Xtrinsic MPXx85/86xxD Tire Pressure Monitor Sensor

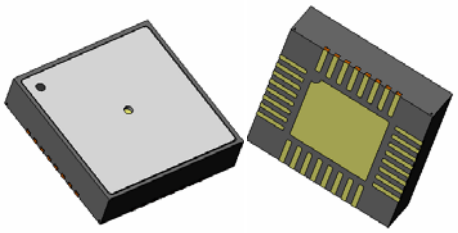
MPXx85/86xxD

The MPXx85/86xxD is a sensor for use in applications that monitor tire pressure and temperature. It contains the pressure and temperature sensors, an X-axis and a Z-axis accelerometer, a microcontroller, an LF receiver and an RF transmitter all within a single package.

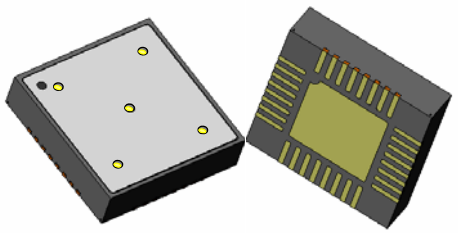
Features

- Pressure sensor with one of two calibrated pressure ranges
- Temperature sensor
- Optional X-axis and Z-axis accelerometers
- Voltage reference measured by ADC10
- 6-channel, 10-bit analog-to-digital converter (ADC10) with two external I/O inputs
- 8-bit MCU
  - S08 Core with SIM, interrupt and debug/monitor
  - 512 RAM
  - 8K FLASH (in addition to 8K providing factory firmware and trim data)
  - 64-byte, low-power, parameter registers
- Dedicated state machines to sequence routine measurement and transmission processes for reduced power consumption
- Internal 315/434 MHz 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 (less than 8 mA at 434 MHz, 5 dBM at 3.0 V, 25°C)
- Differential input LF detector/decoder
- Six multipurpose GPIO pins
  - Optional pullups/pulldowns and wake-up interrupt
  - Two pins can be connected to a channel on the ADC10
  - Two pins can be connected to a channel on the TPM1
  - The two LF receiver pins (LFA/LFB) can be configured instead as GPIO
- Real-time interrupt driven by LFO with interrupt intervals of 8, 16, 32, 64, 128, 256, 512 or 1024 msec
- Low-power, wake-up timer and periodic reset driven by LFO

Top and bottom view

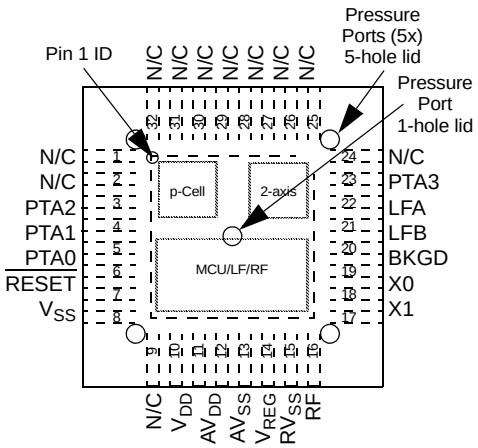


32-Pin, 1-hole lid
9 x 9 QFN
Case 2265-02



32-Pin, 5-hole lid
9 x 9 QFN
Case 2038-06

Top view



Pin connections

freescale Xtrinsic

- 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 msec period
    - Medium-frequency, controller clock (MFO) of 8 msec period
- Low-voltage detection
- Normal temperature restart in hardware (over or under temperature detected by software)

Table 1. Ordering information[1]

Part number	Accelerometer axis		Pressure range	Package
	X	Z		
MPXY8500DK016T1	N/A	Acceleration measurement from -270 g to 350 g	100-900 kPa	2265 (QFN 9 x 9, 1-hole lid)
MPXV8500DK016T1	N/A	Acceleration measurement from -270 g to 350 g	100-900 kPa	2038 (QFN 9 x 9, 5-hole lid)
MPXY8510DK016T1	N/A	Acceleration measurement from -270 g to 350 g	100-450 kPa	2265 (QFN 9 x 9, 1-hole lid)
MPXV8510DK016T1	N/A	Acceleration measurement from -270 g to 350 g	100-900 kPa	2038 (QFN 9 x 9, 5-hole lid)
MPXY8600DK6T1	Acceleration measurement from -70 g to 80 g	Acceleration measurement from -210 g to 240 g	100-900 kPa	2265 (QFN 9 x 9, 1-hole lid)
MPXY8610DK6T1	Acceleration measurement from -70 g to 80 g	Acceleration measurement from -210 g to 240 g	100-450 kPa	2265 (QFN 9 x 9, 1-hole lid)

1. This table includes the only orderable part numbers covered by the current data sheet.

Contents

Related Documentation

The MPXx85/86xxD device features and operations are described in a variety of reference manuals, user guides, and application notes. To find the most-current versions of these documents:

1. Go to the Freescale homepage at:

   http://www.freescale.com/

2. In the Keyword search box at the top of the page, enter the device number MPXx85/86xxD.
3. In the Refine Your Result pane on the left, click on the Documentation link.

1 General Description

1.1 Overall block diagram

The block diagram of the MPXx85/86xxD is shown in Figure 1. This diagram covers all the main blocks mentioned above and their main signal interactions. Power management controls and bus control signals are not shown in this block diagram for clarity.

1.2 Multi-chip interface

The MPXx85/86xxD contains two to three devices using the best process technology for each.

- Microcontroller with accelerometer and pressure sensor interfaces, and RF transmitter (MCU)
- Pressure transducer
- Combined X-axis and Z-axis acceleration transducer

As shown in Figure 1 the MCU interfaces to the RF transmitter using a standard memory mapped registers. The transducers connect to the MCU using custom analog interfaces and inter-chip bonding wires.

1.3 System clock distribution

The various clock sources and their distribution are shown in Figure 2. All clock sources except the low-frequency oscillator, LFO, can be turned off by software control in order to conserve power.

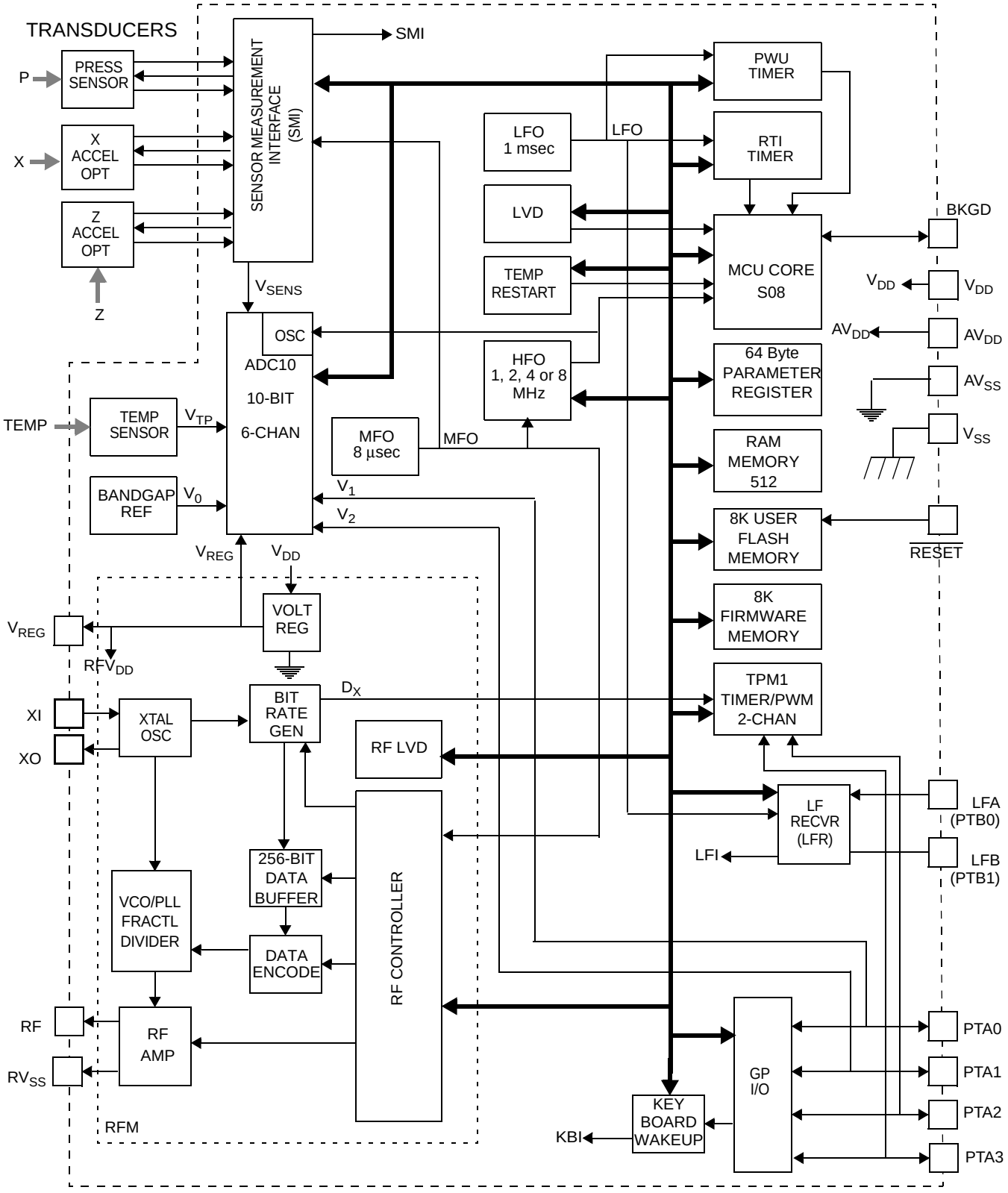


Figure 1. MPXx85/86xxD overall block diagram

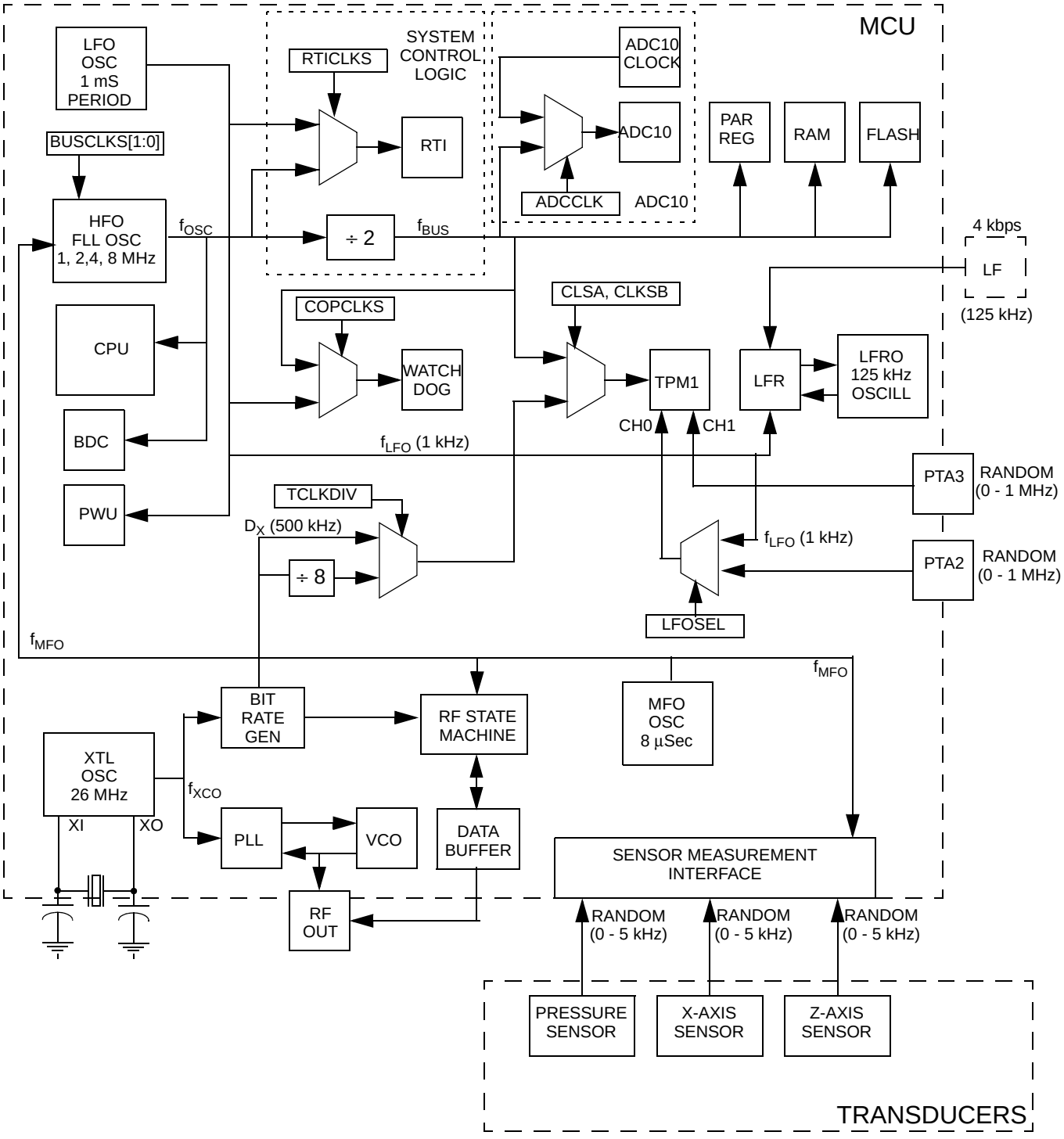


Figure 2. MPXx85/86xxD clock distribution

1.4 Reference documents

The MPXx85/86xxD utilizes the standard product MC9S08 CPU core. The user can obtain further detail on the full capabilities of this core by referring to the HCS08 Family Reference Manual (HCS08RMV1).

2 Pins and Connections

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

2.1 Package pinout

This section describes the pin layout and general function of each pin. The pinout for the QFN MPXx85/86xxD device package is shown in Figure 3 and the orientation of the 2-axis accelerometer for the pressure port up.

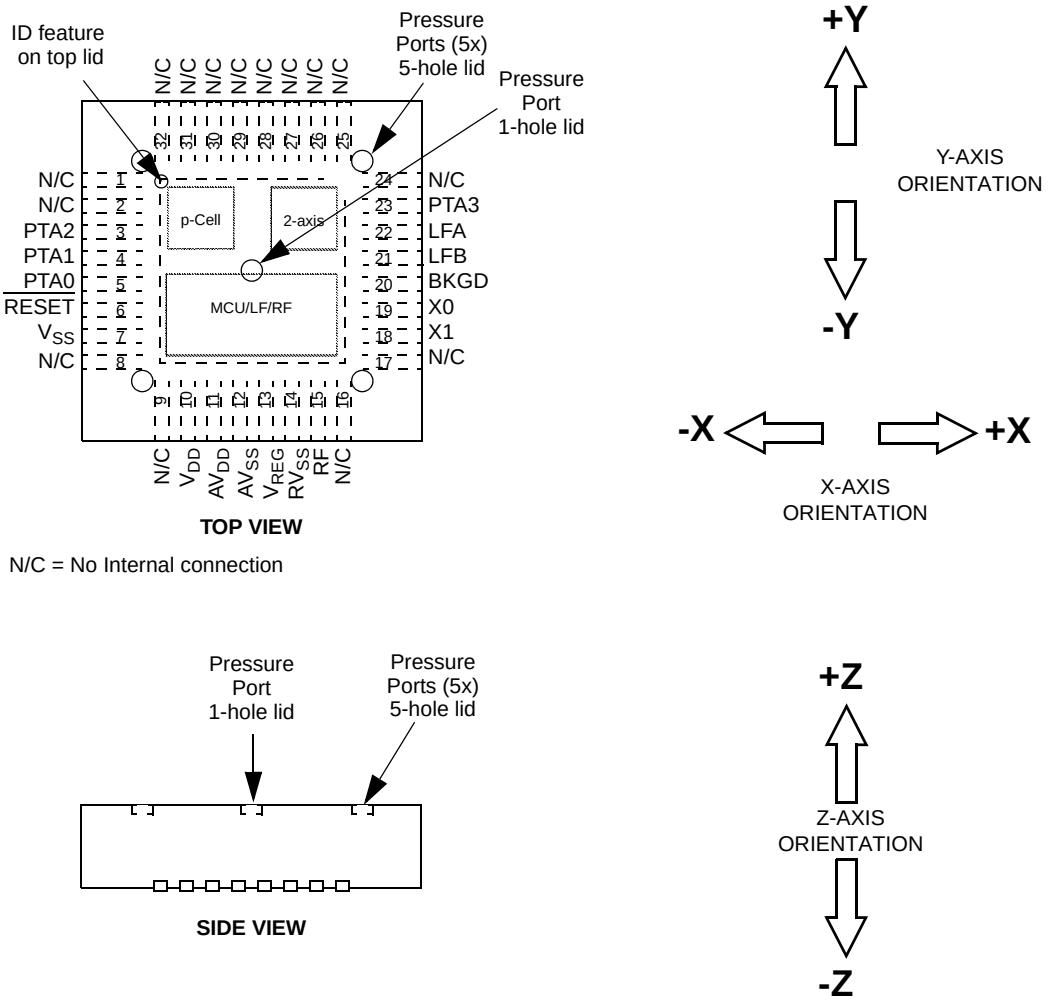


Figure 3. MPXx85/86xxD QFN package pinout

2.2 Recommended application

Example of a simple OOK/FSK tire pressure monitors using the internal PLL-based RF output stage is shown in Figure 4. Any of the PTA[3:0] pins can also be used as general purpose I/O pins. Any of the PTA[3:0] pins that are not used in the application should be handled as described in Section 6.1.

2.3 Signal properties

The following sections describe the general function of each pin.

