ($183C) up through the most significant bit (RFTXD255) in the most significant byte ($185B). This is often referred to as “little-endian” data ordering. Result of reset may be random values.

10.16.11.11 RFM EOM, PLL and PA control register (EPR)

Table 126. RFM EOM, PLL and PA control register (EPR) (address $1860)

Bit	7	6	5	4	3	2	1	0
R	EOM	PLL_LPF2	PLL_LPF1	PLL_LPF0	reserved	reserved	PA_ SLOPE1	PA_ SLOPE0
W								
Reset ($33)	0	0	1	1	0	0	1	1

Table 127. EPR register field descriptions

Field	Description
7 EOM	EOM – RF Transmit End Of Message Flag The EOM control bit selects whether there will be two data bit times of data 1 carrier state at the end of each datagram. The EOM control bit is cleared by an RFMRST. 0 = EOM bit times not added; Result of Reset 1 = EOM bit times added.
6:4 PLL_ LPF[2:0]	PLL_LPF[2:0] – Phase Lock Loop Low Pass Filter Selection The 3 PLL_LPF[3:0] read/write bits select the PLL low pass filter. A reset sets these bits to 0 1 1.
1:0 PA_ SLOPE[1:0]	PA_SLOPE[1:0] – Power Amp Slope Control The two bits PA_SLOPE[1:0] selects the slope of the RFM PA output. These bits are set to 1 1 by the RFMRST signal. 0 0 = Set OOK data slope to 0.3 µs 0 1 = Set OOK data slope to 3.0 µs 1 0 = Set OOK data slope to 6.0 µs 1 1 = Set OOK data slope to 9.0 µs; Result of Reset

10.16.11.12 RFM Pre-charge control register (RFPRECHARGE)

Table 128. RFM Pre-charge control register (RFPRECHARGE) (address $1861)

Bit	7	6	5	4	3	2	1	0
R	TIMEOUT1	TIMEOUT0	0	0	ENARE GCOMP	AREGPC	AREGOK	0
W			—	—			—	—
Reset ($40)	0	1	0	0	0	0	0	0

Table 129. RFPRECHARGE register field descriptions

Field	Description
7:6 TIMEOUT[1:0]	TIMEOUT[1:0] - RF timeout selection. The timeout selection provides a period to allow the PLL to lock, including the Pre-charge if enabled. If the timeout is reached and the PLL has not locked, the RF state machine will halt and set the error status flag RFEF to 1. Times listed below are approximate based upon the crystal: NDK NX3225SA, 26.000 MHz. When AREGPC = 0: 0 0 = disabled 0 1 = 800 µs; Result of Reset 1 0 = 1.6 ms 1 1 = 2.4 ms When AREGPC = 1: 0 0 = disabled 0 1 = 1.304 ms; Result of Reset 1 0 = 2.104 ms 1 1 = 2.904 ms
3 ENAREGCOMP	ENAREGCOMP – Pre-charge comparator. To conserve power, the pre-charge comparator is transient-enabled, meaning that it turns on only when required and is off otherwise. The ENAREGCOMP bit can be set to force the comparator on continuously. During test, this functionality is used to test the comparator. In the application, the MCU can use this function to check if the VREG pin is already pre-charged, i.e. the MCU can enable the comparator, read the value of the AREGOK bit, then disable the comparator (to avoid wasting power). 0 = disabled; Result of Reset 1 = enabled
2 AREGPC	AREGPC – Pre-charge external analog regulator load capacitor function. Enables automatic current-limited pre-charging of the RF analog regulator's external capacitor before transmitting. Use this function to avoid large inrush currents that can occur when the RF analog regulator is enabled. 0 = disabled; Result of Reset 1 = enabled
1 AREGOK	AREGOK – Read-only Pre-charge completion status: Read-only bit that returns the output of the pre-charge comparator. It is only meaningful when the comparator is enabled, i.e. when ENAREGCOMP=1. 0 = pre-charge not complete; Result of Reset 1 = pre-charge completed.

10.17 Analog-to-Digital converter (ADC) module

The following is a brief description of the features the ADC module provides:

- Execute conversions with 8-, 10-, or 12-bit resolutions
- Internal multiplexer capable of converting 14 signals including two external pins, and internal VREFL or VREFH
- Clock select control options (ADICLK)
  - bus clock
  - bus clock divided by 2
  - external source (ALTCLK) – either on-chip or off-chip, as defined by chip-level hookup
  - internal generated source (ADACK) asynchronous to the bus clock

- Conversion executed asynchronous to bus clock
  - Conversion can be performed while operating at any bus clock frequency (>0 Hz to 20 MHz)
- Clock rate control options (ADIV) – selected clock, divided by 1, 2, 4, or 8.
- Triggerable through software or external hardware (ADTRG)
  - for external hardware, trigger is either on-chip or off-chip, as defined by chip-level hookup
- Auto-compare function (ACFE)
  - digital comparator configurable for either "greater than or equal" or "less than" (ACFGT)
  - no conversion complete (COCO) flag setting and no data transfer on false compare
- Runs in stop mode (when ADICLK selects internal asynchronous clock source)
  - capable of asynchronous stop wake-up
  - continuous compare does not wake system until a true compare conversion
- Analog temperature sensor
  - used via conversion on one of the selectable internal channels
- Configurable conversion reference source (REFSEL) – currently an unadvertised feature
  - Normal V_{REFH}/V_{REFL} pad source (the default source – REFSEL[1:0] = 0 0)
  - V_{DDA}/V_{SSA} pad source
  - Alternate source (i.e., on-chip band gap reference circuit)

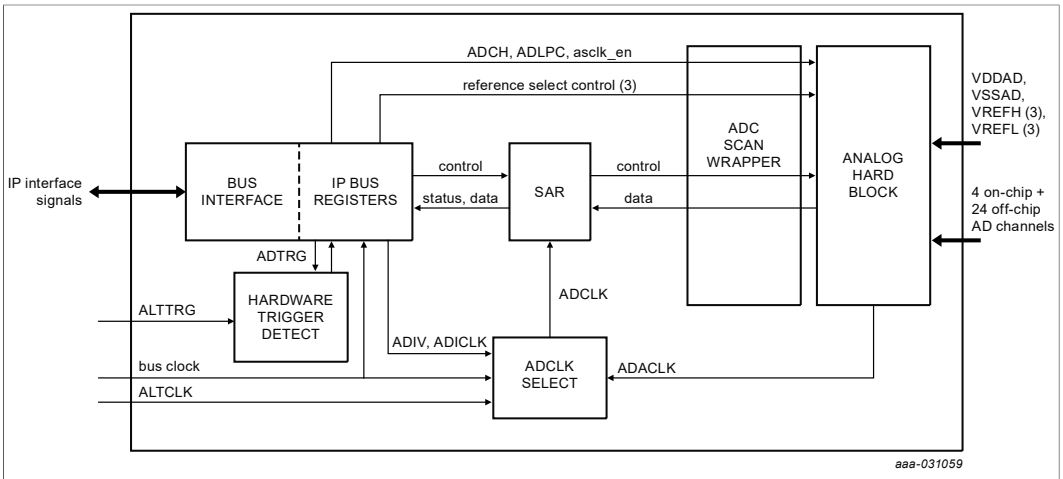


Figure 40. Analog to digital converter block diagram

10.17.1 ADC register descriptions

10.17.1.1 ADC status and control 1 register (ADSC1)

Table 130. ADC status and control 1 register (ADSC1) (address $0030)

Bit	7	6	5	4	3	2	1	0
R	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
W	—							
Reset ($1F)	0	0	0	1	1	1	1	1

Table 131. ADSC1 register field descriptions

Field	Description
7 COCO	COCO – Conversion Complete The COCO flag bit is implemented as read-only and indicates a conversion is complete and conversion data is available in the ADRH and ADRL registers. 0 = Conversion is not complete; Result of Reset 1 = Conversion is complete
6 AIEN	AIEN – ADC Interrupt Enable The AIEN bit allows system-level interrupt requests to be generated when the COCO bit is set. 0 = ADC interrupt is disabled; Result of Reset 1 = ADC interrupt is enabled
5 ADCO	ADCO – Single or Continuous Conversions The ADCO bit selects between single conversions and continuous conversions. 0 = Single conversion selected; Result of Reset 1 = Continuous conversions selected
4:0 ADCH[4:0]	ADCH[4:0] – ADC channel selection The ADCH[4:0] field selects the analog channel to be converted. All other possible values of ADCH[4:0] are valid channel values and result in conversion on 1 of the 28 possible channels, the selected reference high, or the selected reference low. 1 1 1 1 1 = Result of Reset, and disables the ADC.

Table 132. ADCH valid channel values

ADCH[4:0]	Channel	Signal name	Description
00000	AD0	smi_vout_to_adc	SMI output
00001	AD1	adc_temp_sense_out	Temperature sensor
00010	AD2	pmc_1p2_vref	1.2 V VREF internal band gap
00011	AD3	ipp_inouta_ptb<0>	PTB0 external pin
00100	AD4	ipp_inouta_ptb<1>	PTB1 external pin
00101	AD5	ipp_vreg	1.8 V VREG RF analog regulator output
00110	AD6	atb0	Analog test bus 0
00111	AD7	atb1	Analog test bus 1
01000	AD8	ipp_vssa	grounded
01001	AD9	ipp_vssa	grounded
01010	AD10	ipp_vssa	grounded
01011	AD11	ipp_vssa	grounded
01100	AD12	ipp_vssa	grounded
01101	AD13	ipp_vssa	grounded
01110	AD14	ipp_vssa	grounded
01111	AD15	ipp_vssa	grounded
10000	AD16	ipp_vssa	grounded
10001	AD17	ipp_vssa	grounded
10010	AD18	ipp_vssa	grounded

Table 132. ADCH valid channel values*...continued*

ADCH[4:0]	Channel	Signal name	Description
10011	AD19	ipp_vssa	grounded
10100	AD20	ipp_vssa	grounded
10101	AD21	ipp_vssa	grounded
10110	AD22	ipp_vssa	grounded
10111	AD23	ipp_vssa	grounded
11000	AD24	pmc_driver_vdd	2.5 V unswitched digital power supply
11001	AD25	pmc_driver_vdd_sw	2.5 V switched digital power supply
11010	AD26	pmc_driver_vdda_sw	2.5 V switched analog power supply
11011	AD27	ipp_vssa	grounded
11100	AD28	VREFH	ADC high reference voltage
11101	AD29	VREFH	ADC high reference voltage
11110	AD30	VREFL	ADC low reference voltage
11111	--	--	Select to disable ADC

10.17.1.2 ADC status and control 2 register (ADSC2)

Table 133. ADC status and control 2 register (ADSC2) (address $0031)

Bit	7	6	5	4	3	2	1	0
R	ADACT	ADTRG	ACFE	ACFGT	0	0	REFSEL1	REFSEL0
W	—				—	—		
Reset ($00)	0	0	0	0	0	0	0	0

Table 134. ADSC2 register field descriptions

Field	Description
7 ADACT	ADACT – SAR active The ADACT bit is a read-only bit that indicates that the SAR logic is active and a conversion is in progress. 0 = SAR is not active; Result of Reset 1 = SAR is active
6 ADTRG	ADTRG – Software / Hardware triggered conversions The ADTRG bit selects between software triggered conversions and hardware trigger conversions. 0 = Software triggered conversions are selected; Result of Reset 1 = Hardware triggered conversions are selected
5 ACFE	ACFE – Normal or Compare Conversions The ACFE bit selects between normal conversion operation and compare operation. 0 = normal conversions are selected; Result of Reset 1 = Compare conversions are selected
4 ACFGT	ACFGT – Greater Than / Less Than Matching The ACFGT bit selects between "greater than" or "less than" matching criteria when in compare operation. 0 = less than mode is selected; Result of Reset 1 = greater than mode is selected

Table 134. ADSC2 register field descriptions*...continued*

Field	Description
1:0 REFSEL[1:0]	REFSEL[1:0] – Reference selections The REFSEL[1:0] bits select the source for the voltage reference used for analog conversions. The analog block supports three options: • the standard VREFH/VREFL pins (these may be shared with standard I/O pins in the future) • the VDDA/VSSA pins • an alternate reference source (i.e., an on-chip band gap reference) 0 0 = V_{REFH} and V_{REFL}; Result of Reset 0 1 = V_{DDA} and V_{SSA} 1 0 = band gap_H and band gap_L 1 1 = V_{REFH} and V_{REFL}

10.17.1.3 ADC result high and low registers (ADRH/L)

Table 135. ADC result high register (ADRH) (address $0032)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	0	ADR11	ADR10	ADR9	ADR8
W	—	—	—	—	—	—	—	—
Reset ($00)	0	0	0	0	0	0	0	0

Table 136. ADC result low register (ADRL) (address $0033)

Bit	7	6	5	4	3	2	1	0
R	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
W	—	—	—	—	—	—	—	—
Reset ($00)	0	0	0	0	0	0	0	0

Table 137. ADRH/L register field descriptions

Field	Description
11:0 ADRH/L	The 12-bit ADR[11:0] are 2 read-only CPU accessible registers to present data result values following ADC conversions: • ADRH – Data result high • ADRL – data result low A 12-bit data result register holds the final conversion result. The data captured in the result register is driven from the SAR Block. $0000 = Result of Reset

10.17.1.4 ADC compare value high and low registers (ADCVH/L)

Table 138. ADCV compare value high register (ADCVH) (address $0034)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	0	ADCV11	ADCV10	ADCV9	ADCV8
W	—	—	—	—	—	—	—	—
Reset ($00)	0	0	0	0	0	0	0	0

Table 139. ADCV compare value low register (ADCVL) (address $0035)

Bit	7	6	5	4	3	2	1	0
R	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
W	—	—	—	—	—	—	—	—
Reset ($00)	0	0	0	0	0	0	0	0

Table 140. ADCVH/L register field descriptions

Field	Description
11:0 ADCVH/L	The 12-bit compare value ADCV[11:0] is accessible through the ADCVH and ADCVL registers: • ADCVH – Compare value high nibble • ADCVL – Compare value low pair of nibbles The ADCVH and ADCVL registers are standard read/write accessible. CPU writes to the 4 low-order bits of ADCVH update reg_adcv[11:8] and writes to ADCVL update the 8 low-order bits (reg_adcv[7:0]). The ADCV field is used as the referenced compare value on Compare type conversions (when ACFE = 1). $0000 = Result of Reset

10.17.1.5 ADC configuration register

Table 141. ADC configuration register (address $0036)

Bit	7	6	5	4	3	2	1	0
R	ADLPC	ADIV1	ADIV0	ADLSMP	MODE1	MODE0	ADICLK1	ADICLK0
W								
Reset ($00)	0	0	0	0	0	0	0	0

Table 142. ADC register field descriptions

Field	Description
7 ADLPC	ADLPC – Low Power Conversion The ADLPC bit enables a low power conversion option. 0 = Normal power selection (improved noise); Result of Reset 1 = Low power option selected
6:5 ADIV[1:0]	ADIV[1:0] – ADC Clock divide ratio The 2 bits ADIV[1:0] select the clock divide ratio: 0 0 = divide by 1; Result of Reset 0 1 = divide by 2 1 0 = divide by 4 1 1 = divide by 8
4 ADSLMP	ADSLMP – Long or Short sample selection The ADLSMP bit selects between long and short sample during conversions. 0 = Short sample is selected; Result of Reset 1 = Long sample is selected

Table 142. ADC register field descriptions*...continued*

Field	Description
3:2 MODE[1:0]	MODE[1:0] – Mode control The two bits MODE[1:0] select the resolution of the converter 0 0 = 8-bit; Result of Reset 0 1 = 12-bit 1 0 = 10-bit 1 1 = 10-bit
1:0 ADICLK[1:0]	ADICLK[1:0] – ADC Clock Source Selection The two bits ADICLK[1:0] select the clock source for the ADC during conversions: 0 0 = bus clock; Result of Reset 0 1 = bus clock divide by 2 1 0 = off-chip clock 1 1 = internal asynchronous clock

10.17.1.6 Port pin control register

Table 143. Port pin control register (address $0037)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	ADPC4	ADPC3	0	0	0
W	—	—	—			—	—	—
Reset ($00)	0	0	0	0	0	0	0	0

Table 144. Port pint control register field descriptions

Field	Description
4 ADPC4	The ADPCTL1 register allows control of the port I/O pin for use as analog inputs. When asserted, the ADC requests control of the pad and disables the default buffers and pull-up / -down. ADPC4 - Connects PTB1 to ADC MUX channel 4 0 = default PTB1 port operation, Result of Reset 1 = PTB1 connected to ADC MUX channel 4
3 ADPC3	ADPC3 - Connects PTB0 to ADC MUX channel 3 0 = default PTB0 port operation, Result of Reset 1 = PTB0 connected to ADC MUX channel 3

10.18 Serial peripheral interface (SPI) module

The SPI module is configured as a standard slave SPI which allows a full duplex, synchronous, serial communication between the unit and a master SPI device.

The principal features of the SPI block are summarized as follows:

- Slave only mode operation.
- Full-duplex, 4 wire, synchronous, serial communication.
- Command-Response communication format.
- SCLK operation up to 10 MHz supported.

- Fixed Clock polarity and phase supported (CPOL=0, CPHA = 0).
  - The SPI module requires that the base clock value be low (CPOL = 0) with data captured on the rising edge of the clock and data propagated on the falling edge of the clock (CPHA = 0).
- Supports 8-bit register read and write operations via 16 clock transfers.
- Even Parity error-checking.
- Alternate bus master for the system-on-chip (SoC) internal IP Bus system.
  - SPI can be used to access the entire Memory map of the NTM88.
- Contains eight, 8-bit memory mapped registers for user and test mode operations.

As a slave, the SPI interface is compatible with SPI interface mode 00, corresponding to CPOL = 0 and CPHA = 0. For CPOL = 0, the idle value of the clock is zero, and the active value of the clock is 1. For CPHA = 0, data is captured on the clock's rising edge (low to high transition) and data is propagated on the clock’s falling edge (high to low transition).

As a slave, the SS_B pin is driven low at the start of a transaction, held low for the duration of the transfer, and then driven high again after the transaction is completed. During a transaction, the master toggles the clock (SCLK). The SCLK polarity is defined as having an idle value that is low, and an active phase that is high (CPOL = 0). Serial input and output data is captured on the clock's rising edge and propagated on the falling edge (CPHA = 0). Single-byte read and single-byte write operations are completed in two strobes of SS_B of 16 SCLK cycles each; multiple byte reads and writes are completed in additional multiples of 16 SCLK cycles. The first SCLK cycle latches the most significant bit on MOSI to select whether the desired operation is a read (R/W = 1) or a write (R/W = 0). The following 13 SCLK cycles are used to latch the slave register read or write address. The final two SCLK cycles are used to latch the parity calculation results.

When memory has been secured by SEC[1:0] settings, the SPI may access only the address ranges $0000 to $008F and $1800 to $188F. Other access attempts will result in an error status as defined below.