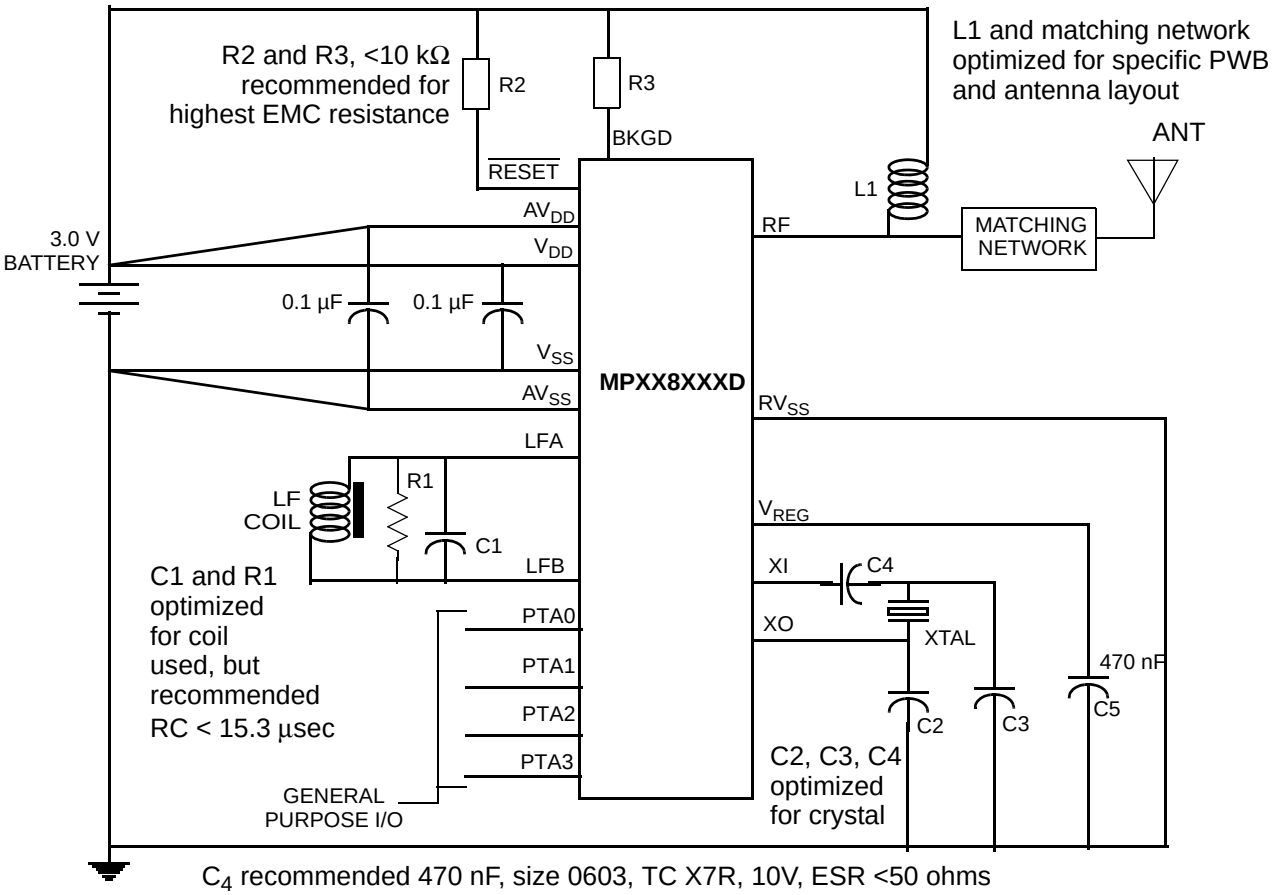


Figure 4. MPXx85/86xxD example application

2.3.1 V_{DD} and V_{SS} pins

The digital circuits operate from a single power supply connected to the MPXx85/86xxD through the V_{DD} and V_{SS} pins. V_{DD} is the positive supply and V_{SS} is the ground. The conductors to the power supply should be connected to the V_{DD} and V_{SS} pins and locally decoupled as shown in Figure 5.

Care should be taken to reduce measurement signal noise by separating the V_{DD}, V_{SS}, AV_{DD}, AV_{SS} and RV_{SS} pins using a “star” connection such that each metal trace does not share any load currents with other external devices as shown in Figure 5.

2.3.2 AV_{DD} and AV_{SS} pins

The analog circuits operate from a single power supply connected to the MPXx85/86xxD through the AV_{DD} and AV_{SS} pins. AV_{DD} is the positive supply and AV_{SS} is the ground. The conductors to the power supply should be connected to the AV_{DD} and AV_{SS} pins and locally decoupled as shown in Figure 5.

Care should be taken to reduce measurement signal noise by separating the V_{DD}, V_{SS}, AV_{DD}, AV_{SS} and RV_{SS} pins using a “star” connection such that each metal trace does not share any load currents with other external devices as shown in Figure 5.

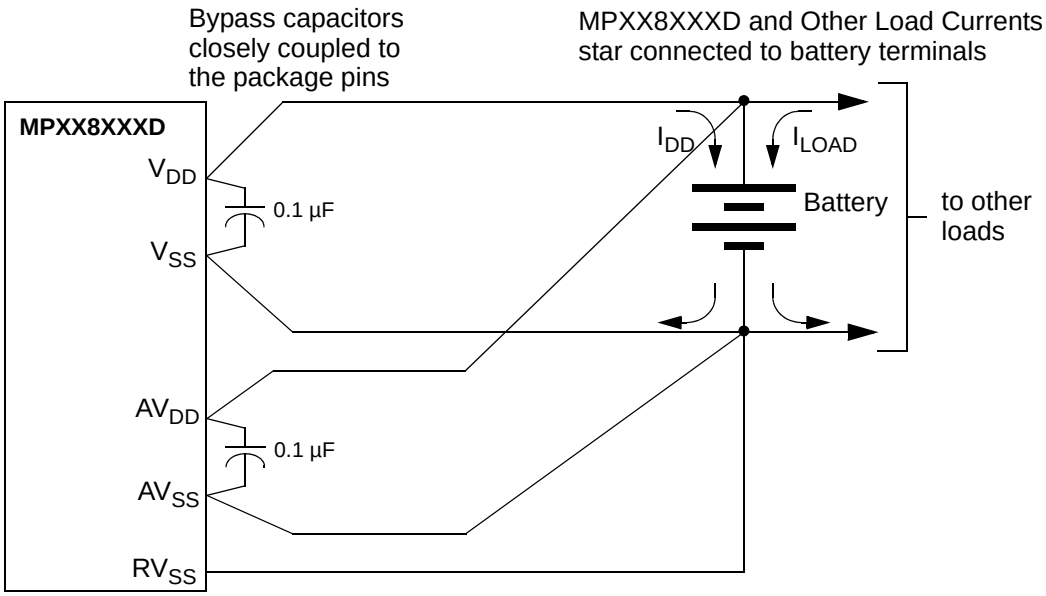


Figure 5. Recommended power supply connections

2.3.3 V_{REG} pin

The internal regulator for the analog circuits requires an external stabilization capacitor to AV_{SS}.

2.3.4 RV_{SS} pin

Power in the RF output amplifier is returned to the supply through the RV_{SS} pin. This conductor should be connected to the power supply as shown in Figure 5 using a “star” connection such that each metal trace does not share any load currents with other supply pins.

2.3.5 RF pin

The RF pin is the RF energy data supplied by the MPXx85/86xxD to an external antenna.

2.3.6 XO, XI pins

The XO and XI pins are for an external crystal to be used by the internal PLL for creating the carrier frequencies and data rates for the RF pin.

2.3.7 LF[A:B] and PTB[1:0] pins

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 LFA and the LFB pins.

Signaling into the LFR pins can place the MPXx85/86xxD into various diagnostic or operational modes. The LFR is comprised of the detector and the decoder.

These two pins can also be configured as normal bidirectional I/O pins, PTB[1:0], with a programmable pullup device, but each will always have an impedance of approximately 500 kΩ to V_{SS} due to the LFR input circuitry. The LFA/LFB pins are used by the LFR when the LFEN control bit is set and are GPIO when the LFEN control bit is clear with the LFA in used as PTB0 and the LFB used as PTB1. User software must configure the general purpose I/O pins so that they do not result in “floating” inputs as described in Section 6.1.

2.3.8 PTA[1:0] pins

The PTA[1:0] pins are general purpose I/O pins. These two pins can be configured as normal bi-directional I/O pins with programmable pullup or pulldown devices and/or wake-up interrupt capability; or one or both can be connected to the two input channels of the A/D converter module. 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 as described in Section 6.1.

2.3.9 PTA[2:3] Pins

The PTA[2:3] pins are general purpose I/O pin. These two pins can be configured as normal bi-directional I/O pin with programmable pullup or pulldown devices and/or wake-up interrupt capability; or one or both can be connected to the two input channels of the Timer Pulse Width (TPM1) module. 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 as described in Section 6.1.

2.3.10 BKGD pin

The BKGD pin is used to place the MPXx85/86xxD in the Background Debug mode (BDM) to evaluate MCU code and to also transfer data to/from the internal memories. If the BKGD pin is held low when the MPXx85/86xxD comes out of a power-on reset the device will go into the Active Background Debug mode (BDM).

The BKGD pin has an internal pullup device and can connected to V_{DD} 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 pin can be left unconnected, but should be connected to V_{DD} 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.

2.3.11 $\overline{\text{RESET}}$ pin

The $\overline{\text{RESET}}$ 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 as described in Section 5.2.

The $\overline{\text{RESET}}$ pin has an internal pullup device and can connected to V_{DD} 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 $\overline{\text{RESET}}$ pin can be left unconnected; but should be connected to V_{DD} 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 RESET pin goes below 0.3 x V_{DD} for at least 100 nsec before rising above 0.7 x V_{DD} as shown in Figure 6.

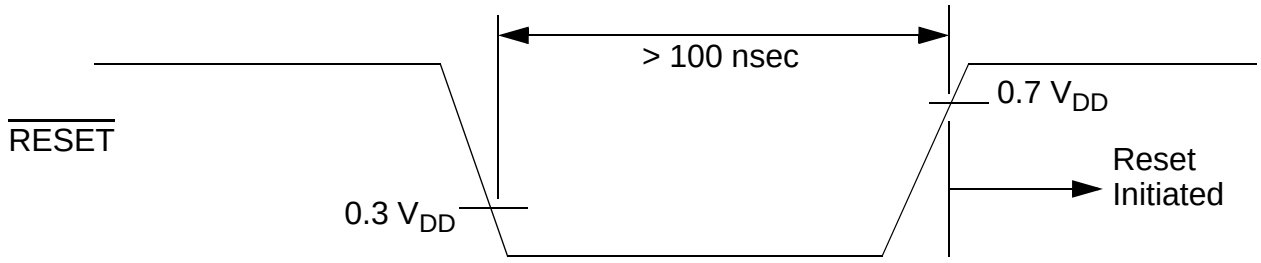


Figure 6. RESET pin timing

3 Modes of Operation

The operating modes of the MPXx85/86xxD are described in this section. Entry into each mode, exit from each mode, and functionality while in each of the modes are described.

3.1 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

3.2 Run mode

This is the normal operating mode for the MPXx85/86xxD. This mode is selected when the BKGD 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).

3.3 Wait mode

The wait mode is also present like other members of the Freescale S08 family members; but is not normally used by the MPXx85/86xxD firmware or typical TPMS applications.

3.4 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 pin is low at the rising edge of a power up reset
- When a BACKGROUND command is received through the BKGD 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 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 bootloader or user application program into the FLASH program memory before the MCU is operated in Run mode for the first time. When the MPXx85/86xxD is shipped from the Freescale 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.

3.5 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 1 summarizes the behavior of the MCU in each of the Stop1 and Stop4 modes. The Stop2 mode found in other Freescale S08 family members is not available; but the Stop3 mode is also present like other members of the Freescale S08 family members; but is not normally used by the MPXx85/86xxD firmware or typical TPMS applications.

3.5.1 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. Exit from Stop1 is done 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.

3.5.2 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 will enter Stop4 instead.

Table 1. Stop mode behavior

Mode	Stop1	Stop4
LFO Oscillator, PWU	Always On & Clocking	
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-Channel Timer/PWM	Off	Off
Digital I/O	Disabled	Standby
Sensor Measurement Interface (SMI)	Off	Optionally On
Pressure P-cell	Off	Optionally On
Z-axis acceleration g-cell	Off	Optionally On
X-axis acceleration g-cell	Off	Optionally On
Temperature Sensor (in ADC10)	Off	Optionally On(3)
Normal Temperature Restart	Optionally On	Optionally On
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
RF Transmitter(5)	Optionally On	Optionally On

Table 1. Stop mode behavior (continued)

Mode	Stop1	Stop4
ADC10	Off	Optionally On
Regulator	Off	On
I/O Pins	Hi-Z	States Held
Wake-Up Methods	Interrupts, resets	Interrupts, resets

1. RTI can be used in Stop1 or Stop4 if the clock selected is the LFO. To use the HFO as the clock the MCU must be in the Run mode.
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. PRF data buffer may be set up to run while the CPU is in the stop modes.

NOTE

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.

3.5.3 Active BDM Enabled in stop mode

Entry into the Active Background Debug mode from Run mode is enabled if the ENBDM bit in BDCSCR is set. 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 ENDBM 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 ENDBM bit is set. Once in Background Debug mode, all background commands are available.

3.5.4 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 (ENDBM = 1), where clocks are kept alive to the stop mode logic, clocks to the peripheral systems are halted to reduce power consumption.

I/O pins (optional PTA[3:0])

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.

Memory

All module interface registers will be 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.

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.

LFO

The LFO remains active regardless of any mode of operation.

MFO

The medium frequency oscillator (MFO) will remain powered up when the MCU enters the stop mode only when the SMI has been initiated to make a pressure, temperature, voltage or acceleration measurement; or when the RF transmitter’s state machine is processing data.

HFO

The HFO is halted in all stop modes.

PWU

The PWU remains active regardless of any mode of operation.

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.

LFR

When the MCU enters stop mode the detectors in the LFR will remain powered up depending on the states of the bits selecting the periodic sampling. Refer to Section 12 for more details.

Bandgap reference

The bandgap reference is enabled whenever the sensor measurement interface requires sensor or voltage measurements.

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 will be reset upon wake-up from stop and must be re-initialized.

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).

Temperature sensor

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

Temperature restart

When the MCU enters a stop mode the temperature restart will remain powered up if the TRE bit is set. If the temperature restart level is reached the MCU will restart from the reset vector.

3.5.5 RFM module in stop modes

When the MCU enters any stop mode the modules on the RFM may or may not power down depending on their respective control bits and whether RFM state machine or the MCU is in control of the RFM.

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.

3.5.6 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.

3.5.7 Optional Z- and X-axis measurements

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

4 Memory

The overall memory map of the MPXx85/86xxD resides on the MCU.

4.1 MCU memory map

As shown in Figure 7, MCU on-chip memory in the MPXx85/86xxD consists of parameter registers, RAM, FLASH program memory for nonvolatile data storage, and I/O and control/status registers. The registers are divided into four groups:

- Direct-page registers ($0000 through $004F)
- Parameter registers ($0050 through $008F)
- RAM ($0090 through $028F)
- High-page registers ($1800 through $182B)

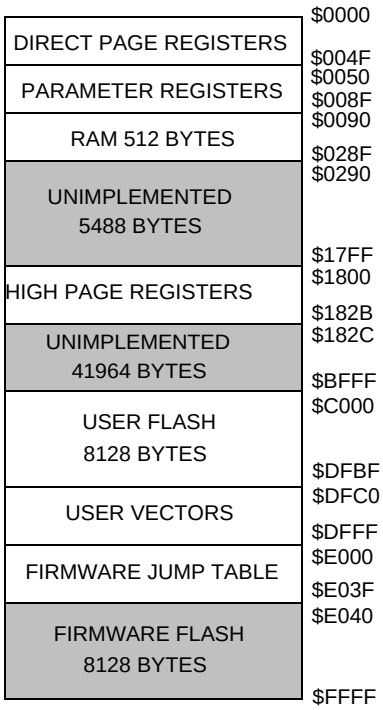


Figure 7. MPXx85/86xxD MCU memory map

The total programmable FLASH memory map is 16K, but the upper 8K is used for firmware and test software. Upon power up the firmware will initialize the device and redirect all vectors to the user area from $DFC0 through $DFFF. Any calls to the firmware subroutines are accessed through a jump table starting at location $E000 (see Section 14).

4.2 Reset and interrupt vectors

Table 2 shows address assignments for jump table to the reset and interrupt vectors. The vector names shown in this table are the labels used in the equate file provided by Freescale in the CodeWarrior project file.

Table 2. Vector summary

User vector addr	Vector name	Module source
$DFE0:DFE1	Vkbi	KBI
$DFE2:DFE3	Reserved	
$DFE4:DFE5	Reserved	
$DFE6:DFE7	Vrti	Sys Ctrl - RTI
$DFE8:DFE9	Vlfrcvr	LFR
$DFEA:DFEB	Vadc1	ADC10
$DFEC:DFED	Vrf	RFM

Table 2. Vector summary (continued)

User vector addr	Vector name	Module source
$DFEE:DFEF	Vsm	SMI
$DFF0:DFF1	Vtpm1ovf	TPM1
$DFF2:DFF3	Vtpm1ch1	TPM1
$DFF4:DFF5	Vtpm1ch0	TPM1
$DFF6:DFF7	Vwuktmr	PWU
$DFF8:DFF9	Vlvd	Sys Ctrl - LVD
$DFFA:DFFB	Reserved	
$DFFC:DFFD	Vswi	SWI opcode
$DFFE:DFFF	Vreset	Sys Ctrl - POR, PRF, COP, LVD Temp Restart, Illegal opcode or address

4.3 MCU register addresses and bit assignments

The registers in the MPXx85/86xxD are divided into these four groups:

- Direct-page registers are located in the first 80 locations in the memory map; these are accessible with efficient direct addressing mode instructions.
- The parameter registers begin at address $0050; these are also accessible with efficient direct addressing mode instructions.
- High-page registers are used less often, so they are located above $1800 in the memory map. This leaves more room in the direct page for more frequently used registers and variables.
- The nonvolatile register area consists of a block of 16 locations in FLASH memory at $FFB0:FFBF. Nonvolatile register locations include:
  - Three values that are loaded into working registers at reset
  - An 8-byte back door comparison key that optionally allows the user to gain controlled access to secure memory.

Because the nonvolatile register locations are FLASH memory, they must be erased and programmed like other FLASH memory locations.

Direct page registers are located within the first 256 locations in the memory map, so they are accessible with efficient direct addressing mode instructions, which requires only the lower byte of the address. Bit manipulation instructions can be used to access any bit in any direct-page register. Table 3 is a summary of all user-accessible direct-page registers and control bits. Those related to the TPMS application and modules are described in detail in this specification.

The register names in column two of the following tables are shown in bold to set them apart from the bit names to the right. Cells that are not associated with named bits are shaded. A shaded cell with a 0 indicates this unused bit always reads as a 0. Shaded cells with dashes indicate unused or reserved bit locations that could read as 1s or 0s.