Note: *The SPI and the MCU share the internal address, data, and control bus, and are arbitrated such that the SPI will take priority over the MCU. Care must be taken by the user application to account for inhibited execution of MCU instructions during the time that the SPI has taken control of the internal bus.*

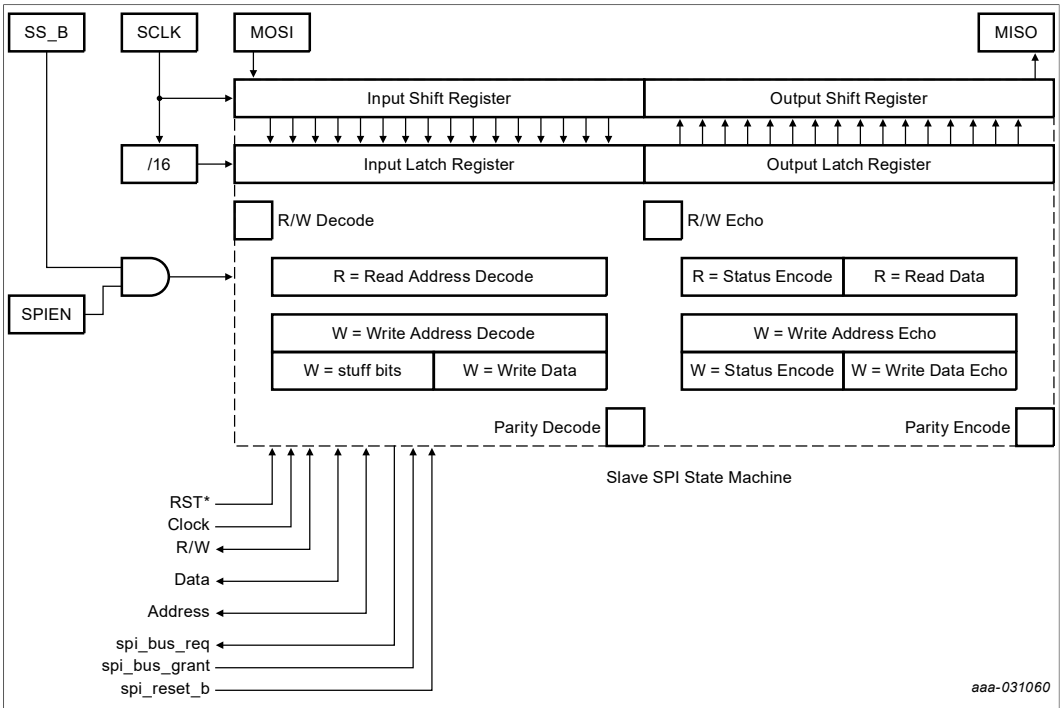


Figure 41. Slave SPI state machine

Transaction event schedule; T1 being the first master transmission, R1 being the first slave response being concurrent with T2 being the second master transmission:

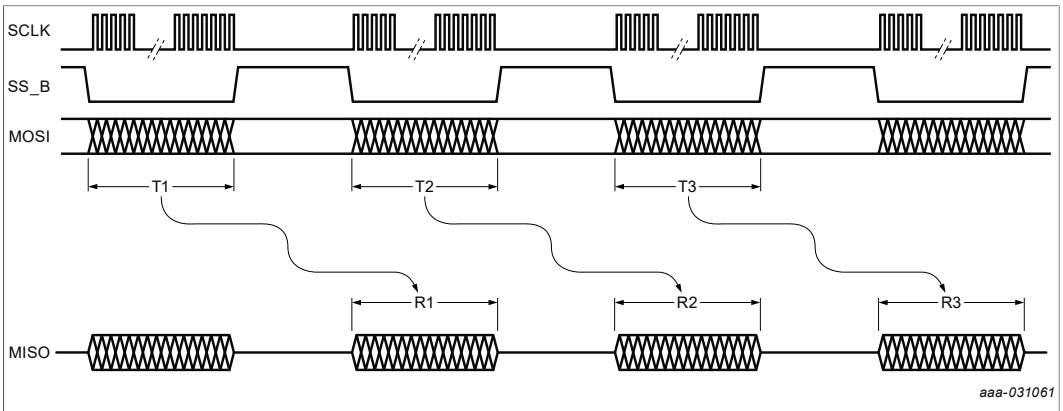


Figure 42. SPI message response protocol

10.18.1 SPI protocol definition

Clock cycle		c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16
Bit assignment		**b15**	**b14**	**b13**	**b12**	**b11**	**b10**	**b9**	**b8**	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
Read byte from Address	T1	0	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0	p1	p0
	R0	r	s4	s3	s2	s1	s0	r	r	r	r	r	r	r	r	p1	p0
	T2	t	t	t	t	t	t	t	t	t	t	t	t	t	t	p1	p0
	R1	0	s4	s3	s2	s1	s0	d7	d6	d5	d4	d3	d2	d1	d0	p1	p0

Clock cycle		c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16
Bit assignment		**b15**	**b14**	**b13**	**b12**	**b11**	**b10**	**b9**	**b8**	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
Write byte to Address	T1	1	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0	p1	p0
	R0	r	s4	s3	s2	s1	s0	r	r	r	r	r	r	r	r	p1	p0
	T2	1	m	m	m	m	m	d7	d6	d5	d4	d3	d2	d1	d0	p1	p0
	R1	1	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0	p1	p0
	T3	t	t	t	t	t	t	t	t	t	t	t	t	t	t	p1	p0
	R2	1	s4	s3	s2	s1	s0	d7	d6	d5	d4	d3	d2	d1	d0	p1	p0

Where:

c1 – c16 = SCLK cycles 1 (b15) through 16 (b0), most significant bit first, least significant bit last

b15:0 = bit assignments for each clock cycle, b15 = 0 for read; b15 = 1 for write

a12:0 = 13 LSB's of address being read or written; $0000 to $1FFF is direct; $C000 to $FFFF is indirect

p1:0 = "Even" parity bits, p1 calculated for contents of b15:9; p0 calculated for contents of b8:2

s4:0 = slave status:

0 0 0 0 0 = all OK, no need for re-try.

1 x x x x = reserved for future fault modes, default to 0 until defined

x 1 x x x =

the response in R0 for first T1 input after reset

in the case of commands ignored by SPI due to error in previous read command; i.e. invalid data in response

in the case the write command did not execute

x x 1 x x = clock fault, not enough clocks or too many clocks per SS_B cycle

x x x 1 x = parity fault from either p1 or p0

x x x x 1 = internal bus contention fault, SPI does not gain access to peripheral bus in the prescribed time, or attempt access illegal or security-blocked address

d7:0 = data being read or written

t = contents of next master transmission T#+1

m = master stuff bits, 0 or 1 by master choice, and included as part of parity calculation

r = contents of previous slave response R#-1

10.18.2 SPI signal timing definition

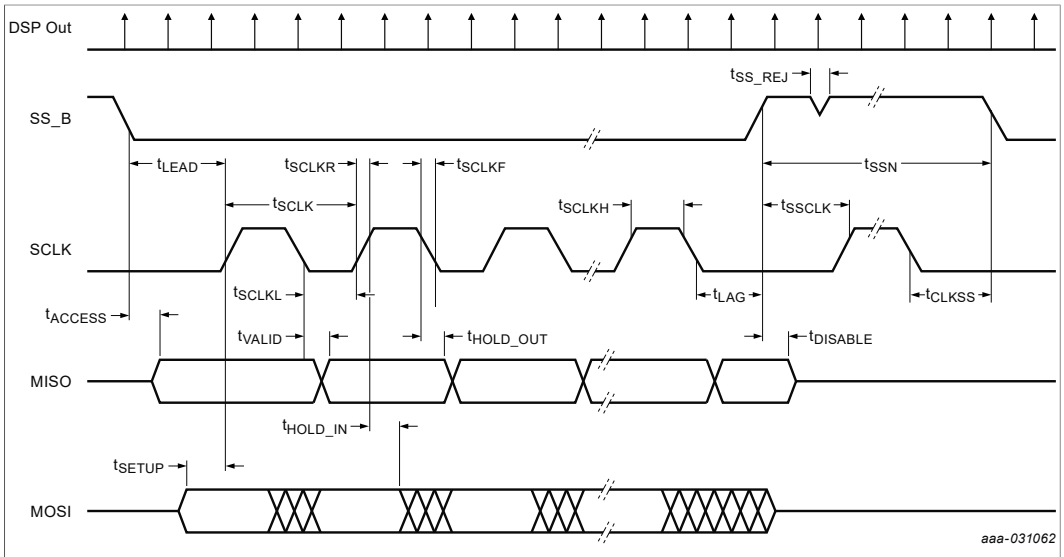


Figure 43. SPI signal timing diagram

SPI may be enabled by either of two methods:

1. MCU application software write to ‘1’ the SPIEN control bit at address $1802
2. At power application, an external host holding the PTA0 pin low for greater than the time t_{SPI_EN}

Care must be taken by the user application to assure the SPI is not disabled by writing '0' to SPIEN bit, or by entering a stop mode, during an ongoing transmission. SPI can be disabled when SS_B signal is in the inactive state, or high.

10.19 Sensor measurement interface (SMI) module

The sensor measurement interface (SMI) module samples, filters, and scales the transducer signals into an analog signal suitable for input to the ADC module. The SMI can be controlled to select the desired transducer channel, filter characteristics, and gain settings.

The SMI has two operating modes available to users:

- Standby, meaning powered but not measuring
- Measurement mode

The standby mode is defined by the configuration of the following control bits:

- SMIEN bit set to 0
- SMIF bit is = 0
- SEN bit is set to 0
- SCAP bit is set to 0
- SST[2:0] bits are set to 0 0 0

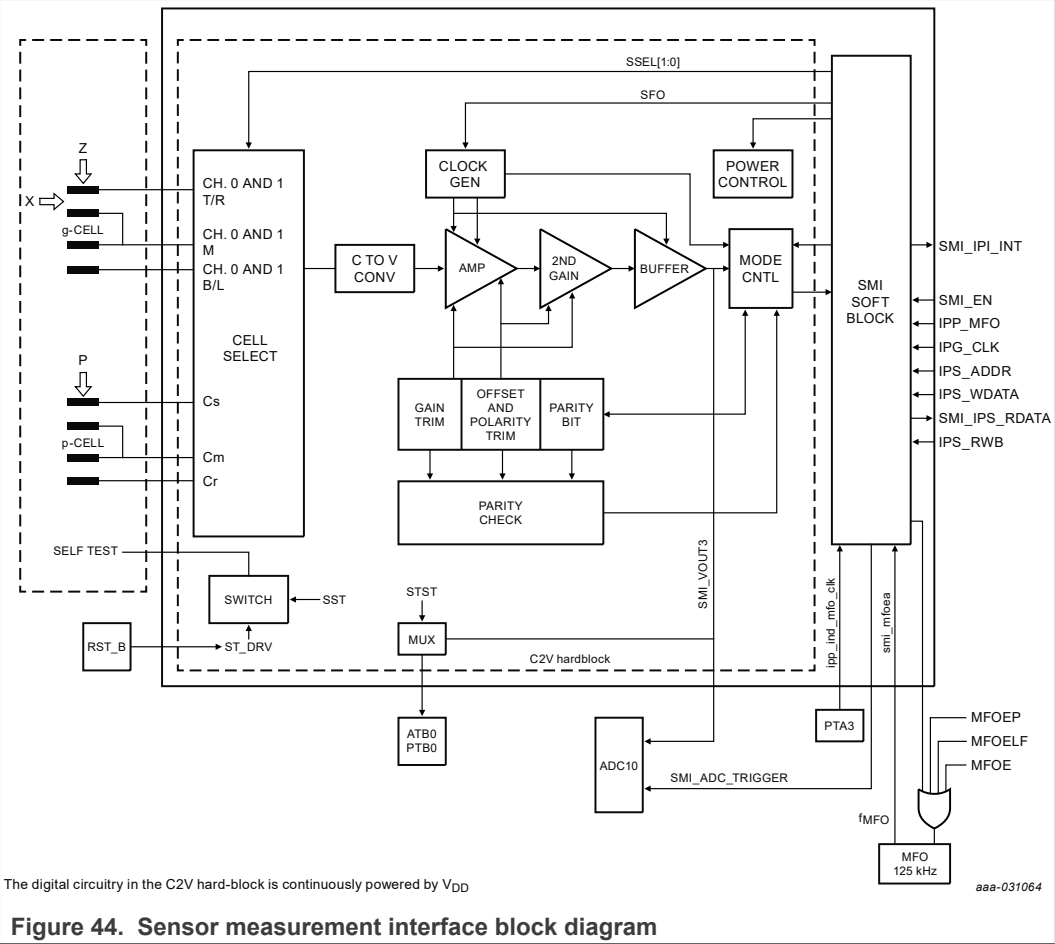


Figure 44. Sensor measurement interface block diagram

10.19.1 SMI signal measurement modes

In the sensor signal measurement mode, the transducer (g-cell or p-cell) and its interface circuit are powered up and the resulting analog voltage appears at the ADC after some time delay. This process can be automatically triggered or controlled directly by the MCU.

The signal measurement mode provides three sub-modes:

- Automatic where the internal hardware circuits control a singular measurement (transducer voltage capture and ADC conversion trigger).
- Direct where the software application controls a series of measurements (sequence of transducer voltage captures and the ADC conversion triggers).
- Low power direct (LPDM) where the software application sets up the initial and subsequent settling times, and then the internal hardware circuits control the series of measurements.

Table 145 summarizes the compatibility of the SMI measurement modes with the MCU operational modes. Since the SMI is disabled during STOP1, the SMI cannot be used when the MCU is in STOP1. All modes are available when the MCU is in RUN mode, although higher noise or offset may be observed. Therefore STOP4 is recommended for highest accuracy. When in the automatic mode, the user application (MCU in Run) controls the trigger of the measurement via the SEN bit. When in the direct modes, the user application (MCU in Run) controls the trigger of the ADC via the SCAP bit. If the application intends a period of Stop4 between the SMI completion and the subsequent

ADC trigger, then a timer such as FRC, PWU, or RTI has to be set to generate the wake up, followed by the ADC trigger in the run mode. The ADC will subsequently generate the COCO signal and, if enabled, its associated interrupt. Note in all cases, the ADC COCO bit will be cleared upon the ADC interrupt service routine reading the ADCRES Low register.

Table 145. Signal measurement sub-modes

SMI Sub-mode	MCU RUN	MCU STOP4	MCU STOP1	Comments
Automatic	Yes	Yes	No	SMI terminated after a single measurement.
Direct	Yes	No	No	SMI signal chain runs continuously, measurements are triggered by the MCU. Stop4 entry not recommended.
LPD	Yes	Yes	No	SMI signal chain runs continuously, measurements are repeatedly triggered automatically by the hardware.

10.19.1.1 SMI automatic signal measurement mode

In the automatic mode of operation the sensor signal measurement is triggered by setting the SEN bit (SMICS[4]) which generates a series of signals given in Figure 45.

The typical sensor measurement software sequence would be to first select the SMI ADC multiplexer channel, select the SMI transducer channel, filter, and Initial Settling Delay, then set the SEN bit, then poll the SEN bit to be sure it is set, and then go into the STOP4 mode.

When the measurement is initialized and the sample delay is completed, the ADC Trigger signal is generated. When ADC finishes a conversion, the ADC_COCO interrupt will wake up the MCU, and the converted result will be available at the ADCRES H/L registers. After the ADC conversion, the SEN bit (SMICS[4]) will be cleared, which in turn powers down the transducer (g-cell or p-cell), and the software can then use the resulting ADC data as desired. As mentioned, this must be completed within the period noted as t_{FIN}. Once the t_{FIN} has completed, the software may then initiate a new measurement cycle by setting the SEN bit, polling the SEN bit to be sure it is set, then entering the STOP4 mode.

Notice in this configuration that the SEN bit will be automatically cleared by the internal signal SFO being low or cleared. Then the internal signal CLR_SEN_SYNC signifies that the SEN signal may be written for the next measurement cycle. Since the CLR_SEN_SYNC signal is not available for the software to access, the software has to poll the SEN bit until it is clear, then wait for a sufficient delay time for the CLR_SEN_SYNC to clear. Since the ADC interrupt service routine will be short, the delay while the SEN bit is clear is recommended to be the sum of the periods t_{S4WU} + t_{FIN}.

Please review the corresponding data sheet for the periods noted as t_{INIT}, t_{MEAS}, T_{ADC}, t_{S4WU}, and t_{FIN}.

In this configuration, the SMIF bit (SMICS[7]) is not available.

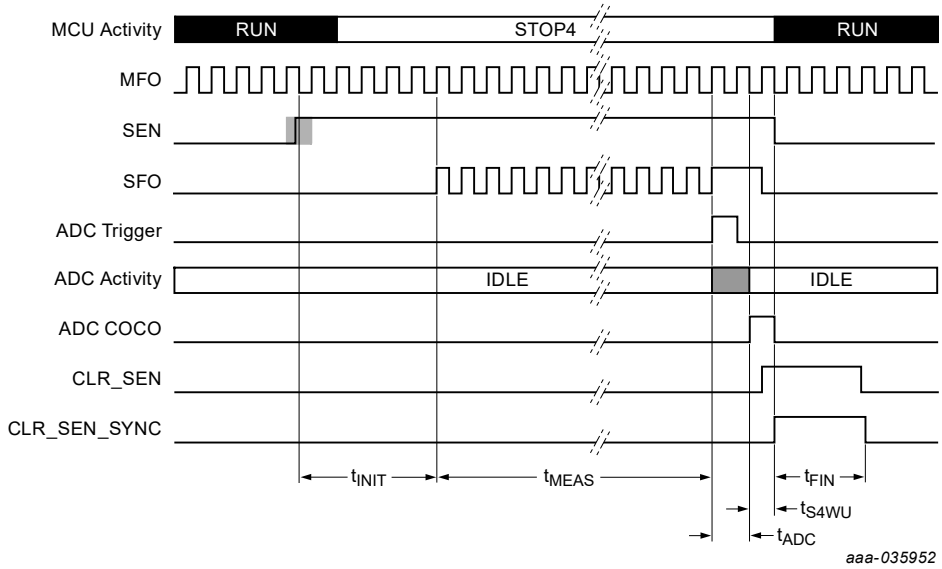


Figure 45. Automatic measurement protocol

10.19.1.2 SMI direct sensor signal measurement mode

In the direct mode of operation, the sensor signal measurement can be directly controlled by the user software by managing the SMIC[7] (SMIEN) and SMIC[6] (SCAP) control bits as shown in Figure 46.

For applications that require accurate measurements, direct mode is not recommended because MCU clock activity induces ripple on the supply and ground that can increase the noise and degrade the accuracy of the measurements. Low power direct mode or automatic mode produce lower noise and offset because the measurements can be taken when the MCU is in STOP4 mode, where its clock is halted.

The typical sensor measurement software sequence would be to first select the SMI ADC multiplexer channel, select the SMI transducer channel, and filter. The software application then sets the SMIEN bit, then delays a period t_{AC}, then sets the SCAP bit. The SMI will then use the SCAP bit being set to synchronize, such that after the period t_{CE} the ADC Trigger will be set. The SCAP bit will be cleared after one full clock cycle. The ADC will then complete the conversion and the ADC COCO interrupt will be asserted.

At this point, the application may choose to stop measurements by clearing the SMIEN bit. Or, if additional measurements are desired, the software application may simply set the SCAP bit again to repeat the next measurement.

Note that if the software application will change any settings within the SMI, such as the transducer channel or filter, then the application should again delay by the period t_{AC} until setting the SCAP bit.

Also note that the SMIEN cannot be cleared while the SCAP bit remains set. The recommended procedure is to poll the SCAP bit until it is clear before clearing the SMIEN bit. After the SMIEN bit is cleared, the system returns to Standby mode.

Please review the corresponding data sheet for the period noted as $t_{CE.}$

The period t_{AC} is decided by the software application, and should be set sufficient to assure useful results.

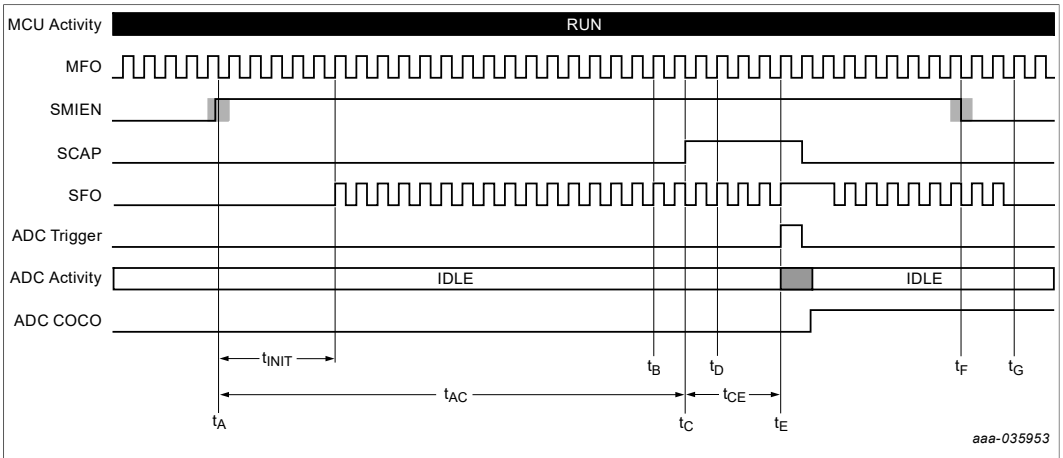


Figure 46. Direct measurement protocol

10.19.1.3 SMI low power direct sensor signal measurement mode

Operationally, two selections are described by the following timing diagram, which assumes the application desires a series of repeating measurements denoted by the end of the t_{ADC} times. The first measurement is available from the ADC after the initial settling delay plus the conversion time; i.e. after t_{ISD} + t_{ADC}. Then each of the subsequent measurements become available from the ADC after each t_{SP} period. The period marked by the term t_{ADC} includes ~15 µs for the ADC to convert, plus ~4 us to wake up, plus ~33 µs for the ADC tasks to finish. The t_{ADC} periods run concurrently with each t_{SP}. The subsequent sample periods start after the prior sample period ADC cycle has finished. Firmware that controls the SMI and ADC must collect the result value from the ADCRES H/L registers and store in either a parameter register or a RAM location. These tasks must be completed before the next t_{SP} begins.