Table 3. MCU direct page register summary

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$00**00**	PTAD	—	—	—	—	PTAD[3:0]			
$00**01**	PTAPE	—	—	—	—	PTAPE[3:0]			
$00**02**	Reserved	—	—	—	—	—	—	—	—
$00**03**	PTADD	—	—	—	—	PTADD[3:0]			
$00**04**	PTBD	—	—	—	—	—	—	PTBD[1:0]	
$00**05**	PTBPE	—	—	—	—	—	—	PTBPE[1:0]	
$00**06**	Reserved	—	—	—	—	—	—	—	—
$00**07**	PTBDD	—	—	—	—	—	—	PTBDD[1:0]	
$00**08**	Reserved	—	—	—	—	—	—	—	—
$00**09**	Reserved	—	—	—	—	—	—	—	—
$00**0A**	Reserved	—	—	—	—	—	—	—	—
$00**0B**	Reserved	—	—	—	—	—	—	—	—
$00**0C**	KBISC	0	0	0	0	KBF	KBACK	KBIE	KBIMOD

Table 3. MCU direct page register summary (continued)

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$000D	KBIPE	—	—	—	—	KBIPE[3:0]			
$000E	KBIES	—	—	—	—	KBEDG[3:0]			
$000F	Reserved	—	—	—	—	—	—	—	—
$0010	TPM1SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
$0011	TPM1CNTH	Bit [15:8]							
$0012	TPM1CNTL	Bit [7:0]							
$0013	TPM1MODH	Bit [15:8]							
$0014	TPM1MODL	Bit [7:0]							
$0015	TPM1C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
$0016	TPM1C0VH	Bit [15:8]							
$0017	TPM1C0VL	Bit [7:0]							
$0018	TPM1C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
$0019	TPM1C1VH	Bit [15:8]							
$001A	TPM1C1VL	Bit [7:0]							
$001B	Reserved	—	—	—	—	—	—	—	—
$001C	PWUDIV	—	—	WDIV[5:0]					
$001D	PWUCS0	WUF	WUFAK	WUT[5:0]					
$001E	PWUCS1	PRF	PRFAK	PRST[5:0]					
$001F	PWUS	PSEL	0	CSTAT[5:0]					
$0020-27	LFR Registers	LFR Registers, see Table 4 and Table 5							
$0028	ADSC1	COCO	AIEN	ADCO	ADCH[4:0]				
$0029	ADSC2	ADACT	ADTRG	ACFE	ADCFGT	0	0	0	0
$002A	ADRH	0	0	0	0	ADR[11:8]			
$002B	ADRL	ADR[7:0]							
$002C	ADCVH	0	0	0	0	ADCV[11:8]			
$002D	ADCVL	ADCV[7:0]							
$002E	ADCFG	ADLPC	ADIV[1:0]		ADLSMP	MODE[1:0]		ADICLK[1:0]	
$002F	ADPCTL1	ADPC[7:0]							
$0030-4F	RFM Registers	RFM Registers, see Table 6 and Table 7							
$0050-8F	Parameter Reg	PARAM[63:0]							

1. Shaded bits are recommended to only be controlled by firmware or factory test.

Table 4. LFR register summary - LPAGE = 0

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$0020	LFCTL1	LFEN	SRES	CARMOD	LPAGE	IDSEL[1:0]		SENS[1:0]	
$0021	LFCTL2	LFSTM[3:0]				LFONTM[3:0]			
$0022	LFCTL3	LFDO	TOGMOD	SYNC[1:0]		LFCDTM[3:0]			
$0023	LFCTL4	LFDRIE	LFERIE	LFCDIE	LFIDIE	DECEN	VALEN	TIMOUT[1:0]	
$0024	LFS	LFDRF	LFERF	LFCDF	LFIDF	LFOVF	LFEOMF	LPSM	LFIAK
$0025	LFDATA	RXDATA[7:0]							
$0026	LFIDL	ID[7:0]							
$0027	LFIDH	ID[15:8]							

1. Shaded bits are recommended to only be controlled by firmware or factory test.

Table 5. LFR register summary - LPAGE = 1

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$00**20**	LFCTL1	LFEN	SRES	CARMOD	LPAGE	IDSEL[1:0]		SENS[1:0]	
$00**21**	LFCTL2	LFSTM[3:0]				LFONTM[3:0]			
$00**22**	LFCTRLD	—	—	—	—	—	—	ONMODE	CHK125
$00**23**	LFCTRLC	—	—	—	—	—	LOWQ[1:0]		DEQEN
$00**24**	LFCTRLB	—	—	—	—	LFPOL	—	—	—
$00**25**	LFCTRLA	—	—	—	—	LFCC[3:0]			
$00**26**	Reserved	—	—	—	—	—	—	—	—
$00**27**	Reserved	—	—	—	—	—	—	—	—

1. Shaded bits are recommended to only be controlled by firmware or factory test.

Table 6. RFM register summary - RPAGE = 0

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$00**30**	RFCR0	BPS[7:0]							
$00**31**	RFCR1	FRM[7:0]							
$00**32**	RFCR2	SEND	RPAGE	EOM	PWR[4:0]				
$00**33**	RFCR3	DATA	IFPD	ISPC	IFID	FNUM[3:0]			
$00**34**	RFCR4	RFBT[7:0]							
$00**35**	RFCR5	BOOST	LFSR[6:0]						
$00**36**	RFCR6	VCO_GAIN[1:0]		RFFT[5:0]					
$00**37**	RFCR7	RFIF	RFEF	RFVF	RFIAK	RFIEN	RFLVDEN	RCTS	RFMRST
$00**38**	PLLCR0	AFREQ[12:5]							
$00**39**	PLLCR1	AFREQ[4:0]					POL	CODE[1:0]	
$003**A**	PLLCR2	BFREQ[12:5]							
$003**B**	PLLCR3	BFREQ[4:0]					CF	MOD	CKREF
$003**C**	RFD0	RFD[7:0]							
$003**D**	RFD1	RFD[15:8]							
$003**E**	RFD2	RFD[23:16]							
$003**F**	RFD3	RFD[31:24]							
$00**40**	RFD4	RFD[39:32]							
$00**41**	RFD5	RFD[47:40]							
$00**42**	RFD6	RFD[55:48]							
$00**43**	RFD7	RFD[63:56]							
$00**44**	RFD8	RFD[71:64]]							
$00**45**	RFD9	RFD[79:72]							
$00**46**	RFD10	RFD[87:80]							
$00**47**	RFD11	RFD[95:88]							
$00**48**	RFD12	RFD[103:96]							
$00**49**	RFD13	RFD[111:104]							
$004**A**	RFD14	RFD[119:112]							
$004**B**	RFD15	RFD[127:120]							
$004**C**	Reserved	—	—	—	—	—	—	—	—
$004**D**	Reserved	—	—	—	—	—	—	—	—
$004**E**	Reserved	—	—	—	—	—	—	—	—
$004**F**	Reserved	—	—	—	—	—	—	—	—

1. Shaded bits are recommended to only be controlled by firmware or factory test.

Table 7. RFM register summary - RPAGE = 1

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$00**30**	RFCR0	BPS[7:0]							
$00**31**	RFCR1	FRM[7:0]							
$00**32**	RFCR2	SEND	RPAGE	EOM	PWR[4:0]				
$00**33**	RFCR3	DATA	IFPD	ISPC	IFID	FNUM[3:0]			
$00**34**	RFCR4	RFBT[7:0]							
$00**35**	RFCR5	BOOST	LFSR[6:0]						
$00**36**	RFCR6	VCO_GAIN[1:0]		RFFT[5:0]					
$00**37**	RFCR7	RFIF	RFEF	RFVF	RFIAK	RFIEN	RFLVDEN	RCTS	RFMRST
$00**38**	EPR	—/VCD3	PLL_LPF_[2:0]/VCD[2:0]			—	—	PA_SLOPE	VCD_EN
$00**39**	Reserved	—	—	—	—	—	—	—	—
$00**3A**	Reserved	—	—	—	—	—	—	—	—
$00**3B**	Reserved	—	—	—	—	—	—	—	—
$00**3C**	RFD0	RFD[135:128]							
$00**3D**	RFD1	RFD[143:136]							
$00**3E**	RFD2	RFD[151:144]							
$00**3F**	RFD3	RFD[159:152]							
$00**40**	RFD4	RFD[167:160]							
$00**41**	RFD5	RFD[175:168]							
$00**42**	RFD6	RFD[183:176]							
$00**43**	RFD7	RFD[191:184]							
$00**44**	RFD8	RFD[199:192]							
$00**45**	RFD9	RFD[207:200]							
$00**46**	RFD10	RFD[215:208]							
$00**47**	RFD11	RFD[223:216]							
$00**48**	RFD12	RFD[231:224]							
$00**49**	RFD13	RFD[239:232]							
$00**4A**	RFD14	RFD[247:240]							
$00**4B**	RFD15	RFD[255:248]							
$00**4C**	Reserved	—	—	—	—	—	—	—	—
$00**4D**	Reserved	—	—	—	—	—	—	—	—
$00**4E**	Reserved	—	—	—	—	—	—	—	—
$00**4F**	Reserved	—	—	—	—	—	—	—	—

1. Shaded bits are recommended to only be controlled by firmware or factory test.

4.4 High-address registers

High-page registers are used much less often, so they are located above $1800 in the memory map. This leaves more room in the direct page for more frequently used registers and variables. The registers control system level features as given in Table 8.

Table 8. MCU high-address register summary[1] [2][3]

Address	Register name	Bit 7	6	5	4	3	2	1	Bit 0
$1800	SRS	POR	PIN	COP	ILOP	ILAD	PWU	LVD	0
$1801	SBDFR	0	0	0	0	0	0	0	BDFR
$1802	SIMOPT1	COPE	COPCLKS	STOPE	RFEN	TRE	TRH	BKGDPE	1
$1803	SIMOPT2	0	COPT[2:0]			LFOSEL	TCLKDIV	BUSCLKS[1:0]	
$1804	Reserved	—	—	—	—	—	—	—	—
$1805	Reserved	—	—	—	—	—	—	—	—
$1806	SDIDH	REV[3:0]				ID[11:8]			
$1807	SDIDL	ID[7:0]							
$1808	SRTISC	RTIF	RTIACK	RTICLKS	RTIE	0	RTIS{2:0]		
$1809	SPMSC1	LVDF	LVDACK	LVDIE	LVDRE	LVDSE	LVDE	0	BGBE
$180A	SPMSC2	0	0	0	PDF	0	PPDACK	PDC	0
$180B	Reserved	—	—	—	—	—	—	—	—
$180C	SPMSC3	LVWF	LVWACK	LVDV	LVWV	0	0	0	0
$180D	SIMSES	0	0	KBF	IRQF	TRF	PWUF	LFF	RFF
$180E	SOTRM	SOTRM[7:0]							
$180F	SIMTST	—	TRH[2:0]			—	—	—	TRO
$1810-1F	Reserved	—	—	—	—	—	—	—	—
$1820	FCDIV	DIVLD	PRDIV8	DIV[5:0]					
$1821	FOPT	KEYEN	FNORED	0	0	0	0	SEC0[1:0}	
$1822	Reserved	—	—	—	—	—	—	—	—
$1823	FCNFG	0	0	KEYACC	0	0	0	0	0
$1824	FPROT	FPS[7:1]							FPDIS
$1825	FSTAT	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	0	0
$1826	FCMD	FERASE	FCMD[6:0]						
$1827-3F	Reserved	—	—	—	—	—	—	—	—

1. Reserved bits shown as 0 must always be written to 0.
2. Reserved bits shown as 1 must always be written to 1.
3. Shaded bits are recommended to only be controlled by firmware or factory test.

4.5 MCU parameter registers

The 64 bytes of parameter registers are located at addresses $0050 through $008F. These registers are powered up at all times and 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.

4.6 MCU RAM

The MPXx85/86xxD includes static RAM. The locations in RAM below $0100 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, Stop3 or Stop4 modes. At power-on or after wake-up from Stop1, the contents of RAM are not 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. See Section 4.8 for a detailed description of the security feature.

None of the RAM locations are used directly by the firmware provided by Freescale. 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 supplied by Freescale.

4.7 FLASH

The FLASH memory is intended primarily for program storage. The operating program can be loaded into the FLASH memory after final assembly of the application product using the single-wire background debug interface. Because no special voltages are needed for FLASH erase and programming operations, in-application programming is also possible through other software-controlled communication paths. For a more detailed discussion of in-circuit and in-application programming, refer to the *HCS08 Family Reference Manual, Volume I,* Freescale document order number HCS08RMV1/D.

4.7.1 Features

Features of the FLASH memory include:

- 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

4.7.2 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 9 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 $t_{FCLK} = 5$ µs. Program and erase times shown include overhead for the command state machine and enabling and disabling of program and erase voltages.

Table 9. Program and erase times

Parameter	Cycles of FCLK	Time if FCLK = 200 kHz
Byte program	9	45 µs
Byte program (burst)	4	20 ms(1)
Page erase	4000	20 ms
Mass erase	20,000	100 ms

1. Excluding start/end overhead.

4.7.3 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.

NOTE

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 (0x05), byte program (0x20), burst program (0x25), page erase (0x40), and mass erase (0x41). 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 any time 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. Figure 8 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.

4.7.4 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.

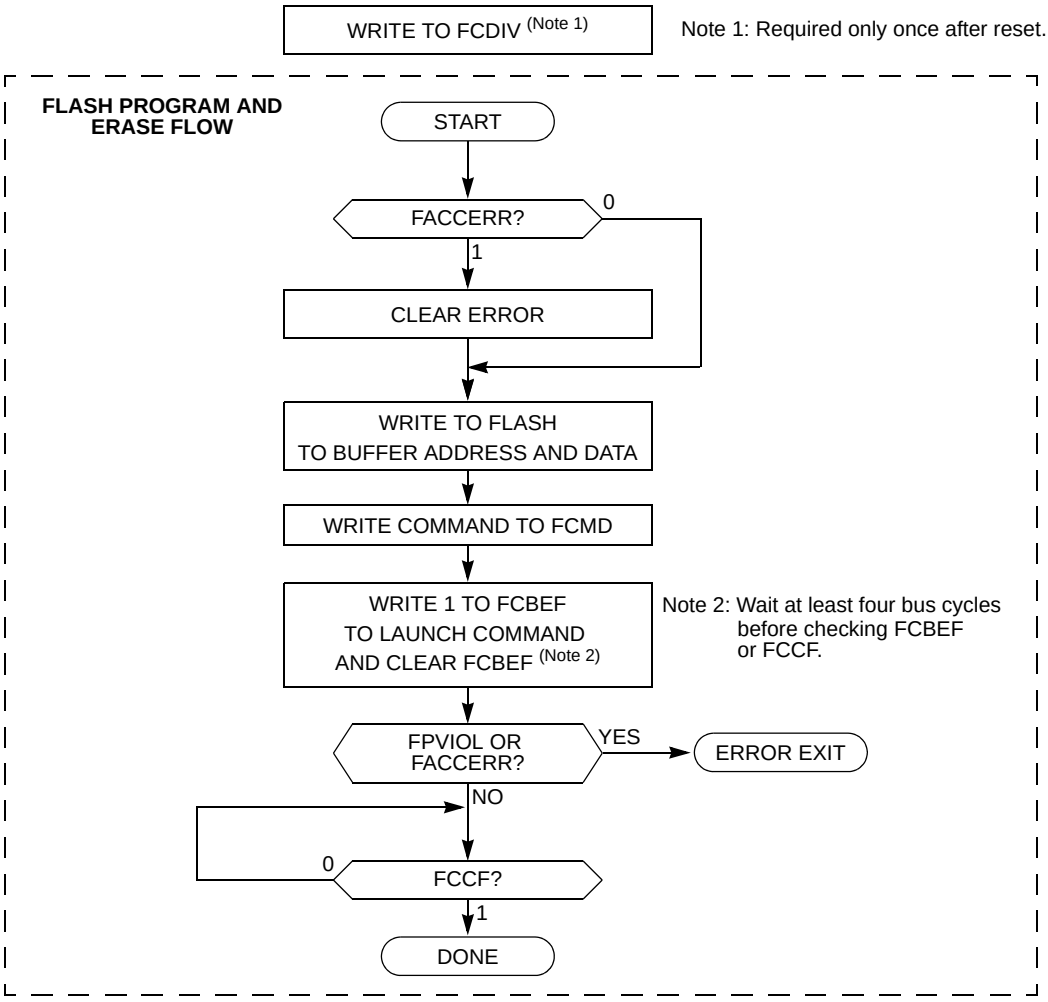


Figure 8. FLASH program and erase flowchart

NOTE

Programming time for the FLASH through the BDM function is dependent on the specific external BDM interface tool and software being used. Consult tool vendor for programming times.

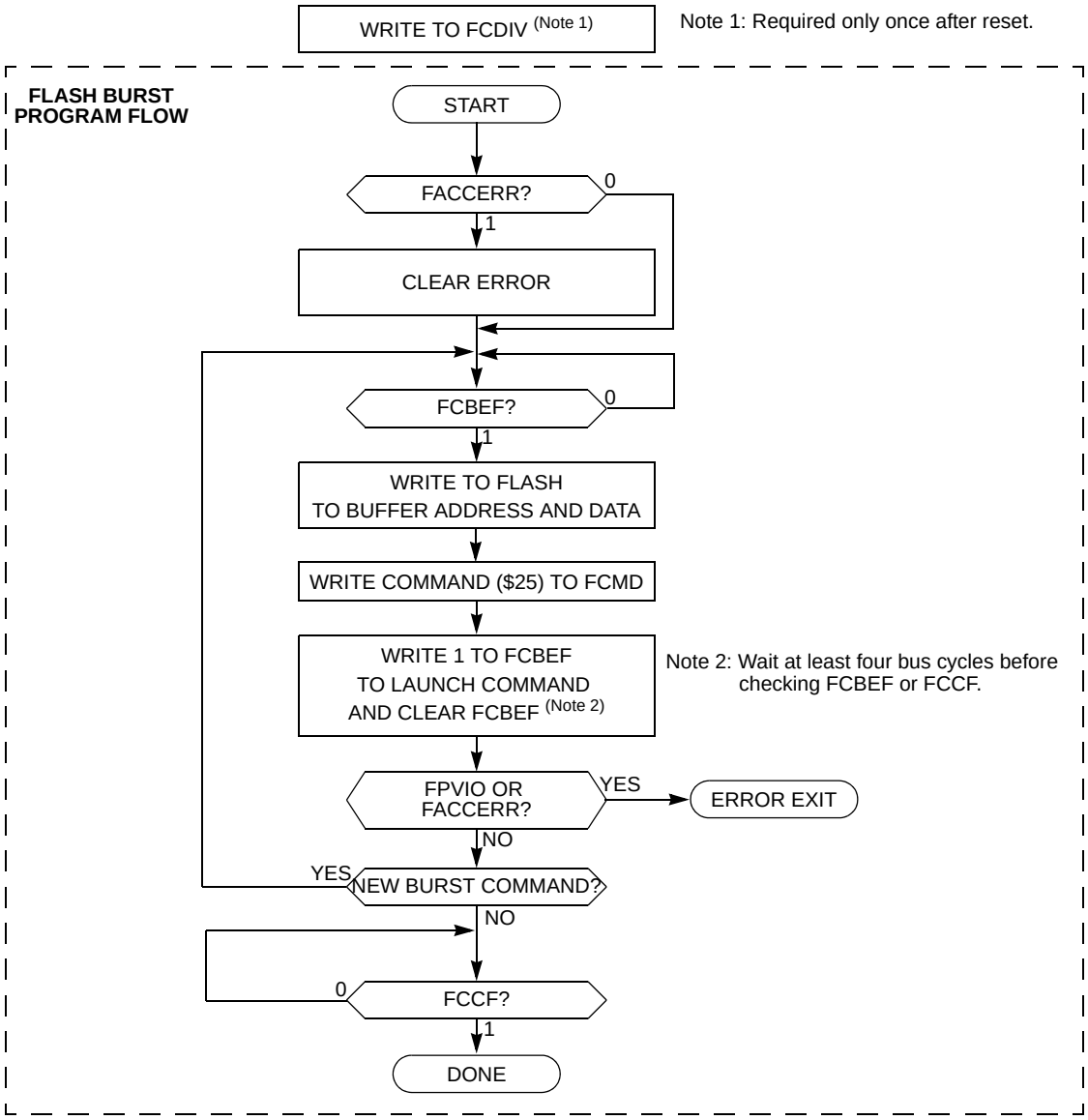


Figure 9. FLASH burst program flowchart

4.7.5 Access errors

An access error occurs whenever the command execution protocol is violated.

Any of the following specific actions will 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 (0x05, 0x20, 0x25, 0x40, or 0x41) 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 (0x20, 0x25, or 0x40) with a background debug 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.

4.7.6 FLASH 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, 0xFFFF. (see Section 4.9.4).

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 0xE000 through 0xFFFF), the FPS bits must be set to 1101 111 which results in the value 0xDFFF 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 0xDE must be programmed into NVPROT to protect addresses 0xE000 through 0xFFFF.

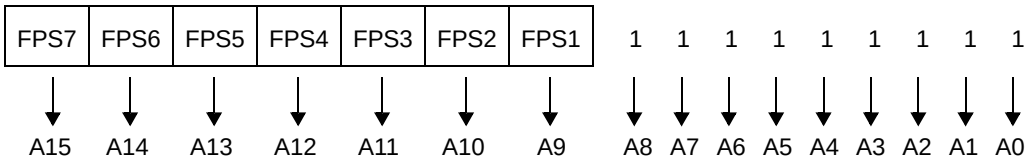


Figure 10. Block protection mechanism

One use for block protection is to block protect an area of FLASH memory for a bootloader program. This bootloader program then can be used to erase the rest of the FLASH memory and reprogram it. Because the bootloader is protected, it remains intact even if MCU power is lost in the middle of an erase and reprogram operation.

4.7.7 Vector redirection

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 the bootloader and resetting vector space. Vector redirection is enabled by programming the FNORED bit in the NVOPT register located at address 0xFFBF 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 0xFFBD. All of the interrupt vectors (memory locations 0xFFC0–0xFFFD) are redirected, though the reset vector (0xFFFE:FFFF) is not.

For example, if 512 bytes of FLASH are protected, the protected address region is from 0xFE00 through 0xFFFF. The interrupt vectors (0xFFC0–0xFFFD) are redirected to the locations 0xFDC0–0xFDFD. Now, if an SPI interrupt is taken for instance, the values in the locations 0xFDE0:FDE1 are used for the vector instead of the values in the locations 0xFFE0: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, which includes the default vector locations, unchanged.

4.8 Security

The MPXx85/86xxD includes circuitry 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.

4.9 FLASH registers and control bits

The FLASH module has nine 8-bit registers in the high-page register space, 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. Refer to Table 8 and Table 9 for the absolute address assignments for all FLASH registers. This section refers to registers and control bits only by their names. A Freescale Semiconductor-provided equate or header file normally is used to translate these names into the appropriate absolute addresses.

4.9.1 FLASH Clock Divider register (FCDIV)

Bit 7 of this register is a read-only status flag. Bits 6 through 0 can be read at any time but can be written only once. Before any erase or programming operations are possible, write to this register to set the frequency of the clock for the nonvolatile memory system within acceptable limits.

Table 10. FLASH Clock Divider register (FCDIV)

$1820	7	6	5	4	3	2	1	0
R	DIVLD	PRDIV8	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 11. FCDIV register field descriptions

Field	Description
7 DIVLD	**Divisor Loaded Status Flag** — When set, this read-only 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. 0 FCDIV has not been written since reset; erase and program operations disabled for FLASH 1 FCDIV has been written since reset; erase and program operations enabled for FLASH
6 PRDIV8	**Prescale (Divide) FLASH Clock by 8** 0 Clock input to the FLASH clock divider is the bus rate clock 1 Clock input to the FLASH clock divider is the bus rate clock divided by 8
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 — $f_{FCLK} = f_{Bus} \div ([DIV5:DIV0] + 1)$ • if PRDIV8 = 1 — $f_{FCLK} = f_{Bus} \div (8 \times ([DIV5:DIV0] + 1))$ Table 12 shows the appropriate values for PRDIV8 and DIV5:DIV0 for selected bus frequencies.

Table 12. FLASH Clock Divider settings

f_{Bus}	PRDIV8 (binary)	DIV5:DIV0 (decimal)	f_{FCLK}	Program/Erase Timing Pulse (5 µs Min, 6.7 µs Max)
20 MHz	1	12	192.3 kHz	5.2 µs
10 MHz	0	49	200 kHz	5 µs
8 MHz	0	39	200 kHz	5 µs
4 MHz	0	19	200 kHz	5 µs
2 MHz	0	9	200 kHz	5 µs
1 MHz	0	4	200 kHz	5 µs
200 kHz	0	0	200 kHz	5 µs
150 kHz	0	0	150 kHz	6.7 µs

4.9.2 FLASH Options 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 13. FLASH Options register (FOPT)

$1821	7	6	5	4	3	2	1	0
R	KEYEN	FNORED	0	0	0	0	SEC01	SEC00
W								
Reset:	This register is loaded from nonvolatile location NVOPT during reset.							

= Unimplemented or Reserved

Table 14. FOPT register field descriptions

Field	Description
7 KEYEN	**Backdoor Key Mechanism Enable** — When this 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. For more detailed information about the backdoor key mechanism, refer to Section 4.8." 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

Table 14. FOPT register field descriptions (continued)

Field	Description
6 FNORED	**Vector Redirection Disable** — When this bit is 1, then vector redirection is disabled. 0 Vector redirection enabled 1 Vector redirection disabled
1:0 SEC0[1:0]	**Security State Code** — This 2-bit field determines the security state of the MCU as shown in Table 15. 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. For more detailed information about security, refer to Section 4.8. SEC01:SEC00 changes to 1:0 after successful backdoor key entry or a successful blank check of FLASH.

Table 15. Security states

SEC01:SEC00	Description
0:0	secure
0:1	secure
1:0	unsecured
1:1	secure

4.9.3 FLASH Configuration register (FCNFG)

Bits 7 through 5 can be read or written at any time. Bits 4 through 0 always read 0 and cannot be written.

Table 16. FLASH Configuration register (FCNFG)

$1823	7	6	5	4	3	2	1	0
R	0	0	KEYACC	0	0	0	0	0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 17. FCNFG register field descriptions

Field	Description
5 KEYACC	**Enable Writing of Access Key** — This bit enables writing of the backdoor comparison key. For more detailed information about the backdoor key mechanism, refer to Section 4.8. 0 Writes to 0xFFB0–0xFFB7 are interpreted as the start of a FLASH programming or erase command 1 Writes to NVBACKKEY (0xFFB0–0xFFB7) are interpreted as comparison key writes

4.9.4 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 18. FLASH Protection register (FPROT)

$1824	7	6	5	4	3	2	1	0
R	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
W	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)
Reset:	This register is loaded from nonvolatile location NVPROT during reset.							

1. Background commands can be used to change the contents of these bits in FPROT.

Table 19. FPROT register field descriptions

Field	Description
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	**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

4.9.5 FLASH Status register (FSTAT)

Bits 3, 1, and 0 always read 0 and writes have no meaning or effect. The remaining five bits are status bits that can be read at any time. Writes to these bits have special meanings that are discussed in the bit descriptions.

Table 20. FLASH Status register (FSTAT)

$1825	7	6	5	4	3	2	1	0
R	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	0	0
W								
Reset:	1	1	0	0	0	0	0	0

= Unimplemented or Reserved

Table 21. FSTAT register field descriptions

Field	Description
7 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
6 FCCF	**FLASH Command Complete Flag** — 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
5 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 1 An attempt was made to erase or program a protected location
4 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. For a more detailed discussion of the exact actions that are considered access errors, see Section 4.7.5. FACCERR is cleared by writing a 1 to FACCERR. Writing a 0 to FACCERR has no meaning or effect. 0 No access error 1 An access error has occurred
2 FBLANK	**FLASH Verified as All Blank (erased) Flag** — 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 1 After a blank check command is completed and FCCF = 1, FBLANK = 1 indicates the FLASH array is completely erased (all 0xFF)

4.9.6 FLASH Command register (FCMD)

Only five command codes are recognized in normal user modes as shown in Table 23. Refer to Section 4.7.3, for a detailed discussion of FLASH programming and erase operations.

Table 22. FLASH Command register (FCMD)

$1826	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	FCMD7	FCMD6	FCMD5	FCMD4	FCMD3	FCMD2	FCMD1	FCMD0
Reset:	0	0	0	0	0	0	0	0

Table 23. FLASH Commands

Command	FCMD	Equate file label
Blank check	0x05	mBlank
Byte program	0x20	mByteProg
Byte program — burst mode	0x25	mBurstProg
Page erase (512 bytes/page)	0x40	mPageErase
Mass erase (all FLASH)	0x41	mMassErase

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.

5 Reset, Interrupts and System Configuration

This section discusses basic reset and interrupt mechanisms and the various sources of reset and interrupts in the MPXx85/86xxD. Some interrupt sources from peripheral modules are discussed in greater detail within other sections of this product specification. 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.

5.1 Features

Reset and interrupt features include:

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

5.2 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 MPXx85/86xxD 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.

5.3 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 SRS. This write does not affect the data in the read-only SRS. 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 24. 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 tolerances of these timeout periods is dependent on the selected clock source (LFO or HFO).

Table 24. COP watchdog timeout period

COPCLKS	COPT			Clock source	COP overflow count	COP overflow time (msec, 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 24. COP watchdog timeout period (continued)

COPCLKS	COPT			Clock source	COP overflow count	COP overflow time (msec, 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 (4MHz)
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 SRS 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, the COP timer is temporarily disabled.

5.4 SIM Test register (SIMTST)

The output of the temperature monitor is available using the SIM Test register as shown in Table 25.

Table 25. SIM Test register (SIMTST)

$180F	Bit 7	6	5	4	3	2	1	Bit 0
R	—	TRH[2:0]			—	—	—	TRO
W								—
RESET:	0	0	1	1	1	0	0	1

Table 26. SIMTST register field descriptions

Field	Description
7 reserved	**Reserved bit** — This bit is reserved for factory trim and should not be altered by the user.
6:4 TRH	**Temperature Restart High** — Temperature restart high threshold, binary coded from 0x00 to 0x07; recommend applications overwrite to 0x06 at each wake-up cycle. These bits are reserved for factory trim and should not be altered by the user.
3:1	**Reserved Bits** — These bits are reserved for factory trim and should not be altered by the user.
0 TRO	**Temperature Reset Output —** This read-only bit shows the output of the temperature reset monitor. Writes to this bit are ignored. This bit is initially set following a reset of the MCU. 0 The temperature of the device has not reached the arming temperature 1 The temperature of the device has passed the arming temperature

5.5 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.

NOTE

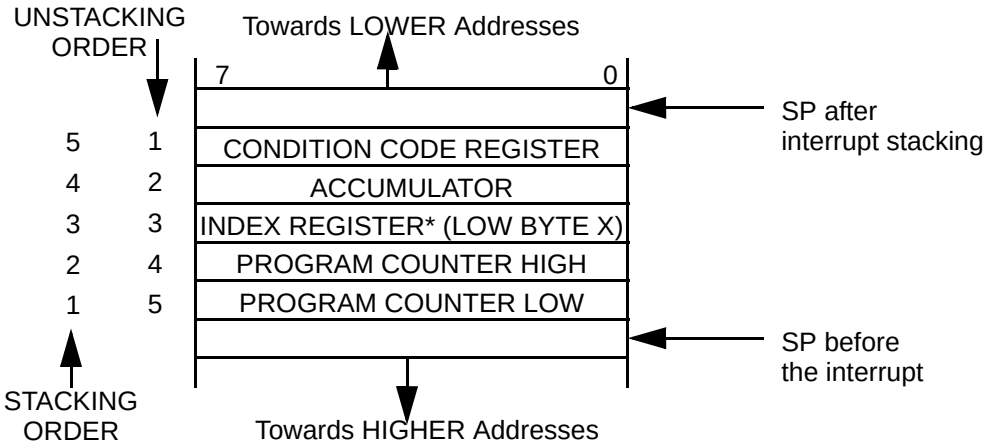
For compatibility with the M68HC08, the H register is not automatically saved and restored. It is good programming practice to push H onto the stack at the start of the interrupt service routine (ISR) and restore it just before the RTI that is used to return from the ISR.

5.5.1 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.



* High byte (H) of index register is not automatically stacked.

Figure 11. Interrupt stack frame

5.5.2 Vector summary

Table 27 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 will be described in Section 14.

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 will wake the MCU from any of the stop modes.

Table 27. Vector summary

Vector Priority	Vector No.	Jump table vector addr (high/low)	Vector name	Module source	Flags	Enables	Description
Lower	15	$DFE0 - $DFE1	Vkbi	KBI	KBF	KBIE	Keyboard interrupt pins PTA[3:0]
	14	$DFE2 - $DFE3	Reserved				
	13	$DFE4 - $DFE5	Reserved				
	12	$DFE6 - $DFE7	Vrti	Sys Ctrl	RTIF	RTIE	Interrupt from the RTI when the periodic wake-up timer has timed out.
	11	$DFE8 - $DFE9	Vlfrcvr	LFR	LFIDF	LFIDIE	Interrupt from LFR in data mode when a valid wake ID has been received.
					LFCDF	LFCDIE	Interrupt from LFR in carrier mode when a carrier present for the required time.
					LFERF	LFERIE	Interrupt from LFR in the Manchester decode mode when an error is detected.
					LFDRF	LFDRIE	Interrupt from LFR in the Manchester decode mode when an 8-bit data byte has been successfully received.
	10	$DFEA - $DFEB	Vadc1	ADC10	COCO	AIEN	Interrupt from the ADC10 when a conversion is complete.
	9	$DFEC - $DFED	Vrf	RFM	RFIF	RFIEN	Interrupt from the RFM when the data buffer has been completely sent.
					RFEF		Interrupt from the RFM when transmission error detected.
	8	$DFEE - $DFEF	Vsm	SMI	SMIF	SMIEN	Interrupt from the SMI when a measurement is complete.
	7	$DFF0 - $DFF1	Vtpm1ovf	TPM1	TOF	TOIE	Interrupt from the TPM1 when the timer overflows.
	6	$DFF2 - $DFF3	Vtpm1ch1	TPM1	CH1F	CH1IE	Interrupt from the TPM1 when the selected event for channel 1 occurs.
	5	$DFF4 - $DFF5	Vtpm1ch0	TPM1	CH0F	CH0IE	Interrupt from the TPM1 when the selected event for channel 0 occurs.
	4	$DFF6 - $DFF7	Vwuktmr	PWU	WUKI	WUK[5:0]	Interrupt from the PWU when the wake-up time interval has elapsed.
	3	$DFF8 - $DFF9	Vlvd	Sys Ctrl	LVDF	LVDIE	Interrupt from the LVD when the supply voltage has dropped below the LVD threshold.
Higher	2	$DFFA - $DFFB	Reserved				
	1	$DFFC - $DFFD	Vswi	SWI opcode	—	—	Interrupt from the CPU when an SWI instruction has been executed.
	0	$DFFE -$DFFF	Vreset	Sys Ctrl - POR	—	—	Reset from power on sequence.
				Sys Ctrl - PRF	PRF	PRST[5:0]	Reset from PWU when the reset interval elapsed.
				Sys Ctrl - COP	—	COPE	Reset when COP watchdog times out.
				Sys Ctrl - LVD	—	LVDRE	Reset from the LVD when the supply voltage has dropped below the LVD threshold.
				Temp Restart	—	TRE	Reset when the temperature falls below the temperature restart threshold
				Illegal opcode	—	—	Reset from the CPU when trying to execute an illegal opcode.
				Illegal address	—	—	Reset from the CPU when trying to access an illegal address.

5.6 Low-voltage detect (LVD) system

The MPXx85/86xxD 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.

5.6.1 Power-on reset operation

When power is initially applied to the MPXx85/86xxD, or when the supply voltage drops below the V_{POR} level, the POR circuit will cause a reset condition. As the supply voltage rises, the LVD circuit will hold 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 SRS are set following a POR.

5.6.2 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 MPXx85/86xxD 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 SRS register is set following either an LVD reset or POR.

5.6.3 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 will be set and an LVD interrupt will occur.

5.6.4 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.

5.7 System clock control

Several clock rate selections are possible with the MPXx85/86xxD using the BUSCLKS[1:0] control bits to select the clock frequency division of the HFO as given in Table 28. These bits are cleared by any MCU reset.

Table 28. 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

5.8 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 29.

Table 29. 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

5.9 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 Figure 30.

Table 30. RTI Status/Control register (SRTISC)

$1808	Bit 7	6	5	4	3	2	1	Bit 0
R	RTIF	0	RTICLKS	RTIE	0	RTIS[2:0]		
W	—	RTIACK						
RESET:	0	0	0	0	0	0	0	0
POR:	0	0	0	0	0	0	0	0

Table 31. SRTISC register field descriptions

Field	Description
7 RTIF	**RTI interrupt flag** — The RTIF bit indicates when a wake-up interrupt has been generated by the RTI. This bit is cleared by writing a one to the RTIACK bit. Writing a zero to this bit has no effect. Reset clears this bit. 0 Wake-up interrupt not generated or was previously acknowledged. 1 Wake-up interrupt generated.
6 RTIACK	**Acknowledge RTIF interrupt flag** — The RTIACK bit clears the RTIF bit if written with a one. Writing a zero to the RTIACK bit has no effect on the RTIF bit. Reading the RTIACK bit returns a zero. Reset has no effect on this bit. 0 No effect. 1 Clear RTIF bit.
5 RTICLKS	**RTI interrupt clock select** — This read-write bit selects the clock source for the real-time interrupt request 0 Real-time interrupt request clock source is the LFO. 1 Real-time interrupt request clock source is the HFO (MCU must be in the Run mode).
4 RTIE	**RTIF interrupt enable** — The RTIE bit enables RTI interrupts if written with a one. Reset clears this bit. 0 Disable RTI interrupts. 1 Enable RTI interrupts.
3 Unused	**Unused**
2:0 RTIS[2:0]	**RTI interrupt delay selects** — The RTIS[2:0] bits select the timing of the RTI interrupts as given in Table 32. Reset clears these bits.

Table 32. Real-time interrupt period

RTIS2	RTIS1	RTIS0	Delay timing (msec) (dependent on 1 kHz LFO)
0	0	0	OFF
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

5.10 Temperature sensor and restart system

The MPXx85/86xxD has two temperature sensing mechanisms. The first is an accurate sensor which is accessible through the ADC10 channel 1. The second is a less accurate, very low power sensor which generates a wake-up from Stop1 when the temperature crosses its threshold of detection. This is the temperature restart wake-up which is used as follows:

1. The temperature restart wake-up is enabled by software following detection of an over temperature condition using the temperature sensor connected to the ADC10.
2. User software enables the temperature restart detector and then instructs the MCU to enter Stop1 mode to halt execution during the out-of-range temperature condition.
3. When the temperature crosses the temperature restart threshold back into the normal range of operation, a wake-up is generated to wake the MCU. Exit from Stop1 will reset the device.

The temperature sensor is enabled whenever the ADC10 is enabled. The temperature restart wake-up is enabled by setting the TRE bit in SIMOPT1 register and whether the detector interrupts the MCU from a very low or a very high temperature is determined by the TRH bit in the SIMOPT1 register.

5.11 Reset, Interrupt and System control registers and bits

One 8-bit register in the direct page register space and eight 8-bit registers in the high-page register space are related to reset and interrupt systems.

5.11.1 System Reset Status register (SRS)

The SRS register at $1800 includes seven read-only status flags to indicate the source of the most recent reset. When a debug host forces reset by writing 1 to BDFR in the SBDFR register, none of the status bits in SRS will be set. Writing any value to this register address clears the COP watchdog timer without affecting the contents of this register. The reset state of these bits depends on what caused the MCU to reset.