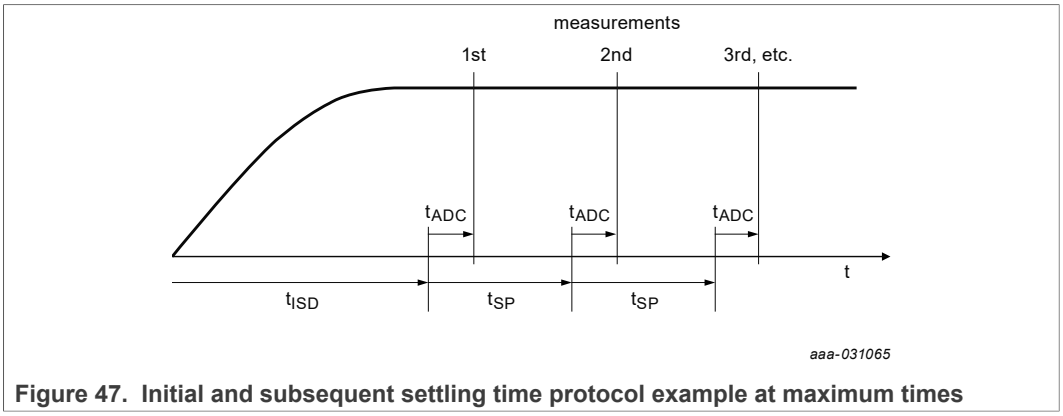


Figure 47. Initial and subsequent settling time protocol example at maximum times

Also, notice that t_{ISD} is depicted as a rising signal level which reaches a steady state. The SMI is designed such that the shortest t_{ISD} selections will commence the measurement process prior to the signal reaching the steady state. This is provided to allow users the option of very high sampling rates, where absolute signal accuracy is typically not required; i.e. the user application is more concerned with quickly checking relative rate of change, and can tolerate reduced absolute accuracy. Typical use cases could be checking for accelerometer changes, and not executing the compensation firmware to save power.

In the low power direct mode of operation, the sensor signal measurement is automated after the user software manages the SMICS[3] (LPDM), SMIC[7] (SMIEN) and SMIC[6] (SCAP) control bits as shown in Figure 48.

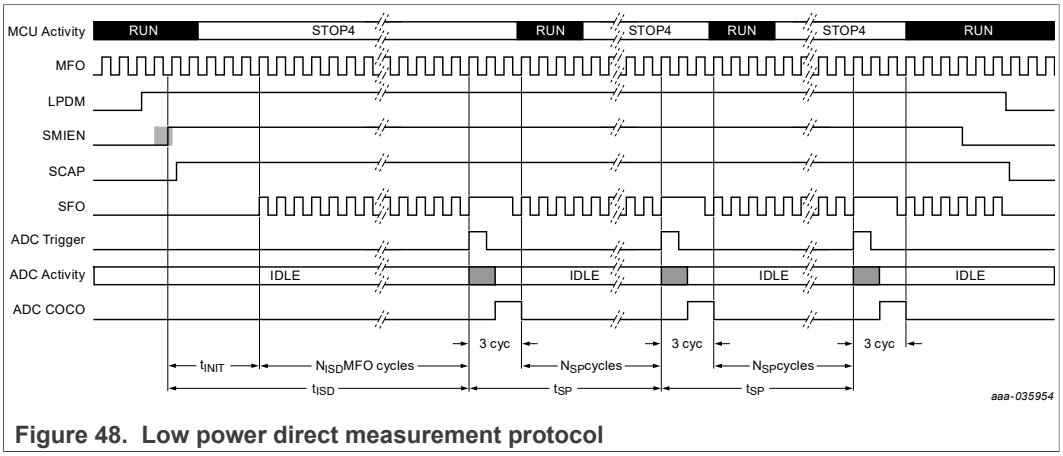


Figure 48. Low power direct measurement protocol

The typical sensor measurement software sequence would be to first select the SMI ADC multiplexer channel, select the SMI transducer channel, and filter. Then execute the sequence to enter LPDM as described below. In this mode, due to the MCU entering STOP4 mode between measurements, the initial and subsequent delays are also to be configured and used by the system. The software application then sets the SMIEN bit, and then sets the SCAP bit. The SMI will then use the SCAP bit being set to synchronize, such that after the period t_{ISD} (on the 1st measurement) the ADC Trigger will be set. The SCAP bit will be cleared by the software application exiting LPDM mode. The ADC will then complete the conversion and the ADC COCO interrupt will be asserted. The ADC COCO signal is cleared when the software reads the results from the ADCRES H/L registers.

At this point, the application may choose to stop measurements by clearing the SMIEN bit and exiting the LPDM mode. LPDM mode is exited by clearing the SMIEN and the LPDM bits on two consecutive write operations performed immediately after the MCU wakes from STOP4. The SCAP bit will automatically clear after the LPDM bit goes low.

Notice the second and subsequent settling periods t_{SP} begin at the prior measurement ADC Trigger bit being set. The process will continue until the software application clears the SMIEN bit and exits the LPDM mode.

Note that if the software application will change any settings within the SMI, such as the transducer channel or filter, then the application should again use the longer ISD to assure the signal is allowed to settle. Each new entry into LPDM will cause the ISD delay to be employed on the first measurement, followed in subsequent measurements by the SP delay.

Please review the corresponding data sheet for the periods noted as t_{ISD} and t_{SP}.

The LPDM bit will allow the system to acquire data as on the Direct Mode, but the MCU can go to STOP4, since the hardware will control the acquisition time. The sequence to get in this mode is important and must be three consecutive writes on the bits SMICS[3] (LPDM), SMIC[7] (SMIEN), and SMIC[6] (SCAP), then the user software can enter STOP4.

1. Set LPDM bit to 1, then
2. Set SMIEN bit to 1, then
3. Set SCAP bit to 1, then Enter STOP4 mode.

The ISD and SP intervals can be modified between two consecutive reads, but it must be done immediately after the MCU wakes-up from the ADC interruption.

To go out the Low Power Direct Mode, immediately after the MCU wakes-up, the user software low power direct mode routine must reset the SMIEN bit, and then reset the LPDM bit on two consecutive writes. The SCAP bit will be automatically cleared when LPDM bit goes low.

After attending to the ADC interrupts, the MCU is allowed to go to STOP4.

10.19.2 SMI register descriptions

10.19.2.1 SMI status and control register (SMICS)

Table 146. SMI status and control register (SMICS) (address $0040)

Bit	7	6	5	4	3	2	1	0
R	SMIF	0	SMIIE	SEN	LPDM	0	0	0
W	—	SMIFAK		1		—	—	—
Reset ($00)	0	0	0	0	0	0	0	0

Table 147. SMICS register field descriptions

Field	Description
7 SMIF	SMIF — SMI Interrupt Flag The SMIF bit indicates when an interrupt has been generated by the SMI at the end of the settling delay, either initial or subsequent only when all of the following conditions exist: • In the MCU run mode and • SMI interrupt enable is enabled (set to 1) and • The SMI low power direct mode is full (set to 0) This bit is cleared by writing a one to the SMIFAK bit. 0 = SMI interrupt not generated or previously acknowledged; Result of Reset 1 = SMI interrupt generated ***Note:*** *Since the SMI Interrupt function is disabled if the MCU is in a Stop mode, or if the Low Power Direct Mode is enabled, users should avoid using the SMI Interrupt function, and instead rely on the ADC COCO function to signify completion of the measurement. Enabling the SMI Interrupt functions will induce unwanted noise or offset into the measurements, and therefore should be avoided in cases where the highest accuracy is needed.*
6 SMIFAK	SMIFAK — Acknowledge SMIF Interrupt Flag The SMIFAK bit clears the SMIF bit if written with a one. Writing a zero to the SMIFAK bit has no effect on the SMIF bit. Reading the SMIFAK bit returns a zero. Reset has no effect on this bit. 0 = No effect 1 = Write 1 to clear SMIF for SMI interrupt acknowledge.
5 SMIIE	SMIIE — SMI Interrupt Enable The SMIIE bit enables the generation of the SMI interrupt when the acquisition cycle is complete. The SMIIE bit has no effect if the SEN bit is clear. 0 = Disable SMI interrupt; Result of Reset 1 = Enable SMI interrupt if SEN is also = 1.

Table 147. SMICS register field descriptions*...continued*

Field	Description
4 SEN	SEN — Sensor Acquisition Enable When the SMI is idle, the SEN bit initiates an acceleration or pressure reading. When the SEN bit is set to 1, a new acquisition cycle is started. When read, the SEN bit indicates the state of a measurement. If the SEN bit is 0, the measurement is complete or idle; if the SEN bit is 1, the measurement is presently underway. 0 = Disable sensor (g-cell or p-cell) and C2V; Result of Reset 1 = Enable C2V and start measurement cycle
3 LPDM	LPDM — Low Power Direct Mode The purpose of LPDM bit is to release the MCU of control the data acquisition and enter STOP4 mode during direct mode. This allows the system to acquire data in the direct mode while in STOP4. 0 = Full Power Direct Mode Enabled. (RUN mode only); Result of Reset 1 = Low Power Direct Mode Enabled. (STOP4 allowed)

10.19.2.2 SMI control register (SMIC)

Table 148. SMI control register (SMIC) (address $0041)

Bit	7	6	5	4	3	2	1	0
R	SMIEN	SCAP	SSEL1	SSEL0	SPAR_ERR	reserved	reserved	reserved
W		1			—			
Reset ($00)	0	0	0	0	0	0	0	0

Table 149. SMIC register field descriptions

Field	Description
7 SMIEN	SMIEN — SMI Enable The SMIEN bit controls the start of a sensor measurement without any automatic delay or trigger of the ADC capture. This bit cannot be set to 1 if the SEN or LPDM bits are set to 1. To enter Direct Mode, the SEN and LPDM bits should both be set to 0 before the SMIEN bit is set to 1. To exit Direct Mode, poll until the SCAP bit is = 0. Then set the SMIEN to 0. Entering LPDM Mode requires multiple steps. First, any auto-mode measurement cycle must be allowed to complete. Therefore, the SEN bit must be = 0. If the SEN bit is <> 0, poll until it = 0. Next, set the LPDM bit to 1. Then, set the SMIEN bit to 1. Finally, in a subsequent write command, set the SCAP bit to 1. To exit LPDM mode, set the SMIEN bit to 0, and then set the LPDM bit to 0 on two consecutive writes. The SCAP bit will be automatically = 0 when LPDM bit goes to 0. 0 = SMI not in Direct or LPDM modes (i.e. idle); Result of Reset 1 = SMI in either Direct or LPDM mode (i.e. measurement in progress).
6 SCAP	SCAP — Sensor Capture In auto-mode, i.e. when SEN is = 1, writing the SCAP bit will have no effect. In direct-mode, ADC conversions are triggered by writing 1 to the SCAP bit. The bit is self-clearing. In LPDM mode, The SCAP bit is set to 1 on entry to LPDM mode. Once written, the bit remains = 1 while a state machine automatically generates triggers for the ADC. The SCAP bit will self-clear after the LPDM exit procedure of setting to 0 the SMIEN and LPDM bits on two successive writes. 0 = ADC capture disabled; Result of Reset 1 = ADC capture enabled

Table 149. SMIC register field descriptions*...continued*

Field	Description
5:4 SSEL[1:0]	SSEL[1:0] – Sensor Selection The two bits SSEL[1:0] control which channel of the C-V multiplexer will be selected for conversion by the ADC block. 0 0 = P-cell selected; Result of Reset 0 1 = G-cell South or 1 selected 1 0 = G-cell North or 0 selected 1 1 = PRT option selected ***Note:*** *Refer to the NTM88 specific data sheet for the available configuration by part number.*
3 SPAR_ERR	SPAR_ERR – SMI Parity Error The SPAR_ERR bit indicates if a parity error has been detected. As a result, the ADC reading will be exactly or near 0x0000 when a parity error has occurred. To become aware of the error, the user application should check the SPAR_ERR flag, whenever the SMI returns values that are near or exactly 0x0000. 0 = no parity error detected; Result of Reset 1 = parity error detected

10.19.2.3 SMI configuration register (SMICFG)

Table 150. SMI configuration register (SMICFG) (address $0042)

Bit	7	6	5	4	3	2	1	0
R	ITRIG	ITIG_EN	0	0	0	FILEN	FILT1	FILT0
W			—	—	—			
Reset ($00)	0	0	0	0	0	0	0	0

Table 151. SMICFG register field descriptions

Field	Description
7 ITRIG	ITRIG — Interval Trigger (write-only) The ITRIG bit triggers interval timing to commence when in Low Power Direct Mode, provided the Interval Trigger feature is enabled. When this bit is set, it is automatically cleared. 0 = No effect; Result of Reset 1 = Commence interval timing
6 ITRIGEN	ITRIGEN — Interval Trigger Enable (write-only) The ITRIGEN bit enables interval trigger functionality. 0 = Interval trigger disabled; Result of Reset 1 = Interval trigger enabled
2 FILEN	FILEN – Filter Enable The FILEN bit enables or disables the low-pass filter block: 0 = low-pass filter bypassed; Result of Reset 1 = low-pass filter enabled

Table 151. SMICFG register field descriptions*...continued*

Field	Description
1:0 FILT[1:0]	FILT[1:0] – Low-pass Filter selection When the FILEN bit is 1, the two bits FILT[1:0] select the cut-off frequency of the low-pass filter. When the FILEN bit is 0, writing to the FILT[1:0] bits has no effect and default to the low-pass filter being bypassed 0 0 = 250 Hz; Result of Reset. This setting induces an offset until the input signal has completely settled, and therefore consumes additional energy; NXP recommends using the 500 Hz, 1000 Hz, or 2000 Hz settings where absolute accuracy and lowest energy consumption are needed. 0 1 = 500 Hz 1 0 = 1000 Hz 1 1 = 2000 Hz

10.19.2.4 SMI settling time register (SMIST)

Table 152. SMI settling time register (SMIST) (address $0043)

Bit	7	6	5	4	3	2	1	0
R	SP3	SP2	SP1	SP0	ISD3	ISD2	ISD1	ISD0
W								
Reset ($00)	0	0	0	0	0	0	0	0

Table 153. SMIST register field descriptions

Field	Description
7:4 SP[3:0]	SP[3:0] - Subsequent sample delay The SMI settling time register holds the subsequent sample delay (SP) in the four most-significant bits. The function maps a binary pattern programmed by the application to a defined count of MFO cycles approximating the SP time selections. The four most-significant bits are used to program the subsequent sample delay (SP) periods. The SP[3:0] control bits result in the approximate subsequent sample delay periods as follows: 0 0 0 0 = Result of Reset. See Table 154.
3:0 ISD[3:0]	ISD[3:0] - Initial sample delay The SMI settling time register holds the initial sample delay (ISD) period in the four least-significant bits. The function maps a binary pattern programmed by the application to a defined count of MFO cycles approximating the IDS time selections. The four least-significant bits are used to program the initial sample delay (ISD) period. The ISD[3:0] control bits result in the approximate initial sample delay as follows: 0 0 0 0 = Result of Reset. See Table 155.

Table 154. SP[3:0]

SP[3:0]	Number of MFO Clock Cycles	~T_{SP} Nom. Sample Per. (us)
0000	8	64
0001	10	80
0010	13	104
0011	16	128
0100	20	160
0101	25	200

Table 154. SP[3:0]*...continued*

SP[3:0]	Number of MFO Clock Cycles	~T_{SP} Nom. Sample Per. (us)
0110	32	256
0111	40	320
1000	51	408
1001	64	512
1010	81	648
1011	102	816
1100	128	1024
1101	161	1288
1110	203	1624
1111	256	2048

Table 155. ISD[3:0]

ISD[3:0]	Number of MFO Clock Cycles	~T_{ISD} Nom. Sample Per. (µs)
0000	13	104
0001	16	128
0010	20	160
0011	25	200
0100	32	256
0101	40	320
0110	51	480
0111	64	512
1000	81	648
1001	102	816
1010	128	1024
1011	161	1288
1100	203	1624
1101	256	2048
1110	323	2584
1111	406	3248

Note: *The initial sample delay settings less than 2584 µs are intended for use cases where absolute accuracy is not required, e.g. relative changes are being measured. Therefore, the published data sheet tolerances are valid for only the initial sample delay settings of 2584, or 3248 µs, and the MCU is placed into the STOP4 mode. All other settings may return an increased offset when compared to an absolute reference.*

10.20 Parameter registers (PARAM0 to PARAM63)

The 64 bytes of parameter registers are located at addresses $0050 through $008F. The 64 bytes of parameter registers are kept active in all modes of operation as long as power is applied to the supply pins. The contents of the parameter registers behave like RAM and are unaffected by any reset. Parameter registers may be used to store temporary or history data during the times that the MCU is in any of the STOP modes. The parameter register at $008F is used by the firmware for interrupt flags.

Table 156. Parameter registers (PARAM0 to PARAM31) (addresses $0050 - $006F)

Bit	7	6	5	4	3	2	1	0
R	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
W								

Table 157. Parameter registers (PARAM32 to PARAM63) (addresses $0070 - $008F)

Bit	7	6	5	4	3	2	1	0
R	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
W								

10.21 Random access memory (RAM0 to RAM511)

The 512 bytes of RAM registers are located at address $0090 through $028F. The locations in RAM from $0090 to $00FF can be accessed using the more efficient direct addressing mode, and any single bit in this area can be accessed with the bit-manipulation instructions (BCLR, BSET, BRCLR, and BRSET). Locating the most frequently accessed program variables in this area of RAM is preferred.

The RAM retains data when the MCU is in low-power WAIT or STOP4 modes. At power-on or after wake-up from STOP1, the contents of RAM will not be initialized. RAM data is unaffected by any reset provided that the supply voltage does not drop below the minimum value for RAM retention (VRAM).

When security is enabled, the RAM is considered a secure memory resource and is not accessible through BDM or through code executing from non-secure memory. None of the RAM locations are used directly by the firmware. The firmware routines utilize RAM only through stack operations; and the user needs to be aware of stack depth required by each routine as described in the CodeWarrior project files.

Table 158. Random access memory registers (RAM0 to RAM255) (addresses $0090 - $00FF)

Bit	7	6	5	4	3	2	1	0
R	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
W								

Table 159. Parameter registers (RAM256 to RAM511) (addresses $0100 - $028F)

Bit	7	6	5	4	3	2	1	0
R	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
W								

10.22 System integration module (SIM)

The SIM_908SZK16 is a chip-specific module targeted for integration into the HC9S08SZK16 MCU. The SIM_908SZK16 is partitioned into many functional blocks. The SIM is divided into IPBI (IP Bus Interface), COP (Computer Operating Properly), BEC (Background Entry Controller), OMC (Operating Mode Controller), SPRC (Stop, POR, and Reset Controller), RCG (Registers and Clock Gating), and GL (Glue Logic) blocks.

The SIM performs the following functions:

- Provides the interface between the S08 bus and the IPbus
- Controls system resets from internal and external sources.
- Computer Operating Properly (COP) timer with 2 selectable clock sources and 8 selectable timeout periods for each clock source.
- Selectable Bus clock divisor to be 2, 4, 8 and 16.
- Supplies system clocks based on either internal clock source or external clock inputs.
- Controls POR, low power, and Stop mode events.
- Generates a synchronous fixed frequency enable signal.
- Detects operating mode from single-pin in user operation.
- Supplies mode dependent signals for scan and functional device testing.

The SIM operates by the following modes:

- Run
- Stop
  - Controls System level STOP entry and exit sequences.
  - Disable the system clocks.
  - Disable the internal voltage regulator and internal clock source.
  - Controls the stop exit sequence due to asynchronous detection of interrupt sources or asynchronous external reset.
- Background Debug Mode (BDM)
  - The Computer Operating Properly (COP) timer is suspended.
- Reset
  - Detect reset and perform a controlled exit from reset.
  - Disable clock generation after extended assertion of the RST_B pin.
  - Enable mode capture logic.