Table 33. System Reset Status register (SRS)

$1800	Bit 7	6	5	4	3	2	1	Bit 0
R	POR	PIN	COP	ILOP	ILAD	PWU	LVD	0
W								—
POR Reset:	1	0	0	0	0	0	1	0
LVD Reset:	1	0	0	0	0	0	1	0
Any Other Reset:	0	(1)	(1)	(1)	(1)	0	0	0

1. Any of these reset sources that are active at the time of reset will cause the corresponding bit(s) to be set; bits corresponding to sources that are not active at the time of reset will be cleared.

Table 34. SRS register field descriptions

Field	Description
7 POR	**Power-on reset —** This bit indicates reset was caused by the power-on detection logic. Because the internal supply voltage was ramping up at the time, the low-voltage reset (LVR) status bit is also set to indicate that the reset occurred while the internal supply was below the LVR threshold. 0 Reset not caused by POR 1 POR caused reset
6 PIN	**External reset pin —** This bit indicates reset was caused by an active-low level on the external reset pin if the device was in either the Stop1 or run modes. This bit is not set if the external reset pin is pulled low when the device is in the Stop1 mode. 0 Reset not caused by external reset pin 1 Reset came from external reset pin
5 COP	**Computer operating properly (COP) watchdog —** This bit indicates that reset was caused by the COP watchdog timer timing out. This reset source may be blocked by COPE = 0. 0 Reset not caused by COP timeout 1 Reset caused by COP timeout

Table 34. SRS register field descriptions (continued)

Field	Description
4 ILOP	**Illegal opcode —** This bit indicates reset was caused by an attempt to execute an unimplemented or illegal opcode. The STOP instruction is considered illegal if stop is disabled by STOPE = 0 in the SOPT register. The BGND instruction is considered illegal if active background mode is disabled by ENBDM = 0 in the BDCSC register. 0 Reset not caused by an illegal opcode 1 Reset caused by an illegal opcode
3 ILAD	**Illegal address —** This bit indicates reset was caused by an attempt to access either data or an instruction at an unimplemented memory address. 0 Reset not caused by an illegal address 1 Reset caused by an illegal address
2 PWU	**Programmable wake-up —** This bit indicates reset was caused by a PWU reset in run, wait, Stop4, and Stop3. After Stop1 exit, PRF in PWUCSI indicates PWU was the source of a wake-up. 0 Reset not caused by PWU. 1 Reset caused by PWU.
1 LVD	**Low voltage detect —** If the LVDRE and LVDSE bits are set and the supply drops below the LVD trip voltage, an LVD reset will occur. This bit is also set by POR. 0 Reset not caused by LVD trip or POR. 1 Reset caused by LVD trip or POR.
0 Unused	**Unused bit —** This bit always reads as a logical zero. Writes

5.11.2 System Options 1 register (SIMOPT1)

The following clock source and frequency selections are available using the system option register 1 as shown in Figure 35.

Table 35. System Option 1 register 1 (SIMOPT1)

$1802	Bit 7	6	5	4	3	2	1	Bit 0
R W	COPE	COPCLKS	STOPE	RFEN	TRE	TRH	BKGDPE	—
RESET:	1	0	0	-	0	0	1	1

Table 36. SIMOPT1 register field descriptions

Field	Description
7 COPE	**COP enable**— This control bit enables the COP watchdog. This bit is a write-once bit so that only the first write after reset is honored. Reset sets the COPE bit. 0 COP Watchdog disabled. 1 COP Watchdog enabled.
6 COPCLKS	**COP clock select**— This control bit selects the clock source for the COP watchdog timer. This bit is a write-once bit so that only the first write after reset is honored. This bit is cleared by an MCU reset. 0 Select the LFO oscillator output. 1 Select the CPU bus clock.
5 STOPE	**Stop mode select**— This control bit enables/disables the STOP instruction to enter a stop mode defined by the SPMSCR2 register. This bit is a write-once bit so that only the first write after reset is honored. This bit is cleared by an MCU reset. 0 Disable stop modes. 1 Enable stop modes.
4 RFEN	**RF module enable** — This bit enables or disables the RF module. This bit is not affected by any reset or power on after stop exit. It is only initialized at the first power up. This bit can be written anytime. 1 RF module enabled. 0 RF module disabled.
3 TRE	**Temperature restart enable**— This control bit enables the temperature restart circuit to interrupt the MCU after being shutdown at either a very high or very low temperature. This bit is cleared by an MCU reset. 0 Temperature restart disabled. 1 Temperature restart enabled.

Table 36. SIMOPT1 register field descriptions (continued)

Field	Description
2 TRH	**Temperature restart level** — This control bit selects whether the temperature restart circuit will interrupt the MCU after being shutdown on returning from either a very high or very low temperature. This bit is cleared by an MCU reset. 0 Temperature restart interrupts MCU on return from a very low temperature. 1 Temperature restart interrupts MCU on return from a very high temperature.
1 BKGDPE	**BKGD pin enable** — 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
0 Reserved	Reserved register bit, always reads 1.

5.11.3 System Operation 2 register (SIMOPT2)

The following clock source and frequency selections are available using the system option register 2 as shown in Figure 37.

Table 37. System Option 2 register (SIMOPT2)

$1803	Bit 7	6	5	4	3	2	1	Bit 0
R W	0	COPT[2:0]			LFOSEL	TCLKDIV	BUSCLKS[1:0]	
RESET:	0	1	1	1	0	0	0	0

Table 38. SIMOPT2 register field descriptions

Field	Description
7 Unused	**Unused bit** — This bit is unused and reads as a logic zero.
6:4 COPT[2:0]	**COP watchdog time out** — These control bits select the timeout period for the COP watchdog timer as given in Table 24. These bits are set by an MCU reset to select the longest watchdog timeout period. These bits are write-once after power up.
3 LFOSEL	**TPM1 channel 0 clock source** — This bit determines which signal is connected to the TPM1 Channel 0, see Section 9. 0 Select clock input driven by PTA2. 1 Select clock input driven by the LFO.
2 TCLKDIV	**TPM1 channel 0 clock source divider** — The divider for the clock Source for TPM1 Channel 0, see Section 9. 0 Select RFM Dx clock source divided by 1. 1 Select RFM Dx clock source divided by 8.
1:0 BUSCLKS [1:0]	**Bus clock select — Bus clock frequency selection by changing HFO FLL ratio as shown in Figure 2. The bus clock frequency is always the HFO frequency divided by two. These bits are cleared by a reset and can be written at any time.** 00 Bus Frequency = 4 MHz (HFO = 8 MHz) 01 Bus Frequency = 2 MHz (HFO = 4 MHz) 10 Bus Frequency = 1 MHz (HFO = 2 MHz) 11 Bus Frequency = 0.5 MHz (HFO = 1 MHz)

5.11.4 System Power Management Status and Control 1 register (SPMSC1)

Table 39. System Power Management Status and Control 1 register (SPMSC1)

$1809	7	6	5	4	3	2	1[1]	0
R	LVDF	0	LVDIE	LVDRE[2]	LVDSE	LVDE[2]	0	BGBE
W		LVDACK						
Reset:	0	0	0	1	1	1	0	0

= Unimplemented or Reserved

1. Bit 1 is a reserved bit that must always be written to 0.
2. This bit can be written only one time after reset. Additional writes are ignored.

Table 40. SPMSC1 Register Field Descriptions

Field	Description
7 LVDF	**Low-voltage detect flag** — Provided LVDE = 1, this read-only status bit indicates a low-voltage detect event.
6 LVDACK	**Low-voltage detect acknowledge** — This write-only bit is used to acknowledge low voltage detection errors (write 1 to clear LVDF). Reads always return logic 0.
5 LVDIE	**Low-voltage detect interrupt enable** — This read/write bit enables hardware interrupt requests for LVDF. 0 Hardware interrupt disabled (use polling) 1 Request a hardware interrupt when LVDF = 1
4 LVDRE	**Low-voltage detect reset enable** — This read/write bit enables LVDF events to generate a hardware reset (provided LVDE = 1). 0 LVDF does not generate hardware resets 1 Force an MCU reset when LVDF = 1
3 LVDSE	**Low-voltage detect stop enable** — Provided LVDE = 1, this read/write bit determines whether the low-voltage detect function operates when the MCU is in stop mode. 0 Low-voltage detect disabled during stop mode 1 Low-voltage detect enabled during stop mode
2 LVDE	**Low-voltage detect enable** — This read/write bit enables low-voltage detect logic and qualifies the operation of other bits in this register. 0 LVD logic disabled 1 LVD logic enabled
0 Reserved	**Reserved bit** — This bit is reserved should not be altered by the user. Any read returns a logical zero. Any write should be a logical zero.
0 BGBE	**Bandgap buffer enable** — The BGBE bit is used to enable an internal buffer for the bandgap voltage reference for use by the ADC module on one of its internal channels. 0 Bandgap buffer disabled 1 Bandgap buffer enabled

5.11.5 System Power Management Status and Control 2 register (SPMSC2)

This register is used to configure the stop mode behavior of the MCU.

Table 41. System Power Management Status and Control 2 register (SPMSC2)

$180A	7	6	5	4	3	2	1	0
R	0	0	0	PDF	0	0	PDC(1)	0
W	—	—	—		—	PPDACK		
Power-on reset:	0	0	0	0	0	0	0	0
Any other reset:	0	0	U	U	0	0	0	0

= Unimplemented or Reserved U = Unaffected by reset

1. This bit can be written only one time after reset. Additional writes are ignored.

Table 42. SPMSC2 register field descriptions

Field	Description
7:5 Reserved	**Reserved bits** — These bits are reserved should not be altered by the user. Any read returns a logical zero.
4 PDF	**Power down flag** — This read-only status bit indicates the MCU has recovered from Stop1 mode. 0 MCU has not recovered from Stop1 mode 1 MCU recovered from Stop1 mode
3 Reserved	**Reserved bit** — This bit is reserved should not be altered by the user. Any read returns a logical zero.
2 PPDACK	**Partial power down acknowledge** — Writing a logic 1 to PPDACK clears the PDF bit.

Table 42. SPMSC2 register field descriptions (continued)

Field	Description
1 PDC	**Power down control** — The PDC bit controls entry into the power down (Stop1) mode 0 Power down mode are disabled 1 Power down mode are enabled
0 Reserved	**Reserved bit** — This bit is reserved should not be altered by the user. Any read returns a logical zero. Any write should be a logical zero.

5.11.6 System Power Management Status and Control 3 register (SPMSC3)

Table 43. System Power Management Status and Control 3 register (SPMSC3)

$180C	7	6	5	4	3	2	1	0
R	LVWF	0	LVDV	LVWV	0	0	0	0
W		LVWACK						
Power-on reset:	0(1)	0	0	0	0	0	0	0
LVD reset:	0(1)	0	U	U	0	0	0	0
Any other reset:	0(1)	0	U	U	0	0	0	0
		= Unimplemented or Reserved				U = Unaffected by reset		

1. LVWF will be set in the case when V_{Supply} transitions below the trip point or after reset and V_{Supply} is already below V_{LVW}.

Table 44. SRTISC register field descriptions

Field	Description
7 LVWF	**Low-voltage warning flag** — The LVWF bit indicates the low voltage warning status. 0 Low-voltage warning **not** present 1 Low-voltage warning is present or was present
6 LVWACK	**Low-voltage warning acknowledge** — The LVWF bit indicates the low voltage warning status. Writing a logic 1 to LVWACK clears LVWF to a logic 0 if a low voltage warning is not present.
5 LVDV	**Low-voltage detect voltage select** — The LVDV bit selects the LVD trip point voltage (V_{LVD}). 0 Low trip point selected ($V_{LVD} = V_{LVDL}$) 1 High trip point selected ($V_{LVD} = V_{LVDH}$)
4 LVWV	**Low-voltage warning voltage select** — The LVWV bit selects the LVW trip point voltage (V_{LVW}). 0 Low trip point selected ($V_{LVW} = V_{LVDL}$) 1 High trip point selected ($V_{LVW} = V_{LVDH}$)
3:0 Reserved	**Reserved bits** — These bits are reserved should not be altered by the user. Any read returns a logical zero.

5.12 System Stop Exit Status register (SIMSES)

The SIMSES register at $180D can be used to determine the source of an MCU wake-up from the stop modes. The flags are as shown in Table 45. All of the flags are automatically cleared when the MCU goes into a stop mode. Writes to any of these bits are ignored.

Table 45. SIM Stop Exit Status (SIMSES)

$180D	Bit 7	6	5	4	3	2	1	Bit 0
R	0	0	KBF	IRQF	TRF	PWUF	LFF	RFF
W	—	—	—	—	—	—	—	—
RESET:	0	0	0	0	0	0	0	0

Table 46. SIMSES register field descriptions

Field	Description
7:6 Unused	**Unused bits** — These bit are unused and reads as a logic zero.
5 KBF	**Keyboard flag —** This bit indicates that any keyboard pin caused the last exit from stop mode. 0 Keyboard pin did not cause the last exit from stop mode 1 Keyboard pin caused the last exit from stop mode
4 IRQF	**IRQ flag —** This bit indicates that IRQ pin caused the last exit from stop mode. 0 IRQ pin did not cause the last exit from stop mode 1 IRQ pin caused the last exit from stop mode
3 TRF	**Temperature restart flag —** This bit indicates that the temperature restart module caused the last exit from stop mode. 0 TR module did not cause the last exit from stop mode 1 TR module caused the last exit from stop mode
2 PWUF	**PWU Flag —** This bit indicates that the PWU module caused the last exit from stop mode. 0 PWU module did not cause the last exit from stop mode 1 PWU module caused the last exit from stop mode
1 LFF	**LFR Flag —** This bit indicates that the LFR module caused the last exit from stop mode. 0 LFR module did not cause the last exit from stop mode 1 LFR module caused the last exit from stop mode
0 RFF	**RFM Flag —** This bit indicates that the RFM module caused the last exit from stop mode. 0 RFM module did not cause the last exit from stop mode 1 RFM module caused the last exit from stop mode

6 General Purpose I/O

This section explains software controls related to general purpose input/output (I/O) and pin control. The MPXx85/86xxD has six general-purpose I/O pins which are comprised of a general use 4-bit port A and a 2-bit port B.

All of these 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 by the general purpose I/O. All of the general purpose I/O are configured as inputs (PTxDDn = 0) with pullup devices disabled (PTxPEn = 0).

NOTE

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 6.1).

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 12.

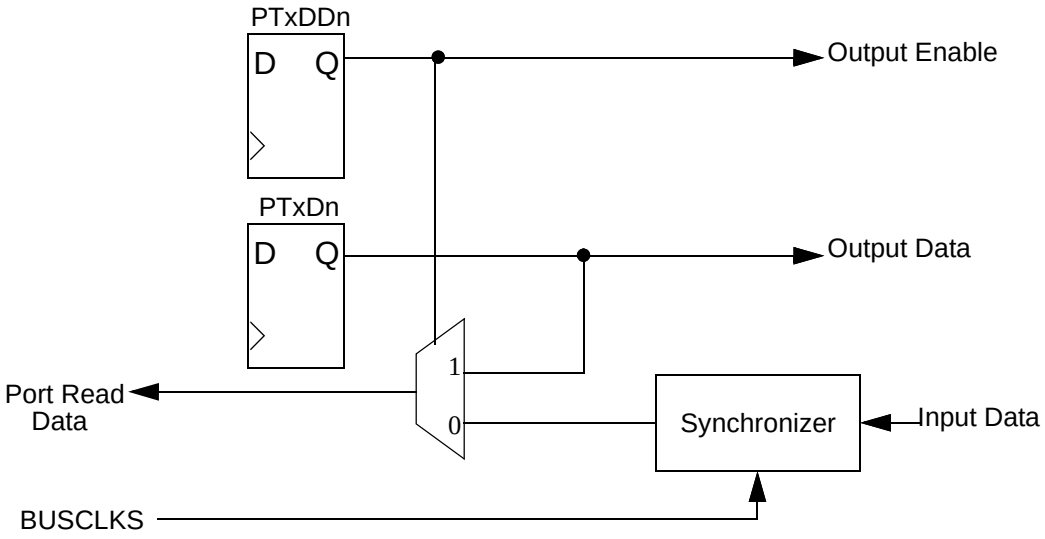


Figure 12. General purpose I/O block diagram

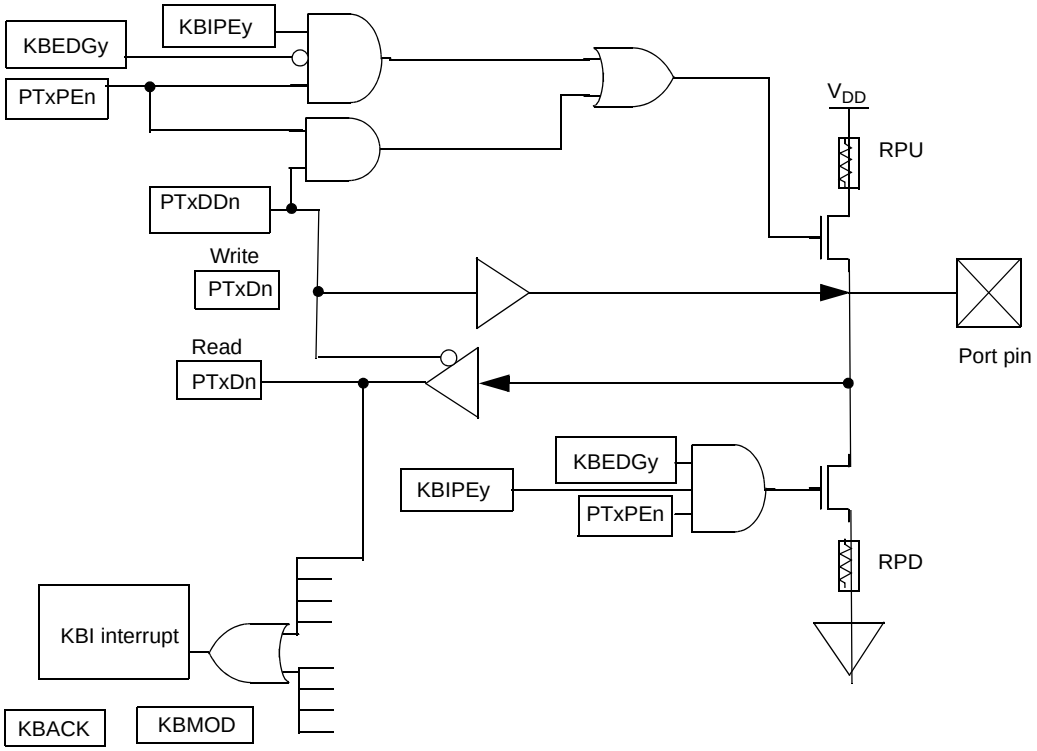


Figure 13. General purpose I/O logic

Table 47. Truth table for pullup and pulldown resistors

PTxPEn (pull enable)	PTxDDn (data direction)	KBIPEy (KBI pin enable)	KBEDGy (KBI edge select)	Pullup	Pulldown
0	0	x	x	disabled	disabled
1	0	0	x	enabled	disabled
X	1	X	x	disabled	disabled
1	0	1	0	enabled	disabled
1	0	1	1	disabled	enabled

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.

6.1 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}.

NOTE

When the device comes out of the reset state the Freescale 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 =1) with the internal pullup activated (PTxPEn =1) and leave the pin disconnected.
2. Configure the general purpose I/O pin as an input (PTxDDn =1) with the pin connected to the V_{DD} source.
3. Configure the general purpose I/O pin as an output (PTxDDn =0) and drive the pin low (PTxDn =0) and leave the pin disconnected.

CAUTION

Avoid externally connecting any general purpose I/O pin directly to AV_{DD}, V_{DD}, AV_{SS}, V_{SS} or RV_{SS} to prevent software code faults from causing excessive supply current states should these pins become outputs.

6.2 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.

6.3 General purpose I/O registers

This section provides information about the registers associated with the general purpose I/O ports and pin control functions. These general purpose I/O registers are located in page zero of the memory map and the pin control registers are located in the high page register section of memory.

6.4 Port A registers

Port A general purpose I/O function is controlled by the registers described in this section.

Table 48. Port A Data register (PTAD)

$0000	Bit 7	6	5	4	3	2	1	Bit 0
R W	—	—	—	—	PTAD[3:0]			
Reset:	0	0	0	0	0	0	0	0

Table 49. Port A Data register field descriptions

Field	Description
3:0 PTAD[3:0]	**Port A Data Register Bit** — 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.

Table 50. Internal Pullup Enable for Port A register (PTAPE)

$0001	Bit 7	6	5	4	3	2	1	Bit 0
R W	—	—	—	—	PTAPE[3:0]			
Reset:	0	0	0	0	0	0	0	0

Table 51. Port A Register Pullup Enable register field descriptions

Field	Description
3:0 PTAPE [3:0]	**Internal Pullup Enable for Port A Bit n** — Each of these control bits determines if the internal pullup device is enabled for the associated PTA pin. For port A pins that are configured as outputs, these bits have no effect and the internal pullup devices are disabled. 0 Internal pullup device disabled for port A bit n. 1 Internal pullup device enabled for port A bit n.

Table 52. Data Direction for Port A register (PTAD)

$0003	Bit 7	6	5	4	3	2	1	Bit 0
R W	—	—	—	—	PTADD[3:0]			
Reset:	0	0	0	0	0	0	0	0

Table 53. Port A Data Direction field descriptions

Field	Description
3:0 PTADD[3:0]	**Data Direction for Port A Bit n** — These read/write bits control the direction of port A pins and what is read for PTAD reads. 0 Input (output driver disabled) and reads return the pin value. 1 Output driver enabled for port A bit n and PTAD reads return the contents of PTADn.

6.5 Port B registers

Port B general purpose I/O function is controlled by the registers described in this section. The contents of the Port B registers are ignored if the LFR is enabled by setting the LFEN bit in the LFCTL1 register (see Section 12).

Table 54. Port B Data register (PTBD)

$0004	Bit 7	6	5	4	3	2	1	Bit 0
R W	—	—	—	—	—	—	PTBD[1:0]	
Reset:	0	0	0	0	0	0	0	0

Table 55. Port B Data register field descriptions

Field	Description
1:0 PTBD[1:0]	**Port B Data Register Bit n** — 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.

Table 56. Internal Pullup Enable for Port B register (PTBPE)

$0005	Bit 7	6	5	4	3	2	1	Bit 0
R W	—	—	—	—	—	—	PTBPE[1:0]	
Reset:	0	0	0	0	0	0	0	0

Table 57. Port B Register Pullup Enable field descriptions

Field	Description
1:0 PTBPE[1:0]	**Internal Pullup Enable for Port B Bit n** — Each of these control bits determines if 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. 0 Internal pullup device disabled for port B bit n. 1 Internal pullup device enabled for port B bit n.

Table 58. Data Direction for Port B register (PTBD)

$0007	Bit 7	6	5	4	3	2	1	Bit 0
R W	—	—	—	—	—	—	PTBDD[1:0]	
Reset:	0	0	0	0	0	0	0	0

Table 59. Port B Data Direction field descriptions

Field	Description
1:0 PTBDD[1:0]	**Data Direction for Port B Bit n** — These read/write bits control the direction of port B pins and what is read for PTBD reads. 0 Input (output driver disabled) and reads return the pin value. 1 Output driver enabled for port B bit n and PTBD reads return the contents of PTBDn.

7 Keyboard Interrupt

The MPXx85/86xxD has a KBI module with general purpose I/O pins.

7.1 Features

The KBI features include:

- Up to two keyboard interrupt pins with individual pin enable bits.
- Each keyboard interrupt pin is programmable as falling edge (or rising edge) only, or both falling edge and low level (or both rising edge and high level) interrupt sensitivity.
- One software enabled keyboard interrupt.
- Exit from low-power modes.

7.2 Modes of operation

This section defines the KBI operation in wait, stop, and background debug modes.

7.2.1 KBI in stop modes

The KBI operates asynchronously in Stop4 mode if enabled before executing the STOP instruction. Therefore, an enabled KBI pin (KBPEx = 1) can be used to bring the MCU out of Stop4 mode if the KBI interrupt is enabled (KBIE = 1).

During Stop1 mode, the KBI is disabled. In some systems, the pins associated with the KBI may be sources of wake-up from Stop1, see the stop modes section in the Section 3. Upon wake-up from Stop1 mode, the KBI module will be in the reset state.

7.2.2 KBI in Active Background mode

When the microcontroller is in active background mode, the KBI will continue to operate normally.

7.3 Block diagram

The block diagram for the keyboard interrupt module is shown Figure 14.

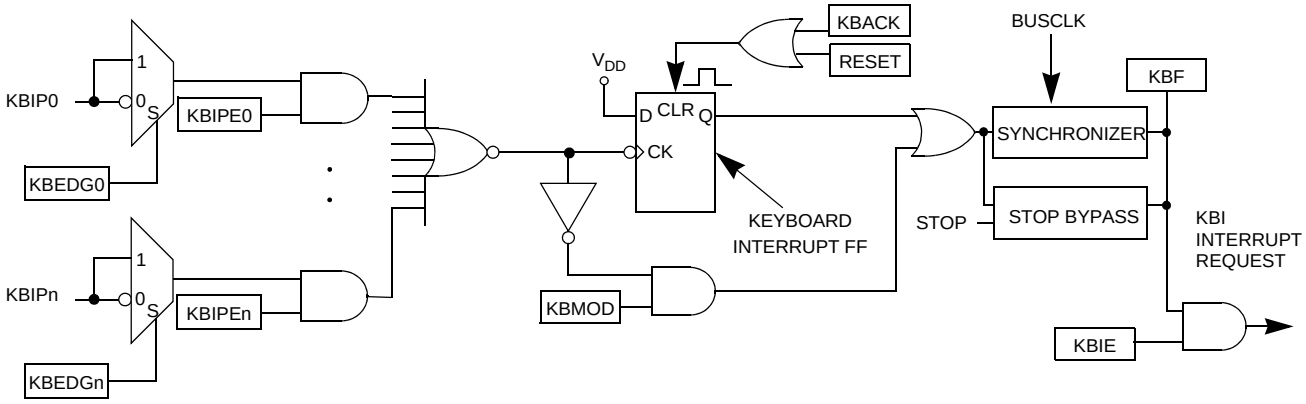


Figure 14. KBI block diagram

7.4 External signal description

The KBI input pins can be used to detect either falling edges, or both falling edge and low level interrupt requests. The KBI input pins can also be used to detect either rising edges, or both rising edge and high level interrupt requests.

The signal properties of KBI are shown in Table 60.

Table 60. Signal properties

Signal	Function	I/O
KBIPn	Keyboard interrupt pins	I

7.5 Register definition

The KBI includes three registers:

- An 8-bit pin status and control register.
- An 8-bit pin enable register.
- An 8-bit edge select register.

7.5.1 KBI Status and Control register (KBISC)

KBISC contains the status flag and control bits, which are used to configure the KBI.

Table 61. KBI Status and Control register

$000C	7	6	5	4	3	2	1	0
R	0	0	0	0	KBF	0	KBIE	KBMOD
W						KBACK		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Table 62. KBISC register field descriptions

Field	Description
7:4	Unused register bits, always read 0.
3 KBF	**Keyboard Interrupt Flag** — KBF indicates when a keyboard interrupt is detected. Writes have no effect on KBF. 0 No keyboard interrupt detected. 1 Keyboard interrupt detected.
2 KBACK	**Keyboard Acknowledge** — Writing a 1 to KBACK is part of the flag clearing mechanism. KBACK always reads as 0.
1 KBIE	**Keyboard Interrupt Enable** — KBIE determines whether a keyboard interrupt is requested. 0 Keyboard interrupt request not enabled. 1 Keyboard interrupt request enabled.
0 KBMOD	**Keyboard Detection Mode** — KBMOD (along with the KBEDG bits) controls the detection mode of the keyboard interrupt pins. 0 Keyboard detects edges only. 1 Keyboard detects both edges and levels.

7.5.2 KBI Pin Enable register (KBIPE)

KBIPE contains the pin enable control bits.

Table 63. KBI Pin Enable Register

$000D	7	6	5	4	3	2	1	0
R W	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
Reset:	0	0	0	0	0	0	0	0

Table 64. KBIPE register field descriptions

Field	Description
7:0 KBIPEn	**Keyboard Pin Enables** — Each of the KBIPEn bits enable the corresponding keyboard interrupt pin. 0 Pin not enabled as keyboard interrupt. 1 Pin enabled as keyboard interrupt.

7.5.3 KBI Edge Select register (KBIES)

KBIES contains the edge select control bits.

Table 65. KBI Edge Select register

$000E	7	6	5	4	3	2	1	0
R W	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBEDG3	KBEDG2	KBEDG1	KBEDG0
Reset:	0	0	0	0	0	0	0	0

Table 66. KBIES register Field descriptions

Field	Description
7:0 KBEDGn	**Keyboard Edge Selects** — Each of the KBEDGn bits selects the falling edge/low level or rising edge/high level function of the corresponding pin). 0 Falling edge/low level. 1 Rising edge/high level.

7.6 Functional description

This on-chip peripheral module is called a keyboard interrupt (KBI) module because originally it was designed to simplify the connection and use of row-column matrices of keyboard switches. However, these inputs are also useful as extra external interrupt inputs and as an external means of waking the MCU from stop or wait low-power modes.

The KBI module allows up to eight pins to act as additional interrupt sources. Writing to the KBIPEn bits in the keyboard interrupt pin enable register (KBIPE) independently enables or disables each KBI pin. Each KBI pin can be configured as edge sensitive or edge and level sensitive based on the KBMOD bit in the keyboard interrupt status and control register (KBISC). Edge sensitive can be software programmed to be either falling or rising; the level can be either low or high. The polarity of the edge or edge and level sensitivity is selected using the KBEDGn bits in the keyboard interrupt edge select register (KBIES).

Synchronous logic is used to detect edges. Prior to detecting an edge, enabled keyboard inputs must be at the reset logic level. A falling edge is detected when an enabled keyboard input signal is seen as a logic 1 (the reset level) during one bus cycle and then a logic 0 (the asserted level) during the next cycle. A rising edge is detected when the input signal is seen as a logic 0 during one bus cycle and then a logic 1 during the next cycle.

7.6.1 Edge-only sensitivity

A valid edge on an enabled KBI pin will set KBF in KBISC. If KBIE in KBISC is set, an interrupt request will be presented to the CPU. Clearing of KBF is accomplished by writing a 1 to KBACK in KBISC.

7.6.2 Edge and level sensitivity

A valid edge or level on an enabled KBI pin will set KBF in KBISC. If KBIE in KBISC is set, an interrupt request will be presented to the CPU. Clearing of KBF is accomplished by writing a 1 to KBACK in KBISC provided all enabled keyboard inputs are at their reset levels. KBF will remain set if any enabled KBI pin is asserted while attempting to clear by writing a 1 to KBACK.

7.6.3 KBI pullup/pulldown resistors

The KBI pins can be configured to use an internal pullup/pulldown resistor using the associated I/O port pullup enable register. If an internal resistor is enabled, the KBIES register is used to select whether the resistor is a pullup (KBEDGn = 0) or a pulldown (KBEDGn = 1).

7.6.4 KBI initialization

When a keyboard interrupt pin is first enabled it is possible to get a false keyboard interrupt flag. To prevent a false interrupt request during keyboard initialization, the user should do the following:

1. Mask keyboard interrupts by clearing KBIE in KBISC.
2. Enable the KBI polarity by setting the appropriate KBEDGn bits in KBIES.
3. If using internal pullup/pulldown device, configure the associated pullup enable bits in PTxPE.
4. Enable the KBI pins by setting the appropriate KBIPEn bits in KBIPE.
5. Write to KBACK in KBISC to clear any false interrupts.
6. Set KBIE in KBISC to enable interrupts.

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,* Freescale 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).

8.1.1 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-Kbyte address space
- 16-bit stack pointer (any size stack anywhere in 64-Kbyte 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-Kbyte 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

8.2 Programmer’s model and CPU registers

Figure 15 shows the five CPU registers. CPU registers are not part of the memory map.

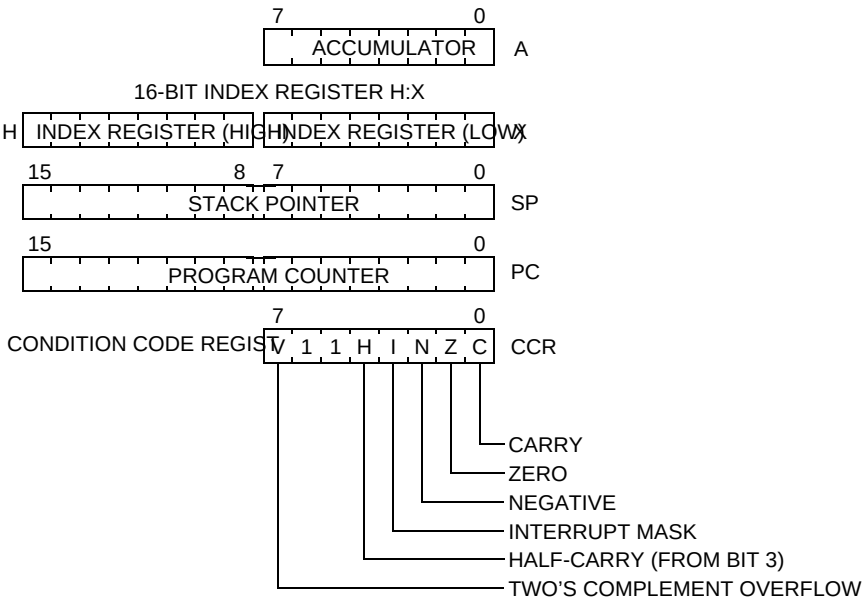


Figure 15. CPU Registers

8.2.1 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.

8.2.2 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.

8.2.3 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-Kbyte 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.

8.2.4 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.

8.2.5 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,* Freescale Semiconductor document order number HCS08RMv1.

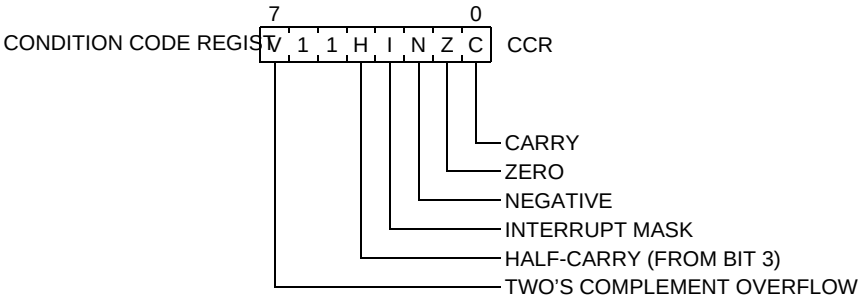


Figure 16. Condition Code register

Table 67. 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
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

Table 67. CCR register field descriptions (continued)

Field	Description
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

8.3 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-Kbyte 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.

8.3.1 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.

8.3.2 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.

8.3.3 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.

8.3.4 Direct Addressing mode (DIR)

In Direct Addressing mode, the instruction includes the low-order eight 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. This is faster and more memory efficient than specifying a complete 16-bit address for the operand.

8.3.5 Extended Addressing mode (EXT)

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

8.3.6 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.

8.3.6.1 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.

8.3.6.2 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.

8.3.6.3 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.

8.3.6.4 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.

8.3.6.5 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.

8.3.6.6 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.

8.3.6.7 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.

8.4 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.

8.4.1 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, refer to the Section 5.

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.

8.4.2 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 three 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.

8.4.3 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 will resume and the interrupt or reset event will be 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 will resume and the CPU will enter 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.

8.4.4 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 will resume and the CPU will enter 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. Refer to the Section 3 for more details.

8.4.5 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.

8.5 HCS08 instruction set summary

Instruction set summary nomenclature

The nomenclature listed here is used in the instruction descriptions in Table 68.

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)

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

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.

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)

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

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

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-Kbyte 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-Kbyte 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 will calculate the 8-bit signed offset and include it in the object code for this instruction.