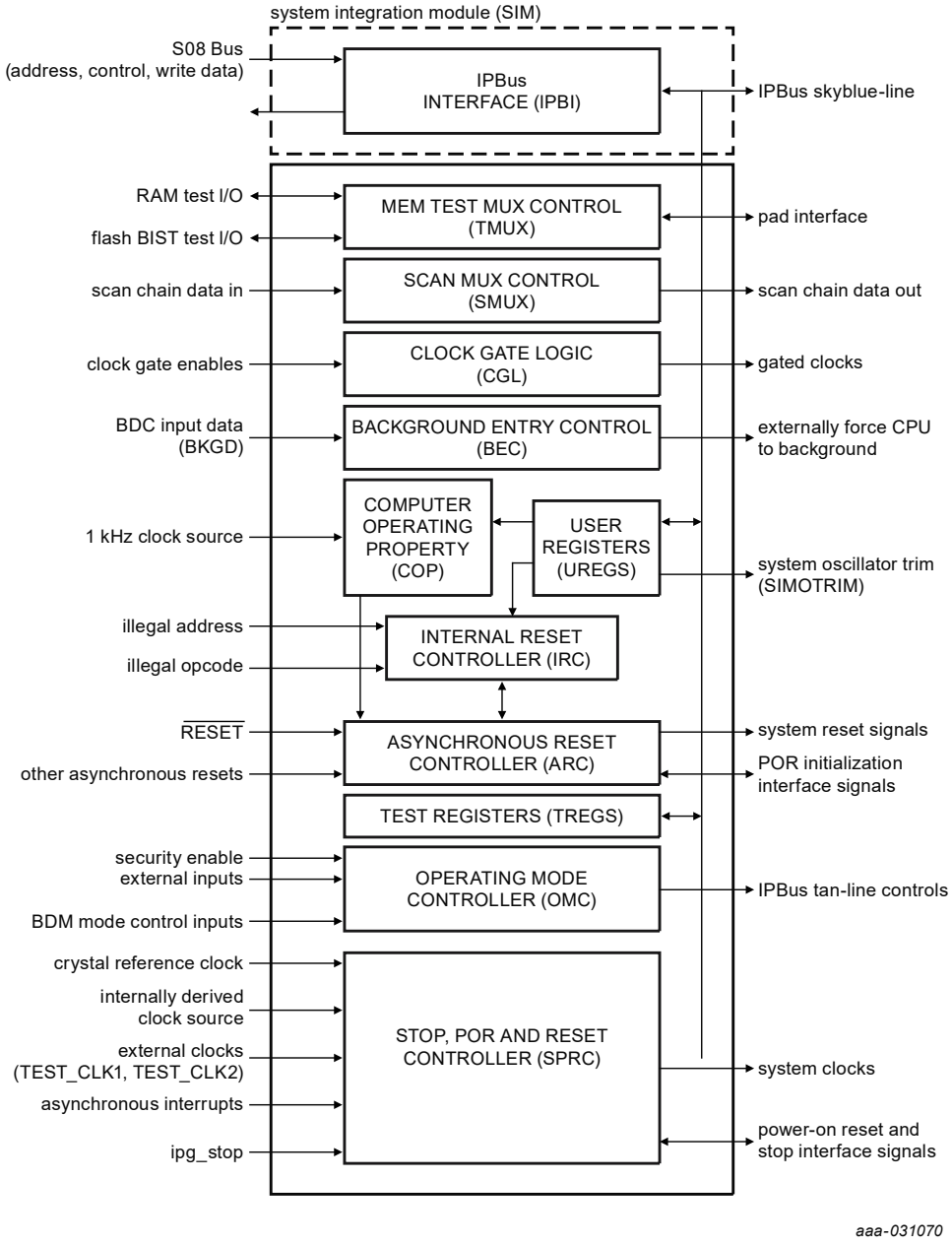


Figure 49. System integration module block diagram

10.22.1 SIM reset exit

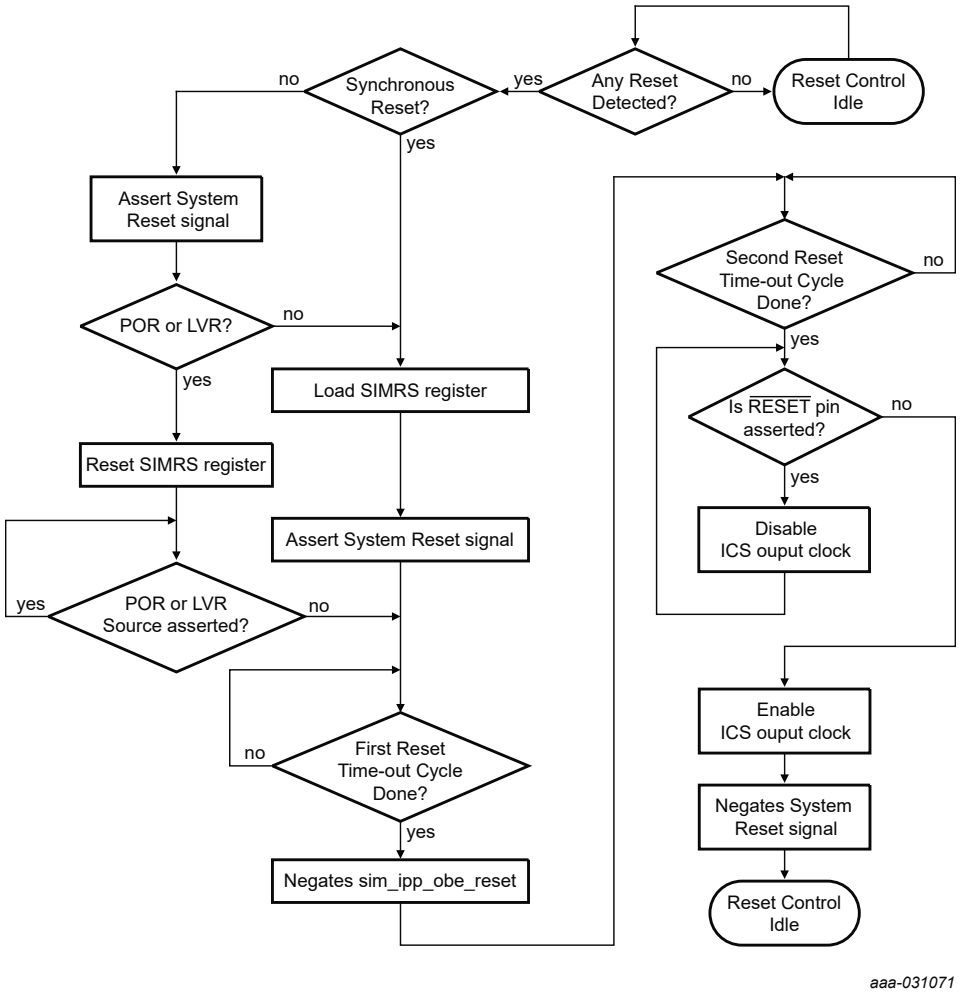
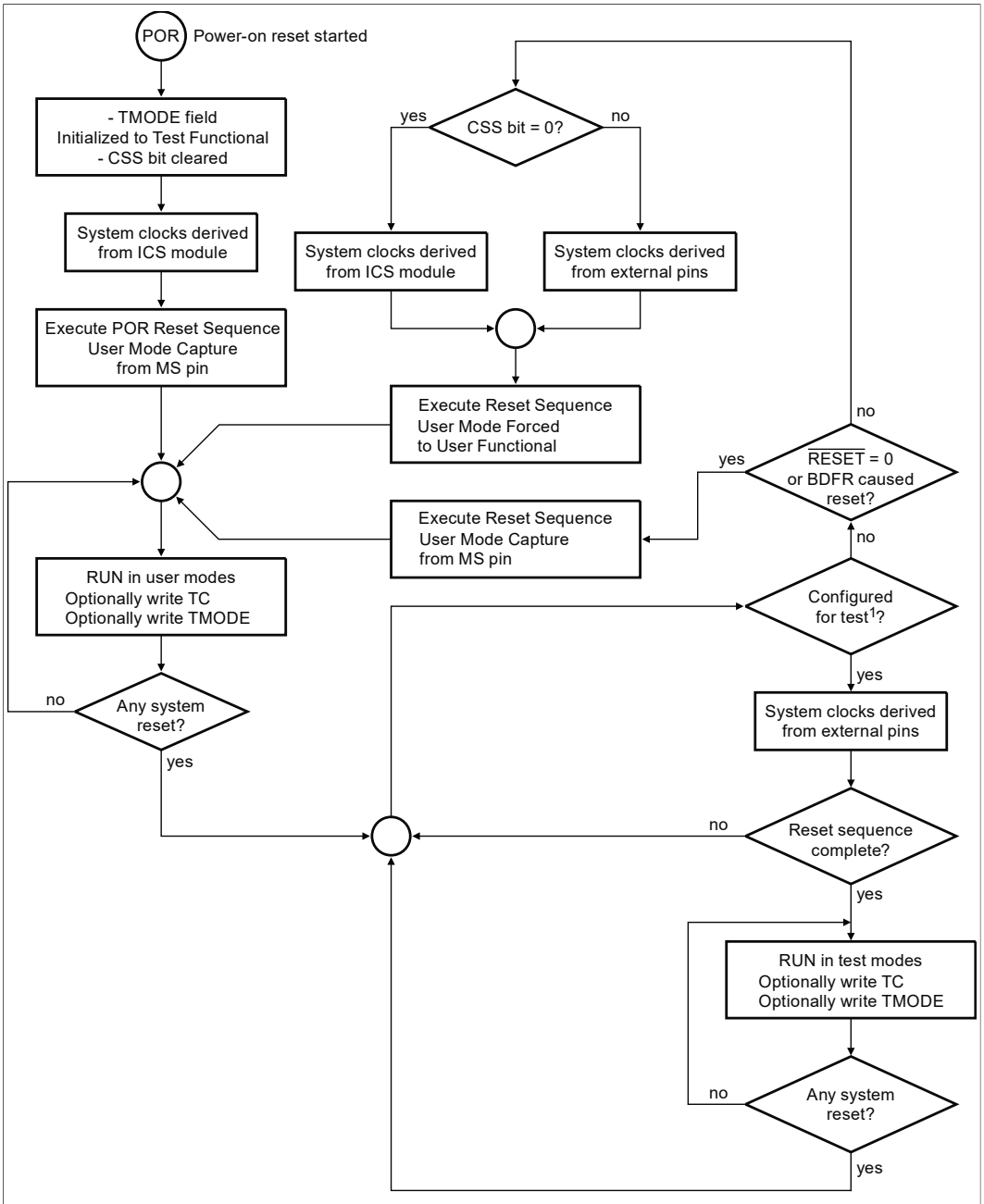


Figure 50. SIM reset exit diagram

10.22.2 SIM MCU mode control



1. The system is configured for test when the TC field in the SIMTCSC register is configured appropriately.

aaa-031072

Figure 51. SIM MCU mode control

10.22.3 SIM register descriptions

10.22.3.1 SIM reset status register (SIMRS)

Table 160. SIM reset status register (SIMRS) (address $1800)

Bit	7	6	5	4	3	2	1	0
R	POR	PIN	COP	ILOP	ILAD	PWU	LVR	SOFT
W								
POR ($82)	1	0	0	0	0	0	1	0

Table 161. SIMRS register field descriptions

Field	Description
7 POR	POR — Power-On Reset Bit The POR bit indicates a power-on reset has been detected. Due to internal supplies ramping, the LVR bit may also be set. 0 = Power-on reset not detected; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = Power-on reset detected; Result of power-on reset.
6 PIN	PIN — External PIN Reset Bit The PIN bit indicates an external reset has been detected on the RST_B pin. 0 = External reset not detected; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = External reset detected.
5 COP	COP — Computer Operating Properly Reset Bit The COP bit indicates a COP reset has been detected. 0 = COP reset not detected; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = COP reset detected.
4 ILOP	ILOP — Illegal Opcode Reset Bit The ILOP bit indicates an illegal opcode reset has been detected. ***Note:*** *The STOP instruction is considered illegal if it has been disabled by STOPE = 0. The BGND instruction is considered illegal if Active Background mode has been disabled by ENBDM = 0.* 0 = Illegal opcode reset not detected; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = Illegal opcode reset detected.
3 ILAD	ILAD — Illegal Address Reset Bit The ILAD bit indicates an illegal address reset has been detected. 0 = Illegal address reset not detected; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = Illegal address reset detected.

Table 161. SIMRS register field descriptions*...continued*

Field	Description
2 PWU	PWU – Power Wake-up Module Reset Bit The PWU bit indicates a PWU reset has been detected. 0 = PWU reset not detected; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = PWU reset detected.
1 LVR	LVR — Low Voltage Reset Bit The LVR bit indicates a low voltage reset has been detected. 0 = Low voltage reset not detected; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = Low voltage reset detected; Result of power-on reset.
0 SOFT	SOFT – Soft Reset Bit 0 = Soft reset not detected or result of subsequent read after initial read; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = Soft reset detected; Result of initial read in the case the Soft Reset function had been triggered.

10.22.3.2 SIM control register (SIMC)

Table 162. SIM control register (SIMC) (address $1801)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	—
W	—	—	—	—	—	—	—	BDFR
Reset ($00)	0	0	0	0	0	0	0	0

Table 163. SIMC register field descriptions

Field	Description
0 BDFR	BDFR — Background Debug Forced Reset Bit While in background debug mode, a write to the BDFR bit forces a system reset. Resetting by this method is directly controlled by the user and does not result in the setting of any bits of the SIMRS register unless another reset source becomes pending on the same cycle. 0 = Do not force system reset; Result of Reset 1 = Force a system reset.

10.22.3.3 SIM option 1 register (SIMOPT1)

A true power-on reset (POR) is caused by first applying power to chip pins or remove and re-apply power to pins. Whereas exit of a Stop mode is not a POR. The bits COPE, COPCLKS and STOPE of SIMOPT1 register are write-once bits. The bits RFEN, SPIEN, and BKGDPE can be written anytime.

Table 164. SIM option 1 register (SIMOPT1) (address $1802)

Bit	7	6	5	4	3	2	1	0
R	COPE	COPCLKS	STOPE	RFEN	0	SPIEN	BKGDPE	0
W					—			—
Reset	1	0	0	U	0	0	1	1
POR ($83)	1	0	0	0	0	0	1	1

Table 165. SIMOPT1 register field descriptions

Field	Description
7 COPE	COPE — COP Enable Bit The COPE bit enables the COP. 0 = COP disabled 1 = COP enabled; Result of Reset
6 COPCLKS	COPCLKS — COP Clock Selection This bit selects the clock source of the COP watchdog. 0 = Internal 1 kHz clock is sourced to COP; Result of Reset 1 = Bus clock is sourced to COP.
5 STOPE	STOPE — STOP Enable Bit The STOPE bit enables the STOP instruction. If the STOP instruction is disabled and a STOP instruction is executed, then an illegal opcode reset occurs. The STOPE bit is writable only once after each exit from a system reset. 0 = STOP instruction disabled; Result of Reset 1 = STOP instruction enabled.
4 RFEN	RFEN — RF Transmitter Enable Bit This bit enables the RF module. 0 = RF disabled; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = RF enabled.
2 SPIEN	SPIEN — SPI Enable Bit This bit enables the SPI module. Result of enabling the SPI is the Port A is configured for SPI IOs. Also, the SPI is enabled if the PTA0 pin is held low for > tSPI_EN as the POR is released. 0 = SPI disabled; Result of Reset 1 = SPI enabled.
1 BKGDPE	BKGDPE — BKGD Pin Enable Bit When BKGD/MS is shared with general-purpose I/O through chip-level hookup, The BKGDPE bit enables the BKGD/MS pin to function as BKGD. When the BKGDPE bit is clear, the pin functions as general-purpose I/O, which must be implemented as output-only. 0 = BKGD pin disabled 1 = BKGD pin enabled; Result of Reset

10.22.3.4 SIM option 2 register (SIMOPT2)

Table 166. SIM option 2 register (SIMOPT2) (address $1803)

Bit	7	6	5	4	3	2	1	0
R	0	COPT2	COPT1	COPT0	LFOSEL	TCLKDIV	BUSCLKS1	BUSCLKS0
W	—							
Reset ($70)	0	1	1	1	0	0	0	0

Table 167. SIMOPT2 field descriptions

Field	Description
6:4 COPT[2:0]	COPT[2:0] — COP Watchdog Timeout The 3 bits COPT[2:0] write-once bits, with COPCLKS in SIMOPT1 select the timeout period of the COP. When COPCLKS = 0 = COP clocked by LFO, ~1 ms COPT[2:0] 0 0 0 = 2^5 LFO bus cycles, ~32 ms 0 0 1 = 2^6 LFO bus cycles, ~64 ms 0 1 0 = 2^7 LFO bus cycles, ~128 ms 0 1 1 = 2^8 LFO bus cycles, ~256 ms 1 0 0 = 2^9 LFO bus cycles, ~512 ms 1 0 1 = 2^{10} LFO bus cycles, ~1024 ms 1 1 0 = 2^{11} LFO bus cycles, ~2048 ms; Result of Reset When COPCLKS = 1 = COP clocked by system bus, which is itself selectable at ~2 µs, ~1 µs, ~0.5 µs, and ~0.25 µs. COPT[2:0] 0 0 0 = 2^{13} bus cycles 0 0 1 = 2^{14} bus cycles 0 1 0 = 2^{15} bus cycles 0 1 1 = 2^{16} bus cycles 1 0 0 = 2^{17} bus cycles 1 0 1 = 2^{18} bus cycles 1 1 0 = 2^{19} bus cycles 1 1 1 = 2^{19} bus cycles; Result of Reset
3 LFOSEL	LFOSEL — LFO Selected The LFOSEL selects the TPM channel 0 input to be connected to LFO or to the PTA2 0 = TPM channel 0 input connected to the PTA2; Result of Reset 1 = TPM channel 0 input connected to LFO
2 TCLKDIV	TCLKDIV — TPM External Clock Divider Enable Enables divider by 8 of DX clock from RF module 0 = TPM External Clock source is the DX clock; Result of Reset 1 = TPM External Clock source is the DX clock divided by 8.
1:0 BUSCLKS[1:0]	BUSCLKS[1:0] — BUS Clock Divider Selection The 2 bits BUSCLKS[1:0] select the internal divisions to result in the desired system bus clock: 0 0 = 16 MHz / 2 = Core of 8 MHz and Bus of 4 MHz; Result of Reset 0 1 = 16 MHz / 4 = Core of 4 MHz and Bus of 2 MHz 1 0 = 16 MHz / 8 = Core of 2 MHz and Bus of 1 MHz 1 1 = 16 MHz / 16 = Core of 1 MHz and Bus of 0.5 MHz

10.22.3.5 SIM part ID high and low byte registers (SIMPID1/SIMPID2)

Table 168. SIM part ID high register (SIMPID1) (address $1806)

Bit	7	6	5	4	3	2	1	0
R W	REV3	REV2	REV1	REV0	ID11	ID10	ID9	ID8
Reset ($00)	0	0	0	0	0	0	0	0

Table 169. SIM part ID low register (SIMPID2) (address $1807)

Bit	7	6	5	4	3	2	1	0
R W	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
Reset	0	0	1	0	1	1	0	0

Table 170. SIMPID1/SIMPID2 register field descriptions

Field	Description
7:4 REV[3:0]	REV[3:0] – Revision ID The 4 bits REV[3:0] identifies the revision number of the design. Each minor change, i.e. metal layers, up to and including all layer changes will increment the revision ID: 0 0 0 0 = First silicon design 0 0 0 1 through 1 1 1 1 = each subsequent design change
3:0, 7:0 ID[11:0]	ID[11:0] – ID for the HC9S08SZK16 CPU = 0 0 0 0 0 0 1 0 1 1 0 0 = $02C.

10.22.3.6 SIM stop exit status register (SIMSES)

The SIMSES register indicates the cause of the last stop exit. All bits in the SIMSES register are read only and are cleared prior to the STOP1 entry.

Table 171. SIM stop exit status register (SIMSES) (address $180D)

Bit	7	6	5	4	3	2	1	0
R	0	0	KBF	IRQF	FRCF	PWUF	LFF	RFF
W	—							
Reset	0	0	U	U	U	U	U	U
POR ($00)	0	0	0	0	0	0	0	0

Table 172. SIMSES register field descriptions

Field	Description
5 KBF	KBF — Keyboard Flag. The KBF flag bit indicates that any keyboard pin caused the last stop exit. 0 = Keyboard pin did not cause last stop exit; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = Keyboard pin caused last stop exit.
4 IRQF	IRQF — IRQ Flag. The IRQF flag bit indicates that IRQ pin caused last stop exit. 0 = IRQ pin did not cause last stop exit; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = IRQ pin caused last stop exit.
3 FRCF	FRCF — FRC Flag. The FRCF flag bit indicates that FRC Interrupt caused last stop exit. 0 = FRC interrupt did not cause last stop exit; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = FRC interrupt caused last stop exit.
2 PWUF	PWUF — Periodic Wake-up Unit Flag. The PWUF flag bit indicates that PWU module caused last stop exit. 0 = PWU did not cause last stop exit; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = PWU caused last stop exit.
1 LFF	LFF — Low Frequency Receiver Flag. The LFF flag bit indicates that LF module caused last stop exit. 0 = LF module did not cause last stop exit; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = LF module caused last stop exit.
0 RFF	RFF — RF Flag. The RFF flag bit indicates that RF module caused last stop exit. 0 = RF module did not cause last stop exit; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = RF module caused last stop exit.