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 68. 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* ADC *opr8a* ADC *opr16a* ADC *oprx16*,X ADC *oprx8*,X ADC ,X ADC *oprx16*,SP ADC *oprx8*,SP	Add with Carry	A ← (A) + (M) + (C)	↕	↕	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP2 SP1	A9 B9 C9 D9 E9 F9 9ED9 9EE9	ii dd hh ll ee ff ff	2 3 4 4 3 3 5 4
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)	↕	↕	–	↕	↕	↕	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	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP2 SP1	A4 B4 C4 D4 E4 F4 9ED4 9EE4	ii dd hh ll ee ff ff	2 3 4 4 3 3 5 4

Table 68. HCS08 instruction set summary (continued)

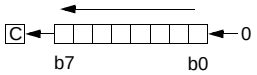
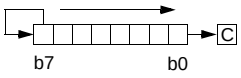
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles[1]
			V	H	I	N	Z	C				
ASL *opr8a* ASLA ASLX ASL *oprx8*,X ASL ,X ASL *oprx8*,SP	Arithmetic Shift Left (Same as LSL)		þ	–	–	þ	þ	þ	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	5 1 1 5 4 6
ASR *opr8a* ASRA ASRX ASR *oprx8*,X ASR ,X ASR *oprx8*,SP	Arithmetic Shift Right		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
BGE *rel*	Branch if Greater Than or Equal To (Signed Operands)	Branch if $(N \oplus V) = 0$	–	–	–	–	–	–	REL	90	rr	3
BGND	Enter Active Background 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) | $(N \oplus V) = 0$	–	–	–	–	–	–	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 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) | $(N \oplus V) = 1$	–	–	–	–	–	–	REL	93	rr	3
BLO *rel*	Branch if Lower (Same as BCS)	Branch if (C) = 1	–	–	–	–	–	–	REL	25	rr	3

Table 68. 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				
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
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

Table 68. 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				
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
CMP #*oprx8* 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)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP2 SP1	A1 B1 C1 D1 E1 F1 9ED1 9EE1	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 4
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	–	–	↕	↕	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)	↕	–	–	↕	↕	↕	EXT IMM DIR SP1	3E 65 75 9EF3	hh ll jj kk dd ff	6 3 5 6
CPX #*oprx8* 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)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP2 SP1	A3 B3 C3 D3 E3 F3 9ED3 9EE3	ii dd hh ll ee ff ff 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	–	–	↕	↕	↕	INH	72		1
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 *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	↕	–	–	↕	↕	–	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	–	–	–	–	↕	↕	INH	52		6

Table 68. HCS08 instruction set summary (continued)

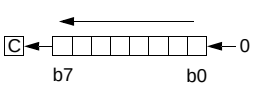
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles[1]
			V	H	I	N	Z	C				
EOR *#opr8i* EOR *opr8a* EOR *opr16a* EOR *oprx16*,X EOR *oprx8*,X EOR ,X EOR *oprx16*,SP EOR *oprx8*,SP	Exclusive OR Memory with Accumulator	A ← (A ⊕ M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX 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	↕	–	–	↕	↕	–	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	–	–	↕	↕	–	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	–	–	↕	↕	–	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	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP2 SP1	AE BE CE DE EE FE 9EDE 9EEE	ii dd hh ll ee ff ff	2 3 4 4 3 3 5 4
LSL *opr8a* LSLA LSLX LSL *oprx8*,X LSL ,X LSL *oprx8*,SP	Logical Shift Left (Same as ASL)		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	5 1 1 5 4 6

Table 68. HCS08 instruction set summary (continued)

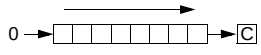
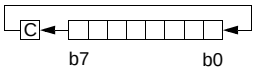
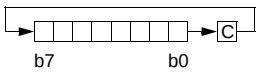
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Bus Cycles[1]
			V	H	I	N	Z	C				
LSR *opr8a* LSRA LSR*X* LSR *oprx8*,X LSR ,X LSR *oprx8*,SP	Logical Shift Right		þ	–	–	0	þ	þ	DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff	5 1 1 5 4 6
MOV *oprx8*,*oprx8* MOV *oprx8*,X+ MOV *#oprx8*,*oprx8* MOV ,X+,*oprx8*	Move	$(M)_{destination} \leftarrow (M)_{source}$ H:X ← (H:X) + 0x0001 in IX+/DIR and DIR/IX+ Modes	0	–	–	þ	þ	–	DIR/DIR DIR/IX+ IMM/DIR IX+/DIR	4E 5E 6E 7E	dd dd dd ii dd dd	5 5 4 5
MUL	Unsigned multiply	X:A ← (X) × (A)	–	0	–	–	–	0	INH	42		5
NEG *oprx8* NEGA NEGX NEG *oprx8*,X NEG ,X NEG *oprx8*,SP	Negate (Two's Complement)	M ← – (M) = 0x00 – (M) A ← – (A) = 0x00 – (A) X ← – (X) = 0x00 – (X) M ← – (M) = 0x00 – (M) M ← – (M) = 0x00 – (M) M ← – (M) = 0x00 – (M)	þ	–	–	þ	þ	þ	DIR INH INH IX1 IX SP1	30 40 50 60 70 9E60	dd ff ff	5 1 1 5 4 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 *#opr8i* ORA *opr8a* ORA *opr16a* ORA *oprx16*,X ORA *oprx8*,X ORA ,X ORA *oprx16*,SP ORA *oprx8*,SP	Inclusive OR Accumulator and Memory	A ← (A) | (M)	0	–	–	þ	þ	–	IMM DIR EXT IX2 IX1 IX SP2 SP1	AA BA CA DA EA FA 9EDA 9EEA	ii dd hh ll ee ff ff ee ff ff	2 3 4 4 3 3 5 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
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		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		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

Table 68. 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				
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	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
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 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 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 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

Table 68. 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				
TAP	Transfer Accumulator to CCR	CCR ← (A)	þ	þ	þ	þ	þ	þ	INH	84		1
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 69. Opcode map

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 69. Opcode map (continued)

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

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+		

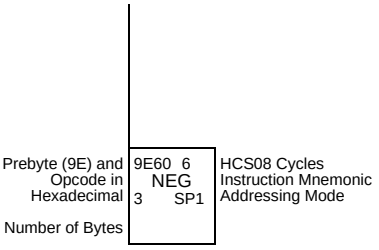
Opcode in Hexadecimal — F0 3 SUB — HCS08 Cycles
Number of Bytes — 1 IX — Instruction Mnemonic / Addressing Mode

Table 69. Opcode map (continued)

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									
										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

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)



9 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 PTA2:3 pins, if those pins are not needed for an LFR channel or other general purpose I/O function. The following clock source and frequency selections are available using the system option register 2 as shown in Figure 37 and Table 38.

In addition one channel of the TPM1 can be connected to a 500 kHz clock (D_X) derived from the crystal oscillator on the RFM. This selection is made by setting the TPM1 to use an external clock. This clock source allows time calibration of the LFO as described in the Section 14.

Features

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

9.1 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 table below.

Table 70. 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

9.1.1 Block diagram

Figure 17 shows the structure of a TPM1.

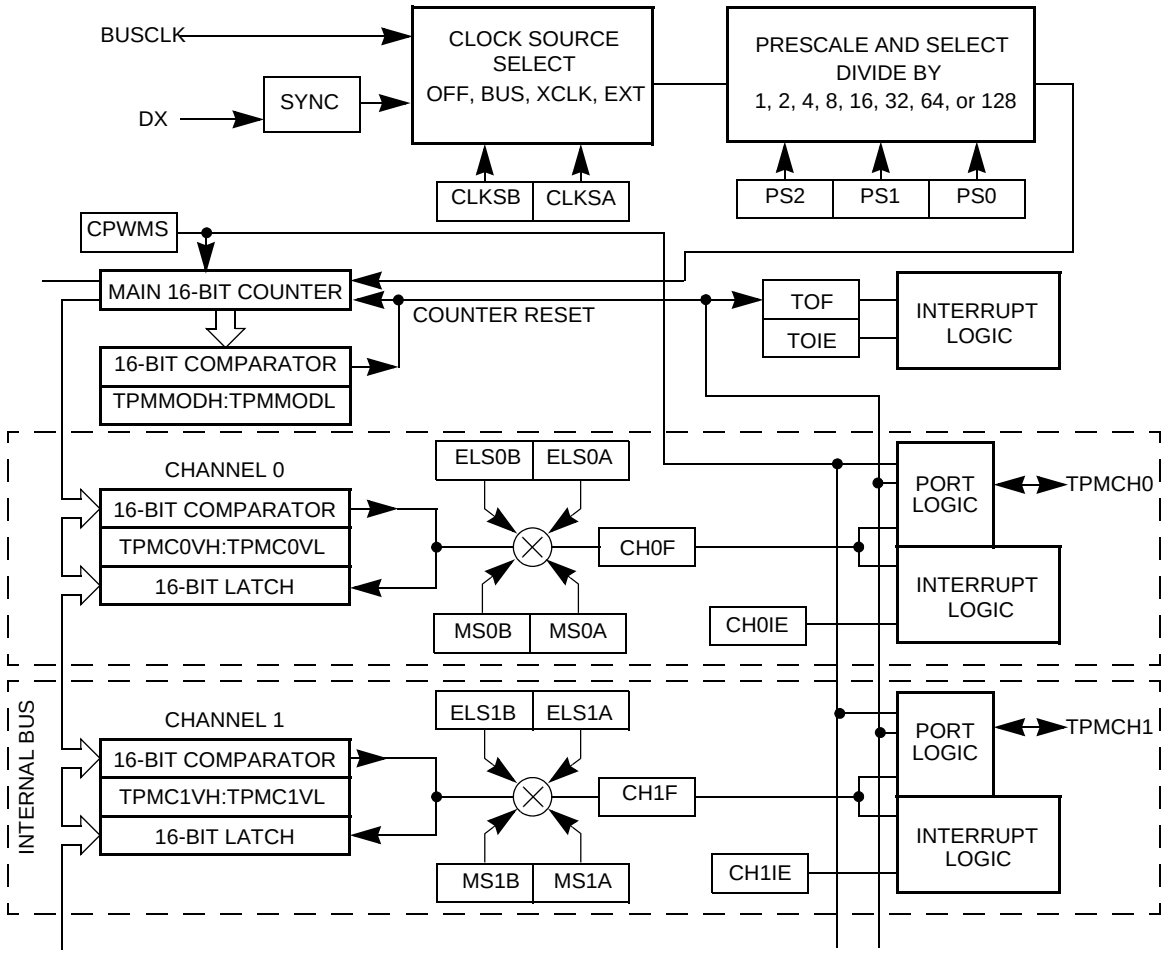


Figure 17. TPM1 block diagram

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.

9.2 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 2 for additional information about shared pin functions.

9.3 Register definition

The TPM1 includes:

- An 8-bit status and control register (TPMSC)
- A 16-bit counter (TPMCNTH:TPMCNTL)
- A 16-bit modulo register (TPMMODH:TPMMODL)

Each timer channel has:

- An 8-bit status and control register (TPMCnSC)
- A 16-bit channel value register (TPMCnVH:TPMCnVL)

9.3.1 Timer Status and Control register (TPM1SC)

TPM1SC contains the overflow status flag and control bits that are used to configure the interrupt enable, TPM1 configuration, clock source, and prescale divisor. These controls relate to all channels within this timer module.

Timer Status and Control register (TPM1SC)

$0010	7	6	5	4	3	2	1	0
R	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 71. TPM1SC register field descriptions

Field	Description
7 TOF	**Timer Overflow Flag** — This flag is set when the TPM1 counter changes to 0x0000 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. Reset clears TOF. Writing a 1 to TOF has no effect. 0 TPM1 counter has not reached modulo value or overflow 1 TPM1 counter has overflowed
6 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. Reset clears TOIE. 0 TOF interrupts inhibited (use software polling) 1 TOF interrupts enabled
5 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. Reset clears CPWMS. 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 1 All TPM channels operate in center-aligned PWM mode
4:3 CLKS[B:A]	**Clock Source Select** — As shown in Table 70, this 2-bit field is 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.
2:0 PS[2:0]	**Prescale Divisor Select** — This 3-bit field selects one of eight divisors for the TPM1 clock input as shown in Table 72. 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.

Table 72. Prescale divisor selection

PS2:PS1:PS0	TPM1 clock source divided-by
0:0:0	1
0:0:1	2
0:1:0	4
0:1:1	8
1:0:0	16
1:0:1	32
1:1:0	64
1:1:1	128

9.3.2 Timer Counter registers (TPM1CNTH:TPM1CNTL)

The two read-only TPM1 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.

Table 73. Timer Counter High register(TPM1CNTH)

$0011	7	6	5	4	3	2	1	0
R	Bit 15	14	13	12	11	10	9	Bit 8
W	Any write to TPMCNTH clears the 16-bit counter.							
Reset	0	0	0	0	0	0	0	0

Table 74. Timer Counter Low register(TPM1CNTL)

$0012	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W	Any write to TPMCNTL clears the 16-bit counter.							
Reset	0	0	0	0	0	0	0	0

When background mode is active, the timer counter and the coherency mechanism are frozen such that the buffer latches remain in the state they were in when the background mode became active even if one or both bytes of the counter are read while background mode is active.

9.3.3 Timer Counter Modulo registers (TPM1MODH:TPM1MODL)

The read/write TPM1 modulo registers contain the modulo value for the TPM1 counter. After the TPM1 counter reaches the modulo value, the TPM1 counter resumes counting from 0x0000 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 sets the TPM1 counter modulo registers to 0x0000, which results in a free-running timer counter (modulo disabled).

Table 75. Timer Counter Modulo High register (TPM1MODH)

$0013	7	6	5	4	3	2	1	0
R W	Bit 15	14	13	12	11	10	9	Bit 8
Reset	0	0	0	0	0	0	0	0

Table 76. Timer Counter Modulo Low register (TPM1MODL)

$0014	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
Reset	0	0	0	0	0	0	0	0

It is good practice to wait for an overflow interrupt so both bytes of the modulo register can be written well before a new overflow. An alternative approach is to reset the TPM1 counter before writing to the TPM1 modulo registers to avoid confusion about when the first counter overflow will occur.

9.3.4 Timer Channel 0 Status and Control register (TPM1C0SC)

TPM1C0SC contains the channel interrupt status flag and control bits that are used to configure the interrupt enable, channel configuration, and pin function.

Table 77. Timer channel 0 Status and Control register (TPM1C0SC)

$0015	7	6	5	4	3	2	1	0
R	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 78. TPM1C0SC register field descriptions

Field	Description
7 CH0F	**Channel 0 Flag** — When channel n is configured for input capture, this flag bit is set when an active edge occurs on the channel n pin. When channel 0 is an output compare or edge-aligned PWM channel, CH0F is set when the value in the TPM1 counter registers matches the value in the TPM1 channel 0 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 correspond to both edges of the active duty cycle period. A corresponding interrupt is requested when CH0F is set and interrupts are enabled (CH0IE = 1). Clear CH0F by reading TPM1C0SC while CH0F is set and then writing a 0 to CH0F. If another interrupt request occurs before the clearing sequence is complete, the sequence is reset so CH0F would remain set after the clear sequence was completed for the earlier CH0F. This is done so a CH0F interrupt request cannot be lost by clearing a previous CH0F. Reset clears CH0F. Writing a 1 to CH0F has no effect. 0 No input capture or output compare event occurred on channel 0 1 Input capture or output compare event occurred on channel 0
6 CH0IE	**Channel 0 Interrupt Enable** — This read/write bit enables interrupts from channel 0. Reset clears CH0IE. 0 Channel 0 interrupt requests disabled (use software polling) 1 Channel 0 interrupt requests enabled
5 MS0B	**Mode Select B for TPM1 Channel 0** — When CPWMS = 0, MS0B = 1 configures TPM1 channel 0 for edge-aligned PWM mode. For a summary of channel mode and setup controls, refer to Table 79.
4 MS0A	**Mode Select A for TPM1 Channel 0** — When CPWMS = 0 and MS0B = 0, MS0A configures TPM1 channel 0 for input capture mode or output compare mode. Refer to Table 79 for a summary of channel mode and setup controls.
3:2 ELS0[B:A]	**Edge/Level Select Bits** — Depending on the operating mode for the timer channel as set by CPWMS:MS0B:MSnA and shown in Table 79, 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 ELS0B:ELS0A 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 79. Mode, edge, and level selection

CPWMS	MS0B:MS0A	ELS0B:ELS0A	Mode	Configuration
X	XX	00	Pin not used for TPM1 channel; use as an external clock for the TPM1 or revert to general-purpose I/O	
0	00	01	Input capture	Capture on rising edge only
		10		Capture on falling edge only
		11		Capture on rising or falling edge
	01	00	Output compare	Software compare only
		01		Toggle output on compare
		10		Clear output on compare
		11		Set output on compare
	1X	10	Edge-aligned PWM	High-true pulses (clear output on compare)
		X1		Low-true pulses (set output on compare)

Table 79. Mode, edge, and level selection (continued)

CPWMS	MS0B:MS0A	ELS0B:ELS0A	Mode	Configuration
1	XX	10	Center-aligned PWM	High-true pulses (clear output on compare-up)
		X1		Low-true pulses (set output on compare-up)

If the associated port pin is not stable for at least two bus clock cycles before changing to input capture mode, it is possible to get an unexpected indication of an edge trigger. Typically, a program would clear status flags after changing channel configuration bits and before enabling channel interrupts or using the status flags to avoid any unexpected behavior.

9.3.5 Timer Channel Value registers (TPM1C0VH:TPM1C0VL)

These 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.

Table 80. Timer Channel 0 Value High register (TPM1C0VH)

$0016	7	6	5	4	3	2	1	0
R W	Bit 15	14	13	12	11	10	9	Bit 8
Reset	0	0	0	0	0	0	0	0

Table 81. Timer channel 0 value Low register (TPM1C0VL)

$0017	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
Reset	0	0	0	0	0	0	0	0

In input capture mode, reading either byte (TPM1C0VH or TPM1C0VL) 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 TPM1C0SC register is written.

In output compare or PWM modes, writing to either byte (TPM1C0VH or TPM1C0VL) 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 TPM1C0SC register.

This latching mechanism allows coherent 16-bit writes in either order, which is friendly to various compiler implementations.

9.3.6 Timer Channel 1 Status and Control register (TPM1C1SC)

TPM1C1SC contains the channel interrupt status flag and control bits that are used to configure the interrupt enable, channel configuration, and pin function.

Table 82. Timer Channel 1 Status and Control register (TPM1C1SC)

$0018	7	6	5	4	3	2	1	0
R	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 83. TPM1C1SC register field descriptions

Field	Description
7 CH1F	**Channel 1 Flag** — When channel n is configured for input capture, this flag bit is set when an active edge occurs on the channel n pin. When channel 1 is an output compare or edge-aligned PWM channel, CH1F is set when the value in the TPM1 counter registers matches the value in the TPM1 channel 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 correspond to both edges of the active duty cycle period. A corresponding interrupt is requested when CH1F is set and interrupts are enabled (CH1IE = 1). Clear CH1F by reading TPM1C1SC while CH1F is set and then writing a 0 to CH1F. If another interrupt request occurs before the clearing sequence is complete, the sequence is reset so CH1F would remain set after the clear sequence was completed for the earlier CH1F. This is done so a CH1F interrupt request cannot be lost by clearing a previous CH1F. Reset clears CH1F. Writing a 1 to CH1F has no effect. 0 No input capture or output compare event occurred on channel 1 1 Input capture or output compare event occurred on channel 1
6 CH1IE	**Channel 1 Interrupt Enable** — This read/write bit enables interrupts from channel 1. Reset clears CH1IE. 0 Channel 1 interrupt requests disabled (use software polling) 1 Channel 1 interrupt requests enabled
5 MS1B	**Mode Select B for TPM1 Channel 1** — When CPWMS = 0, MS1B = 1 configures TPM1 channel 1 for edge-aligned PWM mode. For a summary of channel mode and setup controls, refer to Table 79.
4 MS1A	**Mode Select A for TPM1 Channel 1** — When CPWMS = 0 and MS1B = 0, MS1A configures TPM1 channel 1 for input capture mode or output compare mode. Refer to Table 79 for a summary of channel mode and setup controls.
3:2 ELS1[B:A]	**Edge/Level Select Bits** — Depending on the operating mode for the timer channel as set by CPWMS:MS1B:MS1A and shown in Table 79, 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 ELS1B:ELS1A 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 84. Mode, edge, and level selection

CPWMS	MS1B:MS1A	ELS1B:ELS1A	Mode	Configuration
X	XX	00	Pin not used for TPM1 channel; use as an external clock for the TPM1 or revert to general-purpose I/O	
0	00	01	Input capture	Capture on rising edge only
		10		Capture on falling edge only
		11		Capture on rising or falling edge
	01	00	Output compare	Software compare only
		01		Toggle output on compare
		10		Clear output on compare
		11		Set output on compare
	1X	10	Edge-aligned PWM	High-true pulses (clear output on compare)
		X1		Low-true pulses (set output on compare)
1	XX	10	Center-aligned PWM	High-true pulses (clear output on compare-up)
		X1		Low-true pulses (set output on compare-up)

If the associated port pin is not stable for at least two bus clock cycles before changing to input capture mode, it is possible to get an unexpected indication of an edge trigger. Typically, a program would clear status flags after changing channel configuration bits and before enabling channel interrupts or using the status flags to avoid any unexpected behavior.

9.3.7 Timer Channel Value registers (TPM1C1VH:TPM1C1VL)

These 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.

Table 85. Timer Channel 1 Value High register (TPM1C1VH)

$0019	7	6	5	4	3	2	1	0
R W	Bit 15	14	13	12	11	10	9	Bit 8
Reset	0	0	0	0	0	0	0	0

Table 86. Timer Channel 1 Value Low register (TPM1C1VL)

$001A	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
Reset	0	0	0	0	0	0	0	0

In input capture mode, reading either byte (TPM1C1VH or TPM1C1VL) 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 TPM1C1SC register is written.