10.22.3.7 SIM oscillator trim register (SIMOTRM)

Table 173. SIM oscillator trim register (SIMOTRM) (address $180E)

Bit	7	6	5	4	3	2	1	0
R **W**	SOTRM7	SOTRM6	SOTRM5	SOTRM4	SOTRM3	SOTRM2	SOTRM1	SOTRM0
Reset	from IFR	from IFR	from IFR	from IFR	from IFR	from IFR	from IFR	from IFR

Table 174. SIMOTRM register field descriptions

Field	Description
7:0 SIMOTRM[7:0]	SIMOTRM[7:0] — SIM Oscillator Trim Output of the SIMOTRM[7:0] is directed to ICS to control the temporary setting of the MFO frequency. The register is loaded after any reset with Flash IFR[31:24] bit contents. This register is writable at any time. SIMOTRM[7:0] holds the Trim value for the Medium Frequency Oscillator (MFO).

10.23 Power management controller (PMC) module

A low-power CMOS on-chip fixed voltage regulator provides internal power to the MCU from an external DC source. The nominal output voltage remains steady over the (LV) 1.8 V to 3.6 V input voltage range.

Features of the PMC are described as follows:

- Separate digital (regulated) and analog (referenced to digital) supply outputs.
- No output supply decoupling capacitors required.
- Programmable power saving states
- Real Time Interrupt (RTI) with programmable period using internal oscillator or external clock.
- Automatic wake-up from power saving states by real-time interrupt
- Available wake-up from power saving states via external input.
- Integrated Power-on Reset (POR).
- Integrated Low Voltage Detect (LVD) with reset (brownout) or interrupt capability.
- Programmable LVD trip points.
- Programmable Low Voltage Warning (LVW) indicator.
- Voltage output indicator.
- Buffered band gap reference voltage output
- Factory programmed trim for band gap reference for regulator and LVD.
- No integrated short circuit or thermal protection.
- RTI clock and counter output signals for use by other modules on-chip.
- Optional High Voltage Warning (HVW) with interrupt for over-voltage detection.
- Free-running counter (0.5 ms counter interval nominally) that can operate in all run and stop modes (start, hold, and reset operations)

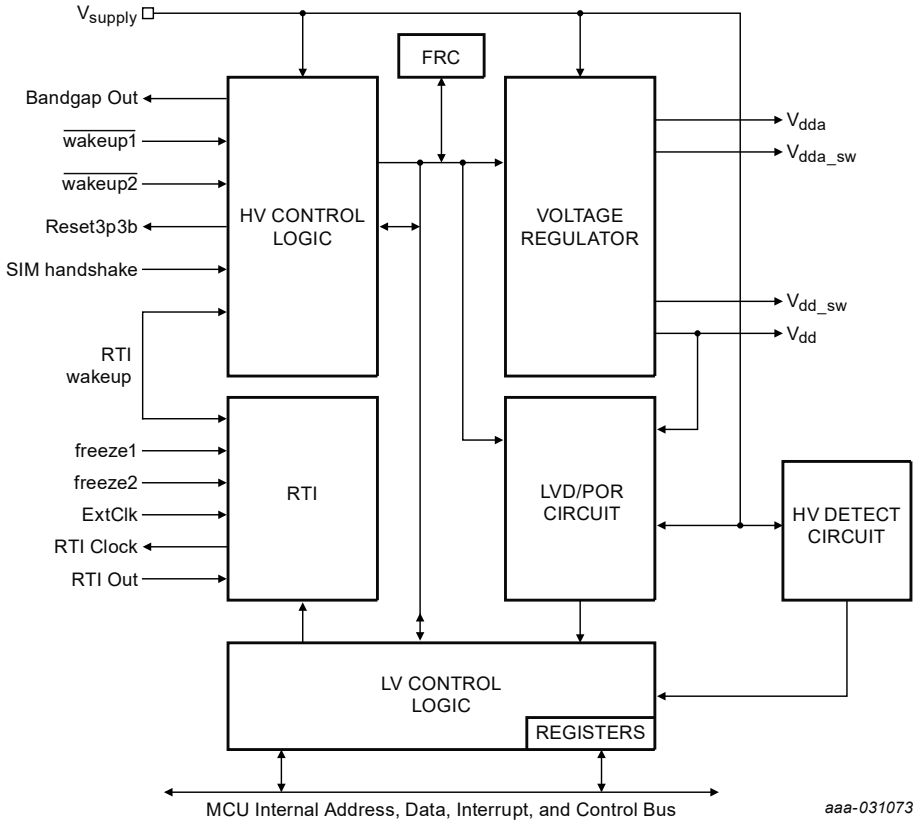


Figure 52. Power management controller module block diagram

10.23.1 PMC state transitions

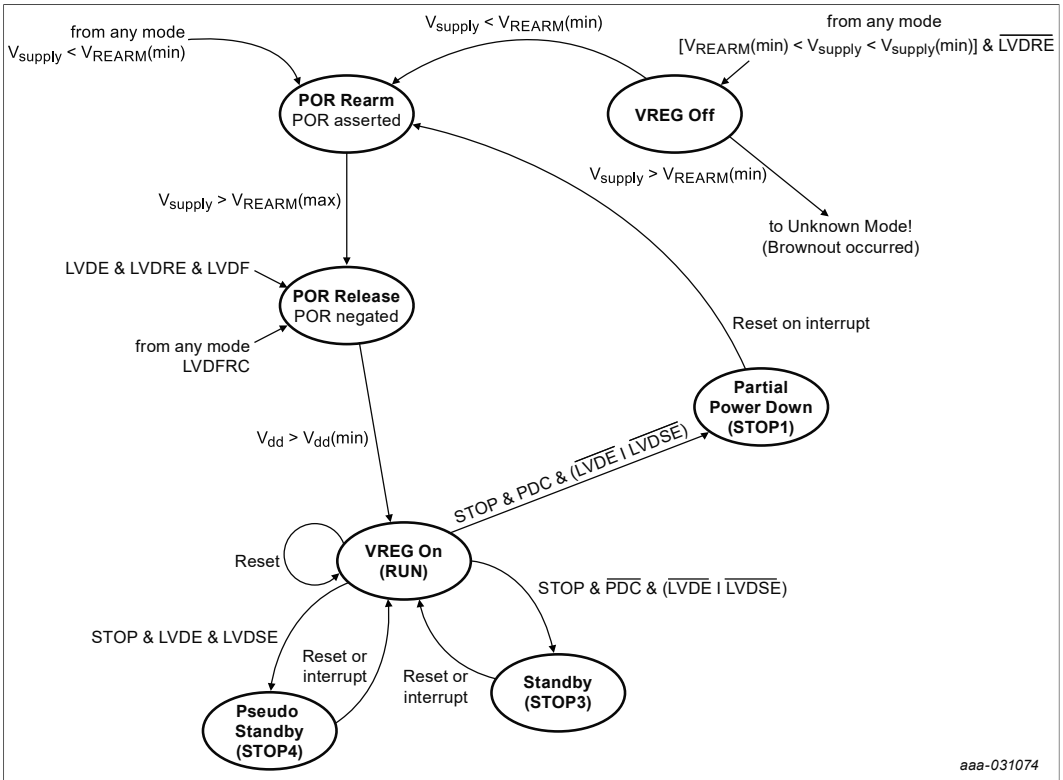


Figure 53. Power management controller state transition diagram

10.23.2 PMC low voltage detection transitions

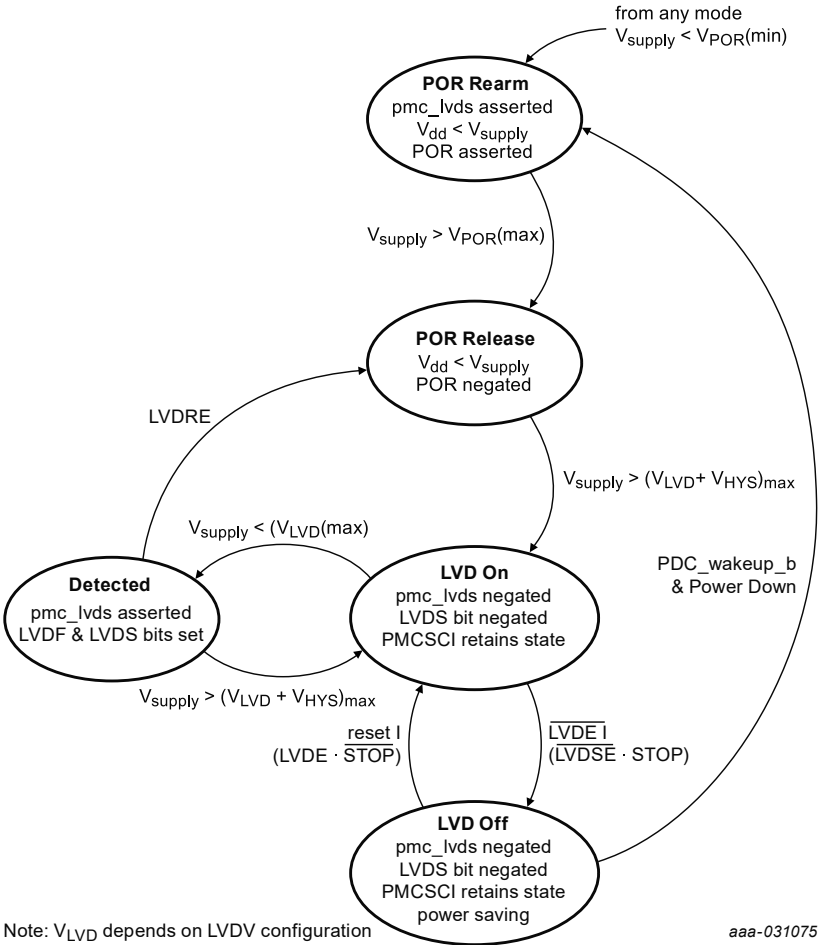


Figure 54. Power management controller low voltage detection transitions state diagram

10.23.3 PMC register descriptions

10.23.3.1 PMC real-time-interrupt status and control register (SRTISC)

The SRTISC register contains the status and control bits associated with the PMC real-time interrupt function. Note: Exit from POR or STOP1 will reset all SRTISC register bits to default $00. The interrupt from RTI operates from all power modes, however the RTIF flag will not be set and the interrupt service routine will not execute if the RTI is configured and STOP1 mode entered. RTIF flag and the interrupt service routine execute if in Run mode or if STOP4 is entered. This means if the RTI is initialized and then the user application enters STOP1, the exit of STOP1 due to RTI will have to be derived by means other than query of the RTIF flag. An example might be user application setting of a bit in a PARAM register prior to initializing the RTI, then entering STOP1. Upon the wake-up, query the PARAM register bit and if set, indicates that RTI had been the cause of the wake-up.

Table 175. PMC real-time-interrupt status and control register (SRTISC) (address $1808)

Bit	7	6	5	4	3	2	1	0
R	RTIF	0	RTICLKS	RTIE	0	RTIS2	RTIS1	RTIS0
W	—	RTIACK			—			
Reset	U	0	0	0	0	0	0	0
POR ($00)	0	0	0	0	0	0	0	0

Table 176. SRTISC register field descriptions

Field	Description
7 RTIF	RTIF – Real-Time Interrupt Flag The RTIF bit indicates the real-time interrupt request status when configured for Run or Stop4 modes. 0 = Real-time interrupt request is not pending; Result of power-on reset and exit of Stop1; Existing state retained from resets of low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and background debugger. 1 = Real-time interrupt request is pending.
6 RTIACK	RTIACK – Real-Time Interrupt Acknowledge Writing a logic 1 to RTIACK clears the real-time interrupt request and clears RTIF to a logic 0. 0 = Read result; Write no effect; Result of Reset 1 = Write 1 to clear RTIF for Real-time interrupt acknowledge.
5 RTICLKS	RTICLKS – Real-Time Interrupt Clock Select The RTICLKS bit selects the clock source for the real-time interrupt request. 0 = Real-time interrupt request clock source is internal oscillator; Result of Reset and exit of Stop1. 1 = Real-time interrupt request clock source is external clock.
4 RTIE	RTIE – Real-Time Interrupt Enable The RTIE bit enables the interrupt output of the RTI module from Run or Stop4 modes. 0 = Real-time interrupt request disabled; Result of Reset and exit of Stop1. 1 = Real-time interrupt request enabled.
2:0 RTIS[2:0]	RTIS[2:0] – The 3 bits RTIS[2:0] selects the real-time interrupt period derived from the selected real-time interrupt clock source. 0 0 0 = Off; Result of Reset and exit of Stop1. 0 0 1 = 2 ms 0 1 0 = 4 ms 0 1 1 = 8 ms 1 0 0 = 16 ms 1 0 1 = 32 ms 1 1 0 = 64 ms 1 1 1 = 128 ms

10.23.3.2 PMC status and control 1 register (SPMSC1)

PMCSC1 contains the status and control bits associated with the PMC Voltage Regulator (VREG) and Low Voltage Detect (LVD) functions.

Table 177. PMC status and control 1 register (SPMSC1) (address $1809)

Bit	7	6	5	4	3	2	1	0
R	LVDF	0	LVDIE	LVDRE	LVDSE	LVDE	—	BGBE
W	—	LVDACK						
Reset	U	0	0	1	1	1	0	0
POR ($1C)	0	0	0	1	1	1	0	0

Table 178. SPMSC1 register field descriptions

Field	Description
7 LVDF	LVDF – Low Voltage Detection Flag The LVDF bit indicates the Low Voltage Detect status if LVDE is set. 0 = Low voltage has not been detected; Result of power-on reset; Existing state retained from resets of Stop1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = Low voltage is being or has been detected.
6 LVDACK	LVDACK – LVD Interrupt Acknowledge Writing a logic 1 to LVDACK clears the LVD interrupt request and clears LVDF to a logic 0 if low voltage is not detected. 0 = Read result; Write no effect; Result of Reset 1 = Write 1 to clear LVDF for Low Voltage Detect acknowledge.
5 LVDIE	LVDIE – LVD Interrupt Enable The LVDIE bit controls the LVD interrupt if LVDE is set. This bit has no effect if the LVDE bit is a logic 0. 0 = LVD interrupt disabled; Result of Reset 1 = LVD interrupt enabled
4 LVDRE	LVDRE – LVD Reset Enable The LVDRE bit controls the LVD reset if LVDE is set. The LVDRE is writable only once after each exit from a system reset. This bit has no effect if the LVDE bit is a logic 0. LVD reset has priority over LVD interrupt, if both are enabled. In all test modes, the LVDRE bit value is ignored and functions as if the LVDRE bit value is equal to 0. 0 = LVD reset disabled. 1 = LVD reset enabled; Result of Reset
3 LVDSE	LVDSE – LVD Stop Enable The LVDSE bit controls the behavior of the LVD when the MCU stop mode is entered if LVDE is set. This bit has no effect if the LVDE bit is a logic 0. 0 = LVD disabled in MCU stop mode. 1 = LVD enabled in MCU stop mode; Result of Reset
2 LVDE	LVDE – LVD Enable Bit The LVDE bit controls whether the LVD is enabled. The LVDE is writable only once after each exit from a system reset. 0 = LVD is disabled. 1 = LVD is enabled; Result of Reset
0 BGBE	BGBE – Band gap Buffer Enable The BGBE bit is used to enable the band gap buffered output. 0 = Band gap buffer disabled; Result of Reset 1 = Band gap buffer enabled.

10.23.3.3 PMC status and control 2 register (SPMSC2)

SPMCSC2 register contains status and control bits associated with the PMC power down modes.

Note: *User applications not enabling STOP1, i.e. not setting PDC to 1, will result in the STOP4 mode if the application enables the STOP instruction, and subsequently executes a STOP instruction.*

Table 179. PMC status and control 2 register (SPMSC2) (address $180A)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	PDF	0	0	PDC	0
W	—	—	—	—	—	PPDACK		—
Reset	0	0	0	U	0	0	0	0
POR ($00)	0	0	0	0	0	0	0	0

Table 180. SPMSC2 register field descriptions

Field	Description
4 PDF	PDF – Power Down Recovery Flag The PDF bit indicates that the PMC has exited the STOP1 mode. PDF is cleared by writing a 1 to PPDACK. PDF defaults to 0 following an LVD reset or true POR. True POR is defined as removing and reapplying power (i.e. both V_{DD} and V_{DDA} are removed) to the MCU. 0 = Not STOP1 exit; Result of power-on reset; Existing state retained from resets of low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = STOP1 exit.
2 PPDACK	PPDACK – Partial Power Down Acknowledge control Writing a logic 1 to PPDACK clears the PDF bit. 0 = Read result; Write no effect; Result of Reset 1 = Write 1 to clear PDF for Power Down acknowledge.
1 PDC	PDC – Power Down Control The PDC bit controls entry into the STOP1 mode. 0 = Power Down states are disabled; Result of Reset 1 = STOP1 mode enabled.

10.23.3.4 PMC status and control 3 register (PMCSC3)

PMCSC3 register contains status and control bits associated with the PMC Low Voltage Warning (LVW) subsystem.

Table 181. PMC status and control 3 register (PMCSC3) (address $180C)

Bit	7	6	5	4	3	2	1	0
R	LVWF	0	LVDV	LVWV	—	0	0	—
W	—	LVWACK			—	—	—	
Reset	U	U	U	U	U	U	U	U
POR ($00)	0	0	0	0	0	0	0	0

Table 182. PMCSC3 register field descriptions

Field	Description
7 LVWF	LVWF – Low Voltage Warning Flag The LVWF bit indicates the Low Voltage Warning status if LVDE is set. 0 = Low voltage warning not present; Result of power-on reset; Existing state retained from resets of STOP1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = Low voltage warning is present or was present.
6 LVWACK	LVWACK – LVW Acknowledge Writing a logic 1 to LVWACK clears LVWF to a logic 0 if a low voltage warning is not present. 0 = LVW not acknowledged; Result of Reset 1 = LVW acknowledged.
5 LVDV	LVDV – Low Voltage Detect Voltage Select The LVDV bit selects the LVD trip point voltage (VLVD). When double trip points are selected during chip integration, the selection below (high and low) is available, otherwise a single trip point, VLVDH is selected (VLVD = VLVDH). 0 = VLVDL selected (VLVD = VLVDL); Result of power-on reset; Existing state retained from resets of STOP1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = VLVDH selected (VLVD = VLVDH).
4 LVWV	LVWV – Low Voltage Warning Voltage Select The LVWV bit selects the low voltage warning detection voltage if LVDE is set. When double trip points are selected during chip integration, the selection below (high and low) is available, otherwise a single trip point, VLVDH is selected (VLVD = VLVDH). 0 = VLVDL selected (VLVD = VLVDL); Result of power-on reset; Existing state retained from resets of STOP1 exit, low voltage detection, external pin, COP, PWU, illegal opcode, illegal address, soft reset, and back-ground debugger. 1 = VLVDH selected (VLVD = VLVDH).

10.24 Flash memory controller (FMC) module

10.24.1 Flash controller general items

The FLASH module has nine 8-bit registers in the high-page register space, and three locations in the nonvolatile register space in FLASH memory which are copied into three corresponding high-page control registers at reset. There is also an 8-byte comparison key in FLASH memory. An NXP Semiconductor-provided equate or header file normally is used to translate these names into the appropriate absolute addresses.

FLASH memory block provides:

- User Program FLASH Size — 8192 bytes (16 pages of 512 bytes each)
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 100,000 program/erase cycles at typical voltage and temperature
- Flexible block protection
- Security feature for FLASH and RAM
- Auto power-down for low-frequency read accesses

10.24.2 FMC program and erase times

Before any program or erase command can be accepted, the FLASH clock divider register (FCDIV) must be written to set the internal clock for the FLASH module to a frequency (f_{FCLK}) between 150 kHz and 200 kHz. This register can be written only once, so normally this write is performed during reset initialization. FCDIV cannot be written if the access error flag, FACCERR in FSTAT, is set. The user must ensure that FACCERR is not set before writing to the FCDIV register. One period of the resulting clock ($1/f_{FCLK}$) is used by the command processor to time program and erase pulses. An integer number of these timing pulses are used by the command processor to complete a program or erase command.

Table 183 shows program and erase times. The bus clock frequency, and FCDIV determine the frequency of FCLK (f_{FCLK}). The time for one cycle of FCLK is $t_{FCLK} = 1/f_{FCLK}$. The times are shown as a number of cycles of FCLK and as an absolute time for the case where $t_{FCLK} = 5$ us. Program and erase times shown include overhead for the command state machine and enabling and disabling of program and erase voltages.

Table 183. Program and erase times

Parameter	Cycles of FCLK	Time[1] assuming FCLK = 200 kHz
Byte program	9	45 µs
Byte program (burst)	4	20 µs
Page erase	4000	20 ms
Mass erase	20,000	100 ms

[1] Burst program time excludes start/end overhead.

10.24.3 FMC program and erase command execution

The steps for executing any of the commands are listed below. The FCDIV register must be initialized and any error flags cleared before beginning command execution. The command execution steps are:

1. Write a data value to an address in the FLASH array. The address and data information from this write is latched into the FLASH interface. This write is a required first step in any command sequence. For erase and blank check commands, the value of the data is not important. For page erase commands, the address may be any address in the 512-byte page of FLASH to be erased. For mass erase and blank check commands, the address can be any address in the FLASH memory. Whole pages of 512 bytes are the smallest block of FLASH that may be erased. Do not program any byte in the FLASH more than once after a successful erase operation. Reprogramming bits to a byte which is already programmed is not allowed without first erasing the page in which the byte resides or mass erasing the entire FLASH memory. Programming without first erasing may disturb data stored in the FLASH.
2. Write the command code for the desired command to FCMD. The five valid commands are blank check ($05), byte program ($20), burst program ($25), page erase ($40), and mass erase ($41). The command code is latched into the command buffer.
3. Write a 1 to the FCBEF bit in FSTAT to clear FCBEF and launch the command (including its address and data information).

A partial command sequence can be aborted manually by writing a 0 to FCBEF anytime after the write to the memory array and before writing the 1 that clears FCBEF and launches the complete command. Aborting a command in this way sets the FACCERR access error flag which must be cleared before starting a new command.

A strictly monitored procedure must be obeyed or the command will not be accepted. This minimizes the possibility of any unintended changes to the FLASH memory contents. The command complete flag (FCCF) indicates when a command is complete. The command sequence must be completed by clearing FCBEF to launch the command. Below is a flowchart for executing all of the commands except for burst programming. The FCDIV register must be initialized before using any FLASH commands. This must be done only once following a reset.

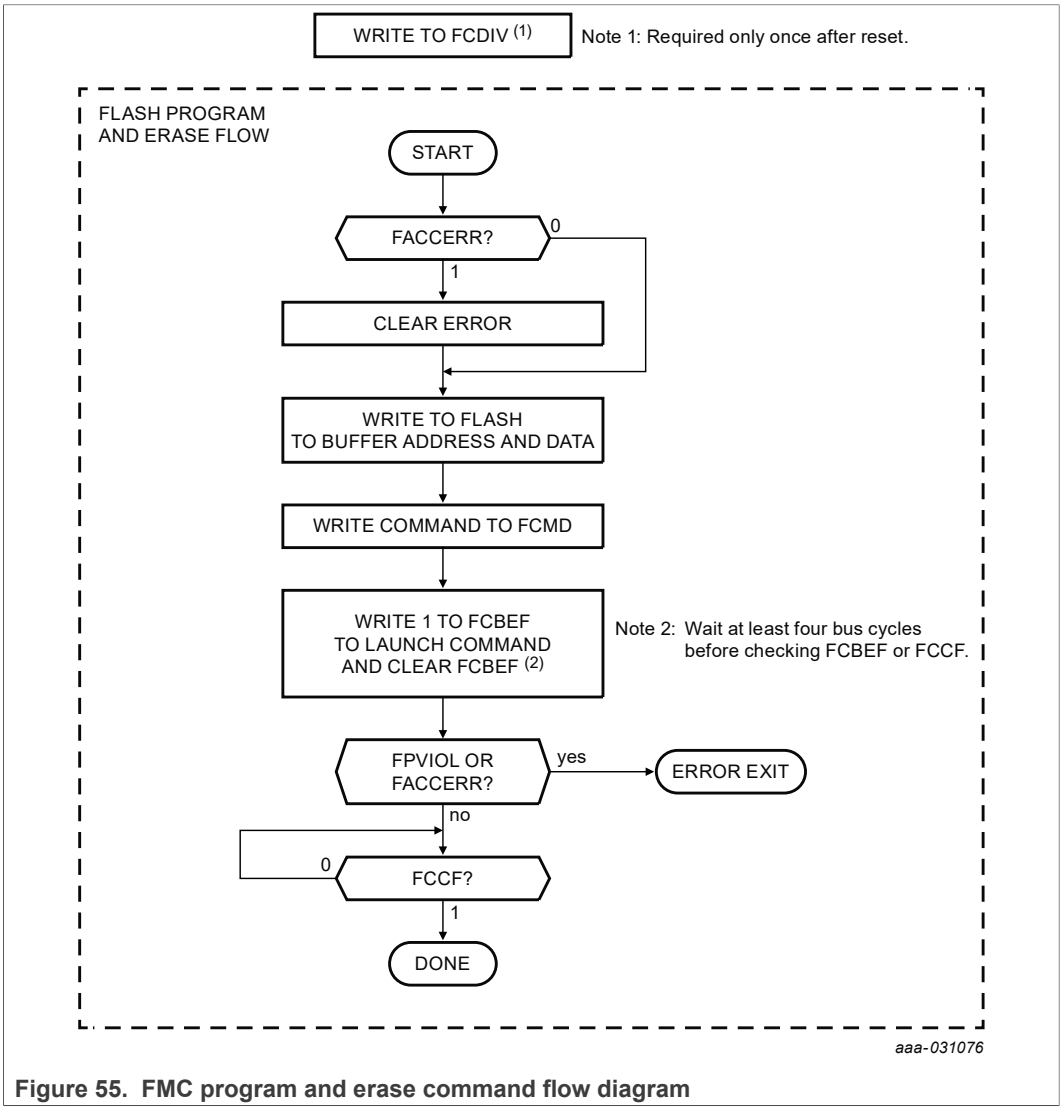


Figure 55. FMC program and erase command flow diagram

10.24.4 FMC burst program execution

The burst program command is used to program sequential bytes of data in less time than would be required using the standard program command. This is possible because the high voltage to the FLASH array does not need to be disabled between program operations. Ordinarily, when a program or erase command is issued, an internal charge

pump associated with the FLASH memory must be enabled to supply high voltage to the array. Upon completion of the command, the charge pump is turned off. When a burst program command is issued, the charge pump is enabled and then remains enabled after completion of the burst program operation if these two conditions are met:

- The next burst program command has been queued before the current program operation has completed.
- The next sequential address selects a byte on the same physical row as the current byte being programmed. A row of FLASH memory consists of 64 bytes. A byte within a row is selected by addresses A5 through A0. A new row begins when addresses A5 through A0 are all zero.

The first byte of a series of sequential bytes being programmed in burst mode will take the same amount of time to program as a byte programmed in standard mode. Subsequent bytes will program in the burst program time provided that the conditions above are met. In the case the next sequential address is the beginning of a new row, the program time for that byte will be the standard time instead of the burst time. This is because the high voltage to the array must be disabled and then enabled again. If a new burst command has not been queued before the current command completes, then the charge pump will be disabled and high voltage removed from the array.

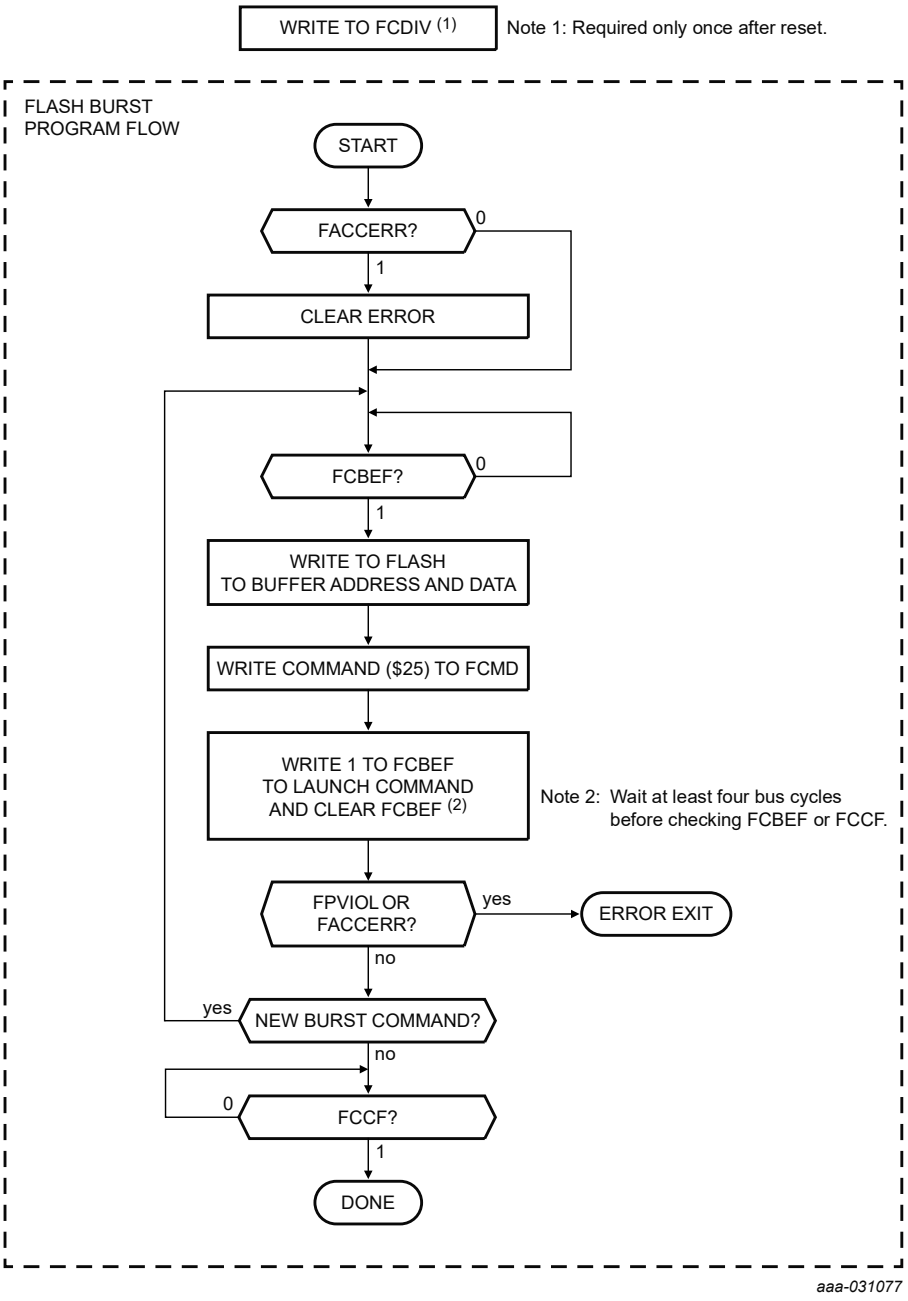


Figure 56. FMC burst program flow diagram

10.24.5 FMC memory access errors

An access error occurs whenever the command execution protocol is violated. Any of the following specific actions cause the access error flag (FACCERR) in FSTAT to be set. FACCERR must be cleared by writing a 1 to FACCERR in FSTAT before any command can be processed.

- Writing to a FLASH address before the internal FLASH clock frequency has been set by writing to the FCDIV register

- Writing to a FLASH address while FCBEF is not set (A new command cannot be started until the command buffer is empty.)
- Writing a second time to a FLASH address before launching the previous command (There is only one write to FLASH for every command.)
- Writing a second time to FCMD before launching the previous command (There is only one write to FCMD for every command.)
- Writing to any FLASH control register other than FCMD after writing to a FLASH address
- Writing any command code other than the five allowed codes ($05, $20, $25, $40, or $41) to FCMD
- Accessing (read or write) any FLASH control register other than the write to FSTAT (to clear FCBEF and launch the command) after writing the command to FCMD.
- The MCU enters STOP mode while a program or erase command is in progress (The command is aborted.)
- Writing the byte program, burst program, or page erase command code ($20, $25, or $40) with a BACKGROUNDDEBUG command while the MCU is secured (the BACKGROUND DEBUG controller can only do blank check and mass erase commands when the MCU is secure.)
- Writing 0 to FCBEF to cancel a partial command.

10.24.6 FMC block protection

The block protection feature prevents the protected region of FLASH from program or erase changes. Block protection is controlled through the FLASH Protection Register (FPROT). When enabled, block protection begins at any 512-byte boundary below the last address of FLASH, $FFFF.

After exit from reset, FPROT is loaded with the contents of the NVPROT location which is in the nonvolatile register block of the FLASH memory. FPROT cannot be changed directly from application software so a runaway program cannot alter the block protection settings. Because NVPROT is within the last 512 bytes of FLASH, if any amount of memory is protected, NVPROT is itself protected and cannot be altered (intentionally or unintentionally) by the application software. FPROT can be written through BACKGROUND DEBUG commands which allows a way to erase and reprogram a protected FLASH memory.

The block protection mechanism is illustrated below. The FPS bits are used as the upper bits of the last address of unprotected memory. This address is formed by concatenating FPS7:FPS1 with logic 1 bits as shown. For example, in order to protect the last 8192 bytes of memory (addresses $E000 through $FFFF), the FPS bits must be set to 1101 111 which results in the value $DFFF as the last address of unprotected memory. In addition to programming the FPS bits to the appropriate value, FPDIS (bit 0 of NVPROT) must be programmed to logic 0 to enable block protection. Therefore the value $DE must be programmed into NVPROT to protect addresses $E000 through $FFFF.

One use for block protection is to block protect an area of FLASH memory for a boot loader program. This boot loader program then can be used to erase the rest of the FLASH memory and reprogram it. Because the boot loader is protected, it remains intact even if MCU power is lost in the middle of an erase and reprogram operation.

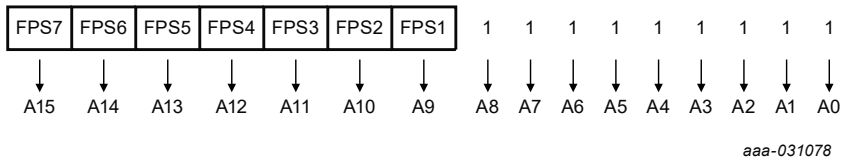


Figure 57. Flash block protection protocol

10.24.7 FMC vector redirection

Vector redirection is not recommended for TPMS applications where NXP firmware has been included in the final image.

Whenever any block protection is enabled, the reset and interrupt vectors will be protected. Vector redirection allows users to modify interrupt vector information without unprotecting boot loader and reset vector space. Vector redirection is enabled by programming the FNORED bit in the NVOPT register located at address $FFBF to zero. For redirection to occur, at least some portion but not all of the FLASH memory must be block protected by programming the NVPROT register located at address $FFBD. All of the interrupt vectors (memory locations $FFC0–$FFFD) are redirected, though the reset vector ($FFFE:FFFF) is not.

For example, if 512 bytes of FLASH are protected, the protected address region is from $FE00 through $FFFF. The interrupt vectors ($FFC0–$FFFD) are redirected to the locations $FDC0–$FDFD. If an SPI interrupt is taken for instance, the values in the locations $FDE0:FDE1 are used for the vector instead of the values in the locations $FFE0:FFE1. This allows the user to reprogram the unprotected portion of the FLASH with new program code including new interrupt vector values while leaving the protected area, including the default vector locations, unchanged.

10.24.8 FMC security

Circuitry is included to prevent unauthorized access to the contents of FLASH and RAM memory. When security is engaged, FLASH and RAM are considered secure resources. Direct-page registers, high-page registers, and the BACKGROUND DEBUG controller are considered unsecured resources. Programs executing within secure memory have normal access to any MCU memory locations and resources. Attempts to access a secure memory location with a program executing from an unsecured memory space or through the BACKGROUND DEBUG interface are blocked (writes are ignored and reads return all 0s).

Security is engaged or disengaged based on the state of two nonvolatile register bits (SEC0[1:0]) in the FOPT register. During reset, the contents of the nonvolatile location NVOPT are copied from FLASH into the working FOPT register in high-page register space. A user engages security by programming the NVOPT location, which can be done at the same time the FLASH memory is programmed. The 1:0 state disengages security and the other three combinations engage security. Notice the erased state (1:1) makes the MCU secure. During development, whenever the FLASH is erased, it is good practice to immediately program the SEC00 bit to 0 in NVOPT so SEC[1:0] = 1:0. This would allow the MCU to remain unsecured after a subsequent reset.

The on-chip debug module cannot be enabled while the MCU is secure. The separate BACKGROUND DEBUG controller can still be used for background memory access commands, but the MCU cannot enter ACTIVE BACKGROUND mode except by holding BKGD/MS low at the rising edge of reset.

A user can choose to allow or disallow a security unlocking mechanism through an 8-byte backdoor security key. If the nonvolatile KEYEN bit in NVOPT/FOPT is 0, the backdoor key is disabled and there is no way to disengage security without completely erasing all FLASH locations. If KEYEN is 1, a secure user program can temporarily disengage security by:

1. Writing 1 to KEYACC in the FCNFG register. This makes the FLASH module interpret writes to the backdoor comparison key locations (NVBACKKEY through NVBACKKEY +7) as values to be compared against the key rather than as the first step in a FLASH program or erase command.
2. Writing the user-entered key values to the NVBACKKEY through NVBACKKEY+7 locations. These writes must be done in order starting with the value for NVBACKKEY and ending with NVBACKKEY+7. STHX must not be used for these writes because these writes cannot be done on adjacent bus cycles. User software normally would get the key codes from outside the MCU system through a communication interface such as a serial I/O.
3. Writing 0 to KEYACC in the FCNFG register. If the 8-byte key that was just written matches the key stored in the FLASH locations, SEC[1:0] are automatically changed to 1:0 and security will be disengaged until the next reset.

The security key can be written only from secure memory (either RAM or FLASH), so it cannot be entered through BACKGROUND commands without the cooperation of a secure user program.

The backdoor comparison key (NVBACKKEY through NVBACKKEY+7) is located in FLASH memory locations in the nonvolatile register space so users can program these locations exactly as they would program any other FLASH memory location. The nonvolatile registers are in the same 512-byte block of FLASH as the reset and interrupt vectors, so block protecting that space also block protects the backdoor comparison key. Block protects cannot be changed from user application programs, so if the vector space is block protected, the backdoor security key mechanism cannot permanently change the block protect, security settings, or the backdoor key.

Security can always be disengaged through the BACKGROUND DEBUG interface by taking these steps:

1. Disable any block protections by writing FPROT. FPROT can be written only with BACKGROUND DEBUG commands, not from application software.
2. Mass erase FLASH if necessary.
3. Blank check FLASH. Provided FLASH is completely erased, security is disengaged until the next reset.

To avoid returning to secure mode after the next reset, program NVOPT so SEC[1:0] = 1:0.

Enabling the security feature disables NXP ability to perform failure analysis without first completely erasing all flash memory contents. If the security feature is implemented, the customer is responsible for providing NXP with unsecured parts for any failure analysis to begin or supplying the entire contents of the device flash memory data as part of the return process, to allow NXP to erase and subsequently restore the device to its original condition.

10.24.9 FMC register descriptions

10.24.9.1 FMC clock divider register (FCDIV)

Table 184. FMC clock divider register (FCDIV) (address $1820)

Bit	7	6	5	4	3	2	1	0
R	DIVLD	PRDIV8	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
W	—							
Reset ($00)	0	0	0	0	0	0	0	0

Table 185. FCDIV register field descriptions

Field	Description
7 DIVLD	DIVLD – Divisor Loaded Status Flag When set, this read-only in user mode status flag indicates that the FCDIV register has been written since reset. Reset clears this bit and the first write to this register causes this bit to become set regardless of the data written. This bit can be written in flash test mode while the FDONE bit remains = 1. 0 = FCDIV has not been written since reset; erase and program operations disabled for FLASH; Result of Reset 1 = FCDIV has been written since reset; erase and program operations enabled for FLASH
6 PRDIV8	PRDIV8 – Write-once in user mode, Prescale (Divide) FLASH Clock by 8. This bit can be written or read in flash test mode while the FDONE bit remains = 1. 0 = Clock input to the FLASH clock divider is the bus rate clock; Result of Reset 1 = Clock input to the FLASH clock divider is the bus rate clock divided by 8
5 DIV[5:0]	DIV[5:0] – Divisor for FLASH Clock Divider The FLASH clock divider divides the bus rate clock (or the bus rate clock divided by 8 if PRDIV8 = 1) by the value in the 6-bit DIV5:DIV0 field plus one. The resulting frequency of the internal FLASH clock must fall within the range of 200 kHz to 150 kHz for proper FLASH operations. Program/Erase timing pulses are one cycle of this internal FLASH clock which corresponds to a range of 5 µs to 6.7 µs. The automated programming logic uses an integer number of these pulses to complete an erase or program operation. • if PRDIV8 = 0 — f_{FCLK} = fBUS / (DIV[5:0] + 1) • if PRDIV8 = 1 — f_{FCLK} = fBUS / (8 / (DIV[5:0] + 1)) 0 0 0 0 0 0 = Result of Reset. These bits can be written or read in flash test mode while the FDONE bit remains = 1.

Table 186. FMC clock divider register settings

f_{Bus} (MHz)	PRDIV8 (Binary)	DIV5:0 (Decimal)	f_{FLCK} (kHz)	Program/Erase Timing Pulse (5 µs Min, 6.7 µs Max) (µs)
20	1	12	192.3	5.2
10	0	49	200	5
8	0	39	200	5
4	0	19	200	5
2	0	9	200	5
1	0	4	200	5

Table 186. FMC clock divider register settings*...continued*

f_{Bus} (MHz)	PRDIV8 (Binary)	DIV5:0 (Decimal)	f_{FLCK} (kHz)	Program/Erase Timing Pulse (5 µs Min, 6.7 µs Max) (µs)
200 kHz	0	0	200	5
150 kHz	0	0	150	6.7

10.24.9.2 FMC option registers (FOPT and NVOPT)

During reset, the contents of the nonvolatile location NVOPT are copied from FLASH into FOPT. Bits 5 through 2 are not used and always read 0. This register may be read at any time, but writes have no meaning or effect. To change the value in this register, erase and reprogram the NVOPT location in FLASH memory as usual and then issue a new MCU reset.

Table 187. FMC option registers (FOPT) (address $1821)

Bit	7	6	5	4	3	2	1	0
R	KEYEN	FNORED	0	0	0	0	SEC1	SEC0
W	—	—	—	—	—	—	—	—
Reset ($)			0	0	0	0		

Table 188. FMC option registers (NVOPT) (address $FFBF)

Bit	7	6	5	4	3	2	1	0
R	KEYEN	FNORED	0	0	0	0	SEC1	SEC0
W	—	—	—	—	—	—	—	—
Reset			0	0	0	0	1 from factory 1 erased	0 from factory 1 erased

Table 189. FOPT and NVOPT register field descriptions

Field	Description
7 KEYEN	KEYEN – Backdoor Key Mechanism Enable When the read-only KEYEN bit is 0, the backdoor key mechanism cannot be used to disengage security. The backdoor key mechanism is accessible only from user (secured) firmware. BDM commands cannot be used to write key comparison values that would unlock the backdoor key. 0 = No backdoor key access allowed 1 = If user firmware writes an 8-byte value that matches the nonvolatile backdoor key (NVBACKKEY through NVBACKKEY+7 in that order), security is temporarily disengaged until the next MCU reset.
6 FNORED	FNORED – Vector Redirection Disable When the read-only FNORED bit is 1, then vector redirection is disabled. 0 = Vector redirection enabled 1 = Vector redirection disabled

Table 189. FOPT and NVOPT register field descriptions*...continued*

Field	Description
1:0 SEC[1:0]	SEC[1:0] – Security State Code The two bits SEC[1:0] select the security state of the MCU as shown below. When the MCU is secure, the contents of RAM and FLASH memory cannot be accessed by instructions from any unsecured source including the BACKGROUND DEBUG interface. SEC[1:0] changes to 1 0 after successful backdoor key entry or a successful blank check of FLASH. When secured, the SPI may access only the address ranges $0000 to $008F and $1800 to $188F. Other access attempts will result in an error status. 0 0 = secured 0 1 = secured 1 0 = unsecured (default from NXP factory) 1 1 = secured (default after erase of the block $FE00 - $FFFF)

10.24.9.3 FMC configuration registers (FCNFG)

Table 190. FMC configuration registers (FCNFG) (address $1823)

Bit	7	6	5	4	3	2	1	0
R	0	0	KEYACC	0	0	0	0	0
W	—	—		—	—	—	—	—
Reset ($00)	0	0	0	0	0	0	0	0

Table 191. FCNFG register field descriptions

Field	Description
5 KEYACC	KEYACC - This bit enables writing of the backdoor comparison key. 0 Writes to $FFB0–$FFB7 are interpreted as the start of a FLASH programming or erase command; Result of Reset 1 Writes to NVBACKKEY ($FFB0–$FFB7) are interpreted as comparison key writes.

10.24.9.4 Flash protection registers (FPROT and NVPROT)

During reset, the contents of the nonvolatile location NVPROT is copied from FLASH into FPROT. Bits 0, 1, and 2 are not used and each always reads as 0. This register can be read at any time, but user program writes have no meaning or effect. BACKGROUND DEBUG commands can write to FPROT.

Table 192. Flash protection register (FPROT) (address $1824)

Bit	7	6	5	4	3	2	1	0
R	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
W								
All resets	from $FFBD	from $FFBD	from $FFBD	from $FFBD	from $FFBD	from $FFBD	from $FFBD	from $FFBD

Table 193. Flash protection register (NVPROT) (address $FFBD)

Bit	7	6	5	4	3	2	1	0
R	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
W								
All resets	from BDM	from BDM	from BDM	from BDM	from BDM	from BDM	from BDM	from BDM

Table 194. FPROT and NVPROT register field descriptions

Field	Description
7:1 FPS[7:1]	FPS[7:1] – FLASH Protect Select Bits When FPDIS = 0, this 7-bit field determines the ending address of unprotected FLASH locations at the high address end of the FLASH. Protected FLASH locations cannot be erased or programmed.
0 FPDIS	FPDIS – FLASH Protection Disable 0 = FLASH block specified by FPS[7:1] is block protected (program and erase not allowed) 1 = No FLASH block is protected

10.24.9.5 FMC status register (FSTAT)

Table 195. FMC status register (FSTAT) (address $1825)

Bit	7	6	5	4	3	2	1	0
R	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	0	0
W		—			—	—	—	—
Reset ($C0)	1		0					

Table 196. FMC status register (FSTAT) (address $1825)

Bit	7	6	5	4	3	2	1	0
R	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	FFAIL	FDONE
W		—			—			—
Reset ($C1)	1	1	0	0	0	0	0	1

Table 197. FSTAT register field descriptions

Field	Description
7 FCBEF	FCBEF – FLASH Command Buffer Empty Flag The FCBEF bit is used to launch commands. It also indicates that the command buffer is empty so that a new command sequence can be executed when performing burst programming. The FCBEF bit is cleared by writing a one to it or when a burst program command is transferred to the array for programming. Only burst program commands can be buffered. 0 = Command buffer is full (not ready for additional commands) 1 = A new burst program command can be written to the command buffer; Result of Reset

Table 197. FSTAT register field descriptions*...continued*

Field	Description
6 FCCF	FCCF – FLASH Command Complete Flag The user mode read-only FCCF is set automatically when the command buffer is empty and no command is being processed. FCCF is cleared automatically when a new command is started (by writing 1 to FCBEF to register a command). Writing to FCCF has no meaning or effect. 0 = Command in progress 1 = All commands complete; Result of Reset
5 FPVIOL	FPVIOL – Protection Violation Flag FPVIOL is set automatically when FCBEF is cleared to register a command that attempts to erase or program a location in a protected block (the erroneous command is ignored). FPVIOL is cleared by writing a 1 to FPVIOL. 0 = No protection violation; Result of Reset 1 = An attempt was made to erase or program a protected location
4 FACCERR	FACCERR – Access Error Flag FACCERR is set automatically when the proper command sequence is not obeyed exactly (the erroneous command is ignored), if a program or erase operation is attempted before the FCDIV register has been initialized, or if the MCU enters STOP while a command was in progress. FACCERR is cleared by writing a 1 to FACCERR. Writing a 0 to FACCERR has no meaning or effect. 0 = No access error; Result of Reset 1 = An access error has occurred
2 FBLANK	FBLANK – FLASH Verified as All Blank (erased) Flag The user mode read-only FBLANK is set automatically at the conclusion of a blank check command if the entire FLASH array was verified to be erased. FBLANK is cleared by clearing FCBEF to write a new valid command. Writing to FBLANK has no meaning or effect. 0 = After a blank check command is completed and FCCF = 1, FBLANK = 0 indicates the FLASH array is not completely erased; Result of Reset 1 = After a blank check command is completed and FCCF = 1, FBLANK = 1 indicates the FLASH array is completely erased (all $FF). This bit can be written in flash test mode.
1 FFAIL	FFAIL – TBA 0 = TBA; Result of Reset 1 = TBA This bit can be written in flash test mode.
0 FDONE	FDONE – TBA 0 = TBA; Result of Reset in user mode 1 = TBA; Result of Reset in test mode This bit can be read in flash test mode

10.24.9.6 FMC command register (FCMD)

Table 198. FMC command register (FCMD) (address $1826)

Bit	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	—	FCMDB6	FCMDB5	DCMDB4	FCMDB3	FCMDB2	FCMDB1	FCMDB0
Reset ($00)	0	0	0	0	0	0	0	0

Table 199. FCMD register field descriptions

Field	Description
6:0 FCMDB[6:0]	FCMDB[6:0] – Flash Commands Available as follows. All other command codes are illegal and generate an access error. It is not necessary to perform a blank check command after a mass erase operation. Only blank check is required as part of the security unlocking mechanism. See Table 200. 0 0 0 0 0 0 0 = Result of Reset

Table 200. FCMD available flash commands

Command	FCMD[6:0]	Equate File Label
Blank check	0 0 0 0 1 0 1	mBlank
Byte program	0 1 0 0 0 0 0	mByteProg
Byte program - burst mode	0 1 0 0 1 0 1	mBurstProg
Page erase (512 bytes/page)	1 0 0 0 0 0 0	mPageErase
Mass erase (all pages)	1 0 0 0 0 0 1	mMassErase

10.25 Free running counter (FRC) module

The Free Running Counter (FRC) module includes an interrupt, triggered when a programmed target value has been reached (or exceeded).

Once the FRC has been enabled, the FRC will run continuously and will roll over when not halted. The user application may benefit from being able to program a predicted future value into the compare register, then executing other processes until interrupted. The interrupt event results from the value of the FRC_TIME[15:0] becoming equal to the FRC_COMP[15:0] compare register. An interrupt vector is allocated to the FRC comparison event.

FRC_COMP_EN will enable or disable the target comparison interrupt function. Writing 1 enables the comparison interrupt, and writing 0 disables the comparison interrupt. Reading FRC_COMP_EN returns its present state. As the FRC may be programmed to operate during all modes, the FRC comparison function and interrupt operate in all modes when enabled. Writing 1 to FRC_COMP_IACK clears a new FRC_IF. Writing 0 to the FRC_COMP_IACK has no effect.

Reading the FRC_COMP_IACK bit has no effect. If FRC_IF is 1, this indicates the comparison function, or that the FRC_COMP_IACK has been written to 1.

Due to the FRC being powered by the 'always on' internal supply, the interrupt function, when enabled, may generate continuous interrupts when the FRC timer registers are equal or greater than the FRC compare registers. This condition could exist, for example, if the FRC timer has rolled-over and the FRC compare registers are remaining at a lower value. To prevent unwanted subsequent interrupts, users must immediately disable the interrupt function.

To set up the next interrupt interval, users should:

- Clear and halt the FRC timer by simultaneously writing FRC_EN_HALT=0 and FRC_CLR=1 in a single write.
- Poll the FRC_EN_HALT bit until the transition from 1 to 0
- Write the next target value into the FRC COMP registers
- Clear the interrupt status by writing 1 to the FRC_COMP_IACK bit

- Enable the FRC timer by writing 1 to the FRC_EN_HALT bit
- Poll the FRCTIMERL least significant bit until a transition from 1 to 0
- Enable the interrupt by writing 1 to the FRC_COMP_EN bit

10.25.1 Clearing or halting the FRC

Clearing or halting of the FRC, for example programming FRC_CLR to 1 or FRC_EN_HALT to 0, simultaneously disables the comparison interrupt function, i.e. force FRC_COMP_EN to 0. Users are to be aware that halting or restarting the FRC from a default value disturbs the real time until the comparison condition occurs, and therefore requires a new compare value to be calculated and written, and a new enable of the comparison interrupt function.

The FRC_EN_HALT and FRC_CLR bits are operated in a mutually exclusive manner, such that halting the FRC will not occur while the FRC_CLR bit is 1, and conversely clearing the FRC will not be attempted while the FRC_EN_HALT is 0.

When the FRC is enabled and running, the user application may clear the count value of FRCTIMERH/L to $0000 by writing 1 to the FRC_CLR bit. The FRC_CLR bit should not be written to 1 while the FRC_EN_HALT is 0.

Note that an expected result of clearing the FRC, followed by reading the FRCTIMERH/L may not return exactly $0000 due to the asynchronous nature of the LFO as clock.

The lower address of the FRC compare register holds the new FRC_COMP[15:8] bits. Then next address holds the FRC_COMP[7:0] bits. The new control and status bits FRC_COMP_EN, FRC_COMP_IACK, and FRC_IF reside in the same register as the existing FRC control bits.

Reset disables the comparison function, i.e. force FRC_COMP_EN, FRC_COMP_IACK, and FRC_IF each to 0, and forces FRC_COMP[15:0] to default value of $FFFF in order to minimize the probability of an instantaneous interrupt upon enabling.

Operation of the comparison interrupt function follows the operation of the FRC in test mode, i.e. if clocked by HFO (default 64 × MFO) in test mode, the comparison interrupt event will be faster than if clocked by the LFO in user mode.

10.25.2 Free running counter register descriptions

10.25.2.1 FRC status and control register (FRCCR)

Table 201. FRC status and control register (FRCCR) (address $1880)

Bit	7	6	5	4	3	2	1	0
R	0	0	FRC_ EN_HALT	FRC_ COMP_EN	0	FRC_IF	0	0
W	FRC_CLR	—			FRC_ COMP_ IACK	—	—	—
Reset	0	0	U	0	0	U	0	0
POR ($00)	0	0	0	0	0	0	0	0

Table 202. FRCCR register field descriptions

Field	Description
7 FRC_CLR	FRC_CLR – Free Running Counter Clear The write-only FRC_CLR bit is used to clear the free running counter. 0 = No effect; Result of Reset 1 = clear the counter to $0000
5 FRC_ EN_HALT	FRC_EN_HALT — Free Running Counter Enable Bit This bit reads the FRC_EN_HOLD signal from FRC analog. 0 = disable and halt the counter; Result of power-on reset. Existing state remains under all other types of reset. 1 = enable and release the counter
4 FRC_ COMP_EN	FRC_COMP_EN – Enable of FRC comparison When set to 1, the bit FRC_COMPEN will enable the FRC interrupt function. 0 = comparison interrupt disabled; Result of Reset 1 = comparison interrupt enabled
3 FRC_ COMP_IACK	FRC_COMP_IACK – FRC Comparison Interrupt Acknowledge The write-only FRC_COMP_IACK bit is used to clear the free running counter interrupt. 0 = no effect; Result of Reset1 = clear the FRC interrupt
2 FRC_IF	FRC_IF – FRC Interrupt Flag The FRC_IF bit indicates that the FRC interrupt is pending 0 = no FRC interrupt is pending; Result of Reset 1 = FRC interrupt is pending.

10.25.2.2 FRC timer high and low registers (FRCTIMERH/L)

Table 203. FRC timer high register (FRCTIMERH) (address $1881)

Bit	15	14	13	12	11	10	9	8
R W	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
POR ($00)	0	0	0	0	0	0	0	0
Other resets	U	U	U	U	U	U	U	U

Table 204. FRC timer low register (FRCTIMERL) (address $1882)

Bit	7	6	5	4	3	2	1	0
R W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
POR ($00)	0	0	0	0	0	0	0	0
Other resets	U	U	U	U	U	U	U	U

Table 205. FRCTIMERH/L register field descriptions

Field	Description
15:0 FRCTIM ERH/L	The 16-bit FRC timer register holds the value of the free running count, which can be halted or restarted, or reset to $0000. $0000 = Result of Reset

10.25.2.3 FRC compare high and low registers (FRCCOMP2/1)

Table 206. FRC compare high register (FRCCOMP2) (address $1883)

Bit	15	14	13	12	11	10	9	8
R W	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
POR ($FF)	1	1	1	1	1	1	1	1
Other resets	U	U	U	U	U	U	U	U

Table 207. FRC compare low register (FRCCOMP1) (address $1884)

Bit	7	6	5	4	3	2	1	0
R W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
POR ($FF)	1	1	1	1	1	1	1	1
Other resets	U	U	U	U	U	U	U	U

Table 208. FRCCOMP2/1 register field descriptions

Field	Description
15:0 FRCCO MP2/1	The 16-bit FRC compare register holds the value of the target comparison for which the interrupt will occur if enabled. $FFFF = Result of Reset

10.26 Other MCU resources

NXP provides a CodeWarrior starter project and a library of the firmware routines, including a user manual, available for download. Customer software applications are developed from the starter project, including the specific firmware routines from the library as needed.

It is not intended that physical parameter measurements be made during the time that LFR may be actively receiving/decoding LF signals; or during the time that the RFM may be actively powered up and/or transmitting RF data. The resulting interactions will degrade the accuracy of the measurements.

The NTM88 measures six physical parameters for use in the tire pressure monitoring application: pressure, temperature, battery voltage, two external voltages and an optional X- and/or Z-axis acceleration. Each parameter is accessed in a different manner and all use firmware subroutine calls as described in the firmware user guide. These subroutines

initialize some control bits within the sensor measurement interface, SMI, and then place the MCU into the STOP4 mode until the measurement is completed with an interrupt back to the MCU.

The accuracy, power consumption, and timing specified for any measurement provided in the data sheet are only guaranteed if the user obtains a reading using the specified firmware subroutine call in firmware user guide. For additional information, contact your NXP sales representative.

The NTM88 uses a 6-channel, 10-bit analog-to-digital converter (ADC10) module. The ADC10 module is an analog-to-digital converter using a successive approximation register (SAR) architecture with sample and hold. Capture of pressure and acceleration sensor readings is controlled by the sensor measurement interface (SMI) and capture of temperature and voltage readings are controlled by the MCU.

When making measurements of the various analog voltages the individual blocks will first be powered up long enough to stabilize their outputs before a conversion is started. The ADC channels are connected in hardware. Conversions are started and ended synchronously with the sampling of the voltages.

The accuracy, power consumption, and timing specifications given in the data sheet are based on using the assigned firmware subroutines in the firmware user guide to make these measurements and convert them into an 8-bit, 9-bit or 10-bit transfer function. These measurement accuracy specifications cannot be guaranteed if the user creates custom software routines to convert these measurements. For additional information, contact your NXP sales representative.

Table 209. ADC10 channel assignments

ADC10 Channel	Input Select	Firmware Call(s)	Characteristic
AD0	Pressure Sensor	TPMS_READ_COMP_PRESSURE	P_{CODE}
	Optional X-axis Acceleration Sensor	TPMS_READ_COMP_ACCEL_X	A_{XCODE}
	Optional Z-axis Acceleration Sensor	TPMS_READ_COMP_ACCEL_Z	A_{ZCODE}
AD1	Temperature Sensor	TPMS_READ_COMP_TEMP	T_{CODE}
AD2	Band gap Reference	TPMS_READ_COMP_VOLTAGE	V_{CODE}
AD3	GPIO PTA0	TPMS_READ_V0	$G0_{CODE}$
AD4	GPIO PTA1	TPMS_READ_V1	$G1_{CODE}$
AD5	V_{REG} Monitor	TPMS_WIRE_CHECK	

10.26.1 Pressure measurement

The pressure measurement consists of an interface to a pressure sensing element. Control bits on the MCU operate the SMI to power up the P-Cell and capture a voltage which is converted by the ADC10. The resulting pressure transfer equation for the 100-500 kPa range:

$$P = \Delta P_{500} \times P_{CODE} + \left(P_{MIN} - \Delta P_{500} \right) \quad (1)$$

The transfer equation of the 100-900 kPa range is:

$$P = \Delta P_{900} \times P_{CODE} + (P_{MIN} - \Delta P_{900}) \quad (2)$$

The transfer equation of the 100-1500 kPa range is:

$$P = \Delta P_{1500} \times P_{CODE} + (P_{MIN} - \Delta P_{1500}) \quad (3)$$

Due to calibration routines and parameters stored in the NTM88, the pressure range is selected at production and cannot be changed in the field.

Note: *Lack of change of the pressure measurement over time may indicate the package pressure port to be blocked or the internal section of the sensor to be contaminated. User application should maintain either locally or at the system data receiver a record of pressure measurements along with temperature and/or accelerometer measurements, and possibly identify the pressure port as blocked or contaminated if no changes are recorded over time.*

Refer to the specific data sheet for reference descriptions of the equation terms, which differ by part number variants.

10.26.2 Temperature measurements

The temperature is measured from a ΔV_B sensor built into channel 1 of the ADC10 in the same manner as is done in the NTM88 devices with the resulting transfer equation:

$$T = \Delta T \times T_{CODE} - 55 \quad (4)$$

Refer to the specific data sheet for reference descriptions of the equation terms, which differ by part number variants.

10.26.3 Voltage measurements

Voltage measurements can be made on the internal band gap to estimate the supply voltage on V_{DD}.

Refer to the specific data sheet for reference descriptions of the equation terms, which differ by part number variants.

10.26.3.1 Internal band gap

An internal band gap voltage reference is provided to take measurements of the supply voltage. The resulting transfer equation:

$$V_{INT} = \Delta V_{INT} \times V_{CODE} + 1.22 \quad (5)$$

10.26.3.2 External voltages

Measurements of an external voltage on either the PTA0 or PTA1 pins can be made and referenced to the internal band gap voltage. The resulting transfer equation:

$$V_{PTAx} = \Delta V_{EXT} \times Gx_{CODE} \quad (6)$$

where x = 0, 1 refers to PTA0 or PTA1.

10.26.4 Optional acceleration measurements

The acceleration measurement consists of an interface to an optional acceleration sensing element. Control bits on the MCU operate the SMI to power up the g-Cell and capture a voltage which is converted by the ADC10. The data from the ADC10 is then pre-processed by a dynamic range firmware routine that will return the two values necessary to calculate the acceleration, A_y, (y = X-axis or Z-axis, depending on selection) in conjunction with values taken from the tables in the data sheet.

Acceleration sensitivity, $\Delta A_{MAX\text{-}MIN}$, varies between each offset step, and should be calculated by dividing the range of g's for each offset step by the usable A_{yCODE} range:

$$\Delta A_{\text{MIN-MAX}} = \frac{(\text{Proof Inertia @ A}_{\text{RATE-MAX}} - \text{Proof Inertia @ A}_{\text{RATE-MIN}})}{A_{\text{RATE-MAX}} - A_{\text{RATE-MIN}}} \quad (7)$$

Once the sensitivity $\Delta A_{MAX\text{-}MIN}$ has been calculated, the acceleration A_y can be calculated by the re-using the $A_{RATE\text{-}MIN}$, 1 value of the offset step and the returned A_{yCODE} value with the following transfer function:

$$A_y = \Delta A_{\text{MAX-MIN}} \times A_{CODE} + (\text{Proof Inertia @ A}_{\text{RATE-MIN}} - \Delta A_{\text{MAX-MIN}}) \quad (8)$$

The pressure, and optional X or Z-axis accelerometer also share the same signal path in the Transducer interface and all the sensors share the same ADC. Therefore, only one of the sensors can be accessed at a given moment.

10.26.5 Optional battery condition check

The condition of the battery can be periodically checked to determine the battery's internal impedance, R_{BATT}, which is a function of both temperature and the remaining battery capacity. This can be performed by user supplied software routine and an external load resistor, R_{LOAD}, connected from the PTA0 pin to V_{SS} as shown in Figure 58 (any of the PTA[3:0] can be used for this purpose).

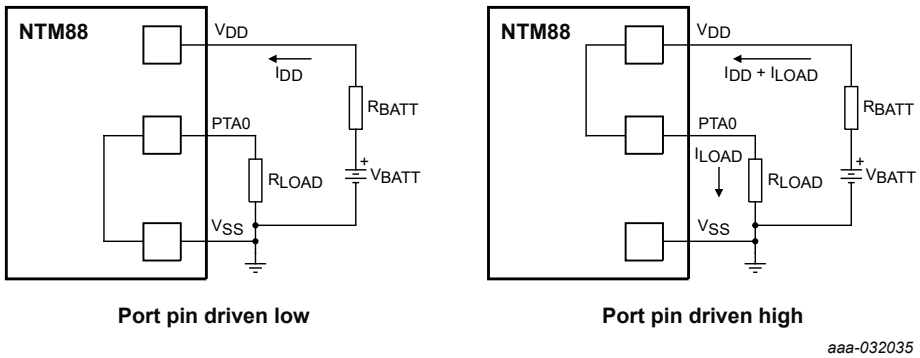


Figure 58. Battery check circuit

$$V_{DD0} = V_{BATT} - I_{DD0} \times R_{BATT} \qquad V_{DD1} = V_{BATT} - (I_{DD1} + I_{LOAD}) \times R_{BATT}$$

$$V_{DD1} = V_{BATT} - \left(I_{DD1} + \frac{V_{DD1}}{R_{LOAD}}\right) \times R_{BATT}$$

The battery voltage can first be checked using the method given in Section 10.26.3 "Voltage measurements" with the selected PTA0 pin set as an output and driven low and then high to determine V_{DD} where only I_{DD} flows or when I_{DD} plus I_{LOAD} flows. The resulting battery impedance can then be calculated as:

$$R_{BATT} = \frac{V_{DD1} - V_{DD0}}{I_{DD0} - I_{DD1} + \frac{V_{DD1}}{R_{LOAD}}} \quad (9)$$

If it is assumed that I_{DD0} and I_{DD1} are not appreciably different at the small change in V_{DD}, then the resulting battery impedance can be approximated as:

$$R_{BATT} = \frac{V_{DD1} - V_{DD0}}{\frac{V_{DD1}}{R_{LOAD}}} = \frac{R_{LOAD} \times (V_{DD1} - V_{DD0})}{V_{DD1}} \quad (10)$$

Where:

V_{DD0} = the voltage determined with the external load resistor connected to V_{SS}

V_{DD1} = the voltage determined with the external load resistor connected to V_{DD}

R_{LOAD} = the resistance of the external load resistance in ohms

R_{BATT} = the implied battery impedance in ohms

NXP recommends this calculation be performed with a reasonable current load on the battery of approximately 3 mA (R_{LOAD} approximately 1000 ohms).

10.26.6 Measurement firmware

The firmware library routines for making measurements are comprised of two function calls as described in the firmware user guide. Each measurement is a combination of a "read" that returns the raw ADC output data and a "comp" routine which compensates that raw reading based on information contained in the Universal Uncompensated Measurement Array (UUMA) assigned in RAM memory.

The read routines fill specific locations in the UUMA with raw data; but the compensation routines depend what is already present in the UUMA as shown in the data flow in Figure 59.

The user, therefore, has the option to decide how often each measurement (and its component terms) is made. The resulting power consumption is then the sum of using these components are defined in the product data sheet.

A typical flow for a compensated pressure measurement would be:

1. Call the TPMS_READ_PRESSURE routine which yields a raw pressure value and fills the UUMA with this data.
2. Call the TPMS_READ_TEMPERATURE routine which yields a raw temperature value and fills the UUMA with this data.
3. Call the TPMS_READ_VOLTAGE routine which yields a raw voltage value and fills the UUMA with this data.

4. Call the TPMS_COMP_PRESSURE routine which then takes the raw pressure, temperature and voltage values from the UUMA and compensates to provide a true pressure reading to the accuracy as specified in the product data sheet.

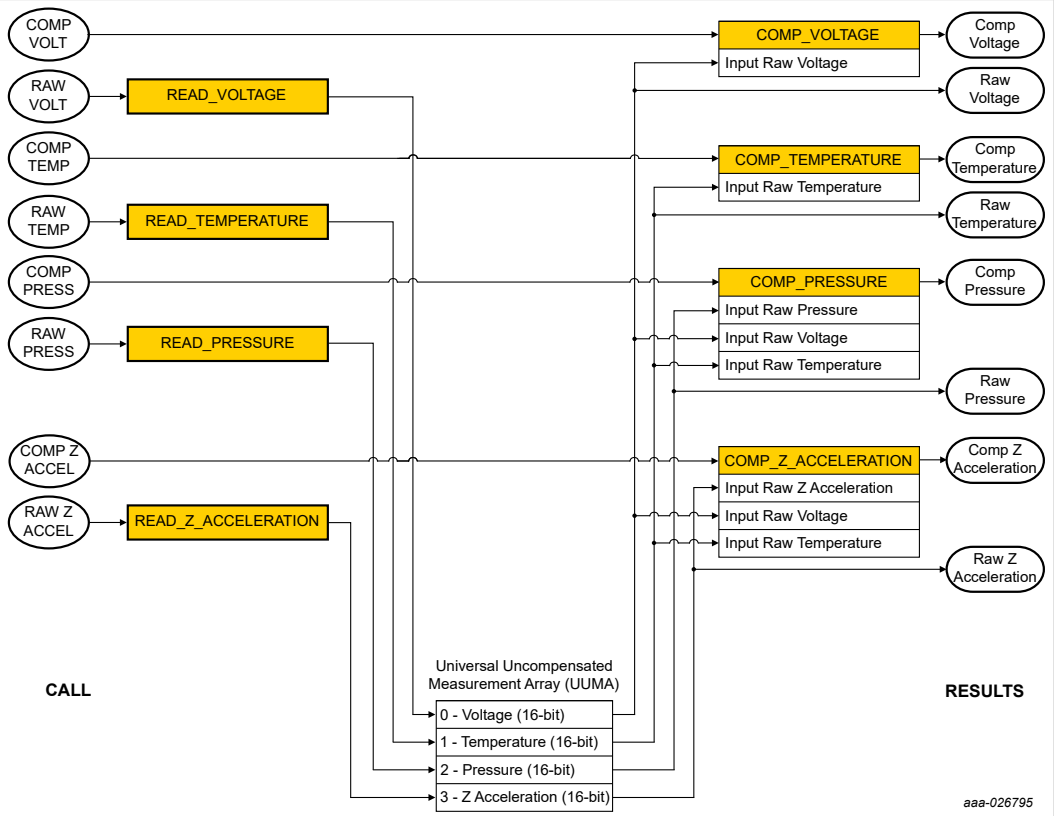


Figure 59. Example single axis accelerometer data flow for measurements

10.27 Battery charge consumption modeling

The supply current consumed by the NTM88 can be estimated using the following basic model.

10.27.1 Standby current

The overall charge consumed by the standby features is:

$$Q_{STDBY} = t_{TOT} \times \frac{(I_{STDBY} + I_{LF})}{1000} \quad (11)$$

Where:

Q_{STDBY} = Standby charge over lifetime, t_{TOT}, in mA-hr

t_{TOT} = Total lifetime in hours

I_{STDBY} = General standby current in µA

I_{LF} = LFR detector (if used) current in µA

10.27.2 Measurement events

The overall charge consumed by the measurements is:

$$Q_{MEAS} = \frac{1}{1000} \times \left(n_{PRESS} \times Q_{PRESS} + n_{TEMP} \times Q_{TEMP} + n_{VOLT} \times Q_{VOLT} \right) \quad (12)$$

Where:

Q_{MEAS} = Total measurement charge over lifetime in mA-sec

Q_{PRESS} = Measurement charge per pressure measurement in µA-sec

Q_{TEMP} = Measurement charge per temperature measurement in µA-sec

Q_{VOLT} = Measurement charge per voltage measurement in µA-sec

n_{PRESS} = Total number of pressure measurements over lifetime

n_{TEMP} = Total number of temperature measurements over lifetime

n_{VOLT} = Total number of voltage measurements over lifetime

10.27.3 Transmission events

The overall charge consumed by the transmissions is:

$$Q_{XMT} = \frac{Q_{FRM}}{1000} \times F \times n_{XMT} \quad (13)$$

Where:

Q_{XMT} = Transmit charge over lifetime, t_{TOT}, in mA-hr

Q_{FRM} = Transmit charge per frame of data in µA-sec

n_{XMT} = Number of transmissions over lifetime

F = Frames transmitted during each datagram

10.27.4 Total consumption

The overall charge consumed is:

$$Q_{TOT} = \frac{\left(Q_{STDBY} + Q_{MEAS} + Q_{XMT}\right)}{\left(1 - Y \times SD/100\right)} \quad (14)$$

Where:

Q_{TOT} = Total charge over lifetime, t_{TOT}, in mA-hr

Q_{STDBY} = Standby charge over lifetime in mA-hr

Q_{MEAS} = Measurement charge over lifetime in mA-hr

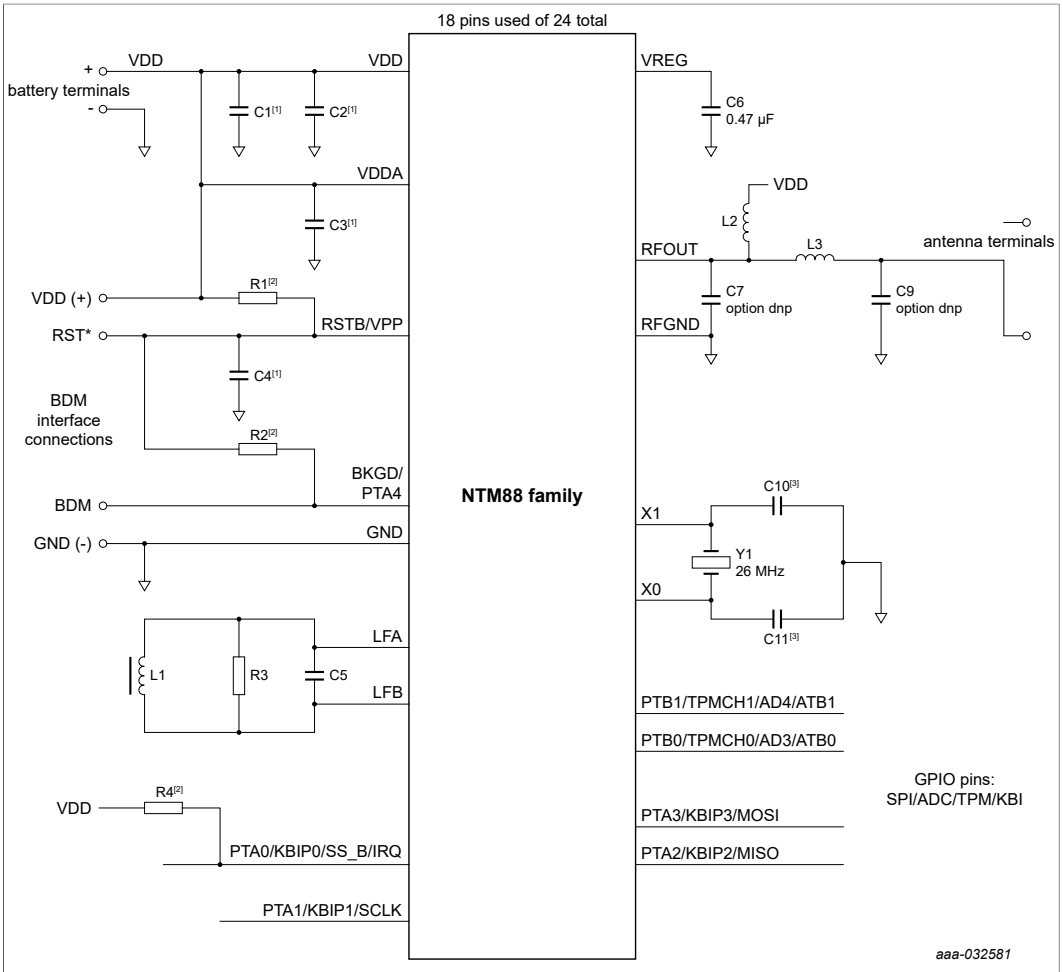
Q_{XMT} = Transmit charge over lifetime in mA-hr

Y = Lifetime in years

SD = Battery self-discharge rate in % / year

Additional margin in battery capacity can be added to the calculated value of Q_{TOT}.

11 Application information



[1] C1, C2, C3 and C4 values may be altered for EMI performance.
[2] R1, R2 and R4 are optional if needed for EMI performance.
[3] C10 and C11 values are in accordance with recommendations of Y1 manufacturer.

Note: *R1, R2, and R4 remain optional. R1, R2, and R4 are recommended to avoid potential inadvertent activation of internal circuits under adverse environment conditions.*

Figure 60. Example electrical schematic diagram

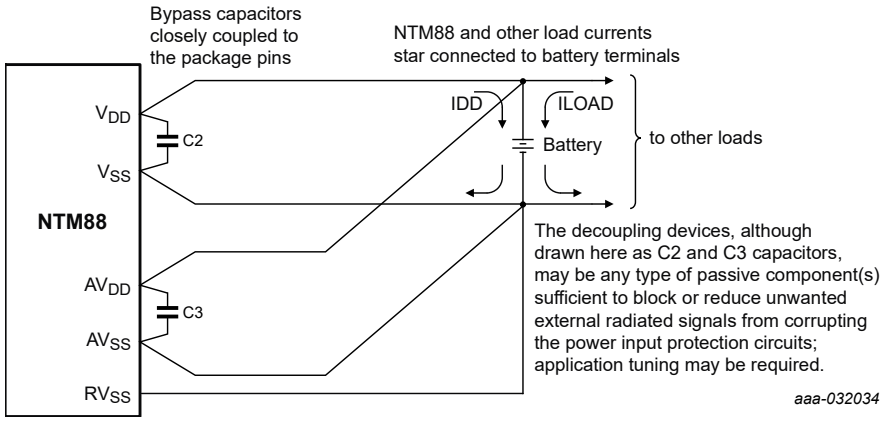


Figure 61. Recommended power supply connections

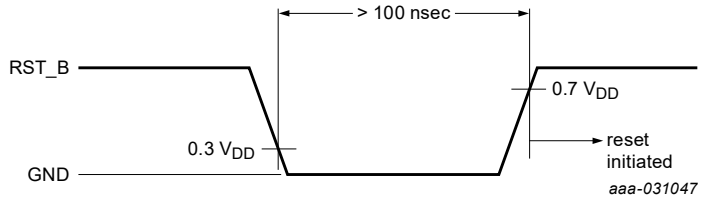


Figure 62. RESET pin timing

Additional components or printed circuit artwork, not depicted in Figure 60, may be necessary to sufficiently block or reduce unwanted external radiated signals from corrupting the crystal oscillator circuit or power supplies. Minimize PCB traces for LFA / LFB, AVDD / VDD, and VSS / AVSS pins and bypass capacitors to reduce unwanted, external radiated signals from corrupting the power input circuits.

The application should assure the pressure applied to all sensor surfaces remains equal. The sensor is constructed from non-hermetic materials and should not be used as a seal between the tire pressure and ambient environment pressure. The seal should be provided by the final module design, not by the surfaces of the sensor.

A gel is used to provide media protection against corrosive elements which may otherwise damage metal bond wires and/or IC surfaces. Highly pressurized gas molecules may permeate through the gel and occupy boundaries between material surfaces within the sensor package. When decompression occurs, gas molecules may collect and form bubbles, or possibly result in delamination of the gel from the material it protects. If a bubble is located on the pressure transducer surface or on the bond wires, the sensor measurement may be shifted from its calibrated transfer function. In some cases, these temporary shifts could be outside the tolerances listed as part of this data sheet. In rare cases, the bubble may bend the bond wires, and result in a permanent shift. The media may also contain electro-static elements. If the highly pressurized media permeates the gel, including contact with the wires or IC surfaces, the same type of permanent shifts can occur, due to the electro-static elements disturbing the calibrated transfer function.

The moving parts of micro-machined electro-mechanical systems (MEMS) accelerometers are fabricated from materials that may adhere to each other when forced into contact. Known as 'stiction', if this occurs, the sensor measurement may be shifted from its calibrated transfer function. In some cases, these temporary shifts could be

outside the tolerances listed as part of the data sheet. In rare cases, this may result in a permanent shift.

Upon power-on reset release or external RSTB rising edge release, the PTx pins default to GPIO inputs with pull-up/down disabled.

PTA4 provides the BKGD function if externally held low prior to the exit release of POR. PTA0:3 provides the SPI function if PTA0 externally held low prior to the exit release of POR.

PTA0 provides the external IRQ function if the IRQ function is enabled prior to entering a stop mode.

12 Legal information

12.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

12.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors. In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory. Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Suitability for use in automotive applications — This NXP Semiconductors product has been qualified for use in automotive applications. Unless otherwise agreed in writing, the product is not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

12.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

CodeWarrior — is a trademark of NXP B.V.

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 29 March 2021
Document identifier: UM11227