In output compare or PWM modes, writing to either byte (TPM1C1VH or TPM1C1VL) 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 TPM1C1SC register.

This latching mechanism allows coherent 16-bit writes in either order, which is friendly to various compiler implementations.

9.4 Functional description

All TPM1 functions are associated with a main 16-bit counter that allows flexible selection of the clock source and prescale divisor. A 16-bit modulo register also is associated with the main 16-bit counter in the TPM1. Each TPM1 channel is optionally associated with an MCU pin and a maskable interrupt function.

The TPM1 has center-aligned PWM capabilities controlled by the CPWMS control bit in TPM1SC. When CPWMS is set to 1, timer counter TPM1CNT changes to an up-/down-counter and all channels in the associated TPM1 act as center-aligned PWM channels. When CPWMS = 0, each channel can independently be configured to operate in input capture, output compare, or buffered edge-aligned PWM mode.

The following sections describe the main 16-bit counter and each of the timer operating modes (input capture, output compare, edge-aligned PWM, and center-aligned PWM). Because details of pin operation and interrupt activity depend on the operating mode, these topics are covered in the associated mode sections.

9.4.1 Counter

All timer functions are based on the main 16-bit counter (TPM1CNTH:TPM1CNTL). This section discusses selection of the clock source, up-counting vs. up-/down-counting, end-of-count overflow, and manual counter reset.

After any MCU reset, CLKSB:CLKSA = 0:0 so no clock source is selected and the TPM1 is inactive. Normally, CLKSB:CLKSA would be set to 0:1 so the bus clock drives the timer counter. The clock source for each of the TPM1 can be independently selected to be off, the bus clock (BUSCLK), the fixed system clock (XCLK), or an external input. The maximum frequency allowed for the external clock option is one-fourth the bus rate. Refer to Section 9.3.1 and Table 78 for more information about clock source selection.

When the microcontroller is in active background mode, the TPM1 temporarily suspends all counting until the microcontroller returns to normal user operating mode. During stop mode, all TPM1 clocks are stopped; therefore, the TPM1 is effectively disabled until clocks resume. During Wait mode, the TPM1 continues to operate normally.

The main 16-bit counter has two counting modes. When center-aligned PWM is selected (CPWMS = 1), the counter operates in up-/down-counting mode. Otherwise, the counter operates as a simple up-counter. As an up-counter, the main 16-bit counter counts from 0x0000 through its terminal count and then continues with 0x0000. The terminal count is 0xFFFF or a modulus value in TPM1MODH:TPM1MODL.

When center-aligned PWM operation is specified, the counter counts upward from 0x0000 through its terminal count and then counts downward to 0x0000 where it returns to up-counting. Both 0x0000 and the terminal count value (value in TPM1MODH:TPM1MODL) are normal length counts (one timer clock period long).

An interrupt flag and enable are associated with the main 16-bit counter. The timer overflow flag (TOF) is a software-accessible indication that the timer counter has overflowed. The enable signal selects between software polling (TOIE = 0) where no hardware interrupt is generated, or interrupt-driven operation (TOIE = 1) where a static hardware interrupt is automatically generated whenever the TOF flag is 1.

The conditions that cause TOF to become set depend on the counting mode (up or up/down). In up-counting mode, the main 16-bit counter counts from 0x0000 through 0xFFFF and overflows to 0x0000 on the next counting clock. TOF becomes set at the transition from 0xFFFF to 0x0000. When a modulus limit is set, TOF becomes set at the transition from the value set in the modulus register to 0x0000. When the main 16-bit counter is operating in up-/down-counting mode, the TOF flag gets set as the counter changes direction at the transition from the value set in the modulus register and the next lower count value. This corresponds to the end of a PWM period. (The 0x0000 count value corresponds to the center of a period.)

Because the HCS08 MCU is an 8-bit architecture, a coherency mechanism is built into the timer counter for read operations. Whenever either byte of the counter is read (TPM1CNTH or TPM1CNTL), both bytes are captured into a buffer so when the other byte is read, the value will represent the other byte of the count at the time the first byte was read. The counter continues to count normally, but no new value can be read from either byte until both bytes of the old count have been read.

The main timer counter can be reset manually at any time by writing any value to either byte of the timer count TPM1CNTH or TPM1CNTL. Resetting the counter in this manner also resets the coherency mechanism in case only one byte of the counter was read before resetting the count.

9.4.2 Channel mode selection

Provided CPWMS = 0 (center-aligned PWM operation is not specified), the MSnB and MSnA control bits in the channel n status and control registers determine the basic mode of operation for the corresponding channel. Choices include input capture, output compare, and buffered edge-aligned PWM.

9.4.2.1 Input capture mode

With the input capture function, the TPM1 can capture the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the TPM1 latches the contents of the TPM1 counter into the channel value registers (TPM1CnVH:TPM1CnVL). Rising edges, falling edges, or any edge may be chosen as the active edge that triggers an input capture.

When either byte of the 16-bit capture register is read, both bytes are latched into a buffer to support coherent 16-bit accesses regardless of order. The coherency sequence can be manually reset by writing to the channel status/control register (TPM1CnSC).

An input capture event sets a flag bit (CHnF) that can optionally generate a CPU interrupt request.

9.4.2.2 Output compare mode

With the output compare function, the TPM1 can generate timed pulses with programmable position, polarity, duration, and frequency. When the counter reaches the value in the channel value registers of an output compare channel, the TPM1 can set, clear, or toggle the channel pin.

In output compare mode, values are transferred to the corresponding timer channel value registers only after both 8-bit bytes of a 16-bit register have been written. This coherency sequence can be manually reset by writing to the channel status/control register (TPM1CnSC).

An output compare event sets a flag bit (CHnF) that can optionally generate a CPU interrupt request.

9.4.2.3 Edge-aligned PWM mode

This type of PWM output uses the normal up-counting mode of the timer counter (CPWMS = 0) and can be used when other channels in the same TPM1 are configured for input capture or output compare functions. The period of this PWM signal is determined by the setting in the modulus register (TPM1MODH:TPM1MODL). The duty cycle is determined by the setting in the timer channel value register (TPM1CnVH:TPM1CnVL). The polarity of this PWM signal is determined by the setting in the ELSnA control bit. Duty cycle cases of 0 percent and 100 percent are possible.

As Figure 18 shows, the output compare value in the TPM1 channel registers determines the pulse width (duty cycle) of the PWM signal. The time between the modulus overflow and the output compare is the pulse width. If ELSnA = 0, the counter overflow forces the PWM signal high and the output compare forces the PWM signal low. If ELSnA = 1, the counter overflow forces the PWM signal low and the output compare forces the PWM signal high.

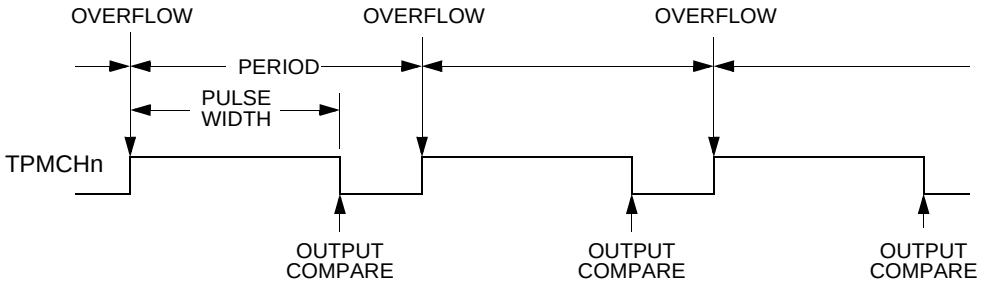


Figure 18. PWM period and pulse width (ELSnA = 0)

When the channel value register is set to 0x0000, the duty cycle is 0 percent. By setting the timer channel value register (TPMCnVH:TPMCnVL) to a value greater than the modulus setting, 100% duty cycle can be achieved. This implies that the modulus setting must be less than 0xFFFF to get 100% duty cycle.

Because the HCS08 is a family of 8-bit MCUs, the settings in the timer channel registers are buffered to ensure coherent 16-bit updates and to avoid unexpected PWM pulse widths. Writes to either register, TPM1CnVH or TPM1CnVL, write to buffer registers. In edge-PWM mode, values are transferred to the corresponding timer channel registers only after both 8-bit bytes of a 16-bit register have been written and the value in the 1TPMCNTH:TPM1CNTL counter is 0x0000. (The new duty cycle does not take effect until the next full period.)

9.4.3 Center-aligned PWM mode

This type of PWM output uses the up-/down-counting mode of the timer counter (CPWMS = 1). The output compare value in TPM1CnVH:TPM1CnVL determines the pulse width (duty cycle) of the PWM signal and the period is determined by the value in TPM1MODH:TPM1MODL. TPM1MODH:TPM1MODL should be kept in the range of 0x0001 to 0x7FFF because values outside this range can produce ambiguous results. ELS0A will determine the polarity of the CPWM output.

pulse width = 2 x (TPM1CnVH:TPM1CnVL)

period = 2 x (TPM1MODH:TPM1MODL);
for TPM1MODH:TPM1MODL = 0x0001–0x7FFF

If the channel value register TPM1CnVH:TPM1CnVL is zero or negative (bit 15 set), the duty cycle will be 0%. If TPM1CnVH:TPM1CnVL is a positive value (bit 15 clear) and is greater than the (nonzero) modulus setting, the duty cycle will be 100% because the duty cycle compare will never occur. This implies the usable range of periods set by the modulus register is 0x0001 through 0x7FFE (0x7FFF if generation of 100% duty cycle is not necessary). This is not a significant limitation because the resulting period is much longer than required for normal applications.

TPM1MODH:TPM1MODL = 0x0000 is a special case that should not be used with center-aligned PWM mode. When CPWMS = 0, this case corresponds to the counter running free from 0x0000 through 0xFFFF, but when CPWMS = 1 the counter needs a valid match to the modulus register somewhere other than at 0x0000 in order to change directions from up-counting to down-counting.

Figure 19 shows the output compare value in the TPM1 channel registers (multiplied by 2), which determines the pulse width (duty cycle) of the CPWM signal. If ELSnA = 0, the compare match while counting up forces the CPWM output signal low and a compare match while counting down forces the output high. The counter counts up until it reaches the modulo setting in TPM1MODH:TPM1MODL, then counts down until it reaches zero. This sets the period equal to two times TPM1MODH:TPM1MODL.

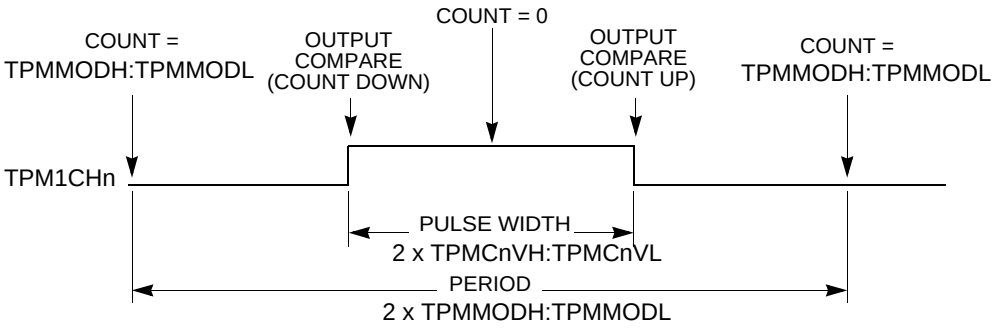


Figure 19. CPWM period and pulse width (ELSnA = 0)

Center-aligned PWM outputs typically produce less noise than edge-aligned PWMs because fewer I/O pin transitions are lined up at the same system clock edge. This type of PWM is also required for some types of motor drives.

Because the HCS08 is a family of 8-bit MCUs, the settings in the timer channel registers are buffered to ensure coherent 16-bit updates and to avoid unexpected PWM pulse widths. Writes to any of the registers, TPM1MODH, TPM1MODL, TPM1CnVH, and TPM1CnVL, actually write to buffer registers. Values are transferred to the corresponding timer channel registers only after both 8-bit bytes of a 16-bit register have been written and the timer counter overflows (reverses direction from up-counting to down-counting at the end of the terminal count in the modulus register). This TPM1CNT overflow requirement only applies to PWM channels, not output compares.

Optionally, when TPM1CNTH:TPM1CNTL = TPM1MODH:TPM1MODL, the TPM1 can generate a TOF interrupt at the end of this count. The user can choose to reload any number of the PWM buffers, and they will all update simultaneously at the start of a new period.

Writing to TPM1SC cancels any values written to TPM1MODH and/or TPM1MODL and resets the coherency mechanism for the modulo registers. Writing to TPM1C0SC cancels any values written to the channel value registers and resets the coherency mechanism for TPM1C0VH:TPM1C0VL.

9.5 TPM1 interrupts

The TPM1 generates an optional interrupt for the main counter overflow and an interrupt for each channel. The meaning of channel interrupts depends on the mode of operation for each channel. If the channel is configured for input capture, the interrupt flag is set each time the selected input capture edge is recognized. If the channel is configured for output compare or PWM modes, the interrupt flag is set each time the main timer counter matches the value in the 16-bit channel value register. See Section 5 for absolute interrupt vector addresses, priority, and local interrupt mask control bits.

For each interrupt source in the TPM1, a flag bit is set on recognition of the interrupt condition such as timer overflow, channel input capture, or output compare events. This flag may be read (polled) by software to verify that the action has occurred, or an associated enable bit (TOIE or CHnIE) can be set to enable hardware interrupt generation. While the interrupt enable bit is set, a static interrupt will be generated whenever the associated interrupt flag equals 1. It is the responsibility of user software to perform a sequence of steps to clear the interrupt flag before returning from the interrupt service routine.

9.5.1 Clearing timer interrupt flags

TPM1 interrupt flags are cleared by a two-step process that includes a read of the flag bit while it is set (1) followed by a write of 0 to the bit. If a new event is detected between these two steps, the sequence is reset and the interrupt flag remains set after the second step to avoid the possibility of missing the new event.

9.5.2 Timer overflow interrupt description

The conditions that cause TOF to become set depend on the counting mode (up or up/down). In up-counting mode, the 16-bit timer counter counts from 0x0000 through 0xFFFF and overflows to 0x0000 on the next counting clock. TOF becomes set at the transition from 0xFFFF to 0x0000. When a modulus limit is set, TOF becomes set at the transition from the value set in the modulus register to 0x0000. When the counter is operating in up-/down-counting mode, the TOF flag gets set as the counter changes direction at the transition from the value set in the modulus register and the next lower count value. This corresponds to the end of a PWM period. (The 0x0000 count value corresponds to the center of a period.)

9.5.3 Channel event interrupt description

The meaning of channel interrupts depends on the current mode of the channel (input capture, output compare, edge-aligned PWM, or center-aligned PWM).

When a channel is configured as an input capture channel, the ELSnB:ELSnA control bits select rising edges, falling edges, any edge, or no edge (off) as the edge that triggers an input capture event. When the selected edge is detected, the interrupt flag is set. The flag is cleared by the 2-step sequence described in Section 9.5.1.

When a channel is configured as an output compare channel, the interrupt flag is set each time the main timer counter matches the 16-bit value in the channel value register. The flag is cleared by the 2-step sequence described in Section 9.5.1.

9.5.4 PWM end-of-duty-cycle events

For channels that are configured for PWM operation, there are two possibilities:

- When the channel is configured for edge-aligned PWM, the channel flag is set when the timer counter matches the channel value register that marks the end of the active duty cycle period.
- When the channel is configured for center-aligned PWM, the timer count matches the channel value register twice during each PWM cycle. In this CPWM case, the channel flag is set at the start and at the end of the active duty cycle, which are the times when the timer counter matches the channel value register.

The flag is cleared by the 2-step sequence described in Section 9.5.1.

10 Other MCU Resources

NOTE

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 MPXx85/86xxD measures seven physical parameters for use in the tire pressure monitoring application: pressure, temperature, battery voltage, two external voltages and both X-axis and Z-axis acceleration. Each parameter is accessed in a different manner and all use firmware subroutine calls as described in Section 14. These subroutines initialize some control bits within the sensor measurement interface, SMI, and then place the MCU into the Stop1 mode until the measurement is completed with an interrupt back to the MCU.

NOTE

The accuracy, power consumption and timing specified for any measurement given in the electrical specifications in Section 17 are only guaranteed if the user obtains a reading using the specified firmware subroutine call in Section 14.

The MPXx85/86xxD 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 electrical specifications in Section 17 are based on using the assigned firmware subroutines in Section 14 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.

Table 87. ADC10 channel assignments

ADC10 channel	Input select	Firmware call(s)	Characteristics
AD0	Pressure Sensor X-axis Acceleration Sensor Z-axis Acceleration Sensor	TPMS_READ_PRESSURE TPMS_READ_ACCELERATION_X TPMS_READ_ACCELERATION_Z	P_{CODE} A_{XCODE} A_{ZCODE}
AD1	Temperature Sensor	TPMS_READ_TEMPERATURE	T_{XCODE}
AD2	Bandgap Reference	TPMS_READ_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	

10.1 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 100-450 kPa range:

$$P = \Delta P_{450} \times P_{CODE} + (100 - \Delta P_{450}) \qquad \textit{Eqn. 1}$$

The transfer equation of the 100-900 kPa range is:

$$P = \Delta P_{900} \times P_{CODE} + (100 - \Delta P_{900}) \qquad \textit{Eqn. 2}$$

Due to calibration routines and parameters stored in the MPXx85/86xxD the pressure range is selected at production and cannot be changed in the field.

10.2 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 MPXx85/86xxD device with the resulting transfer equation:

$$T = \Delta T \times T_{CODE} - 55 \quad \textit{Eqn. 3}$$

10.3 Voltage measurements

Voltage measurements can be made on the internal bandgap to estimate the supply voltage on V_{DD}.

10.3.1 Internal bandgap

An internal bandgap voltage reference is provided to take measurements of the supply voltage. The resulting transfer equation:

$$V_{INT} = \Delta V_{INT} \times V_{CODE} + 1.22 \quad \textit{Eqn. 4}$$

10.3.2 External voltages

Measurements of an external voltage on either the PTA0 or PTA1 pins can be made and referenced to the internal bandgap voltage. The resulting transfer equation:

$$V_{PTAx} = \Delta V_{EXT} \times Gx_{CODE} \quad \textit{Eqn. 5}$$

where x = 0, 1 refers to PTA0 or PTA1.

10.4 Optional acceleration measurements

The MPXx85/86xx can be ordered from a selection of:

- No accelerometers
- X-axis only with range of -70 to +80 g
- Z-axis only with range of -270 to +350 g
- Z-and X-axis with Z-axis range of -210 to +240 g and X-axis range of -70 to +80 g

Sensitivities are calibrated at the center of the measurement ranges, and noted by the parameter Az-(step#) in Section 17.10 and Section 17.12. Each accelerometer axis supports 16 offset steps (numbered 0 - 15) to provide the total range, and the step number used for calibration is noted in Section 17.10 and Section 17.12.

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 preprocessed by a dynamic range firmware routine that will return the two values necessary to calculate the acceleration, A_Z or A_X in conjunction with values taken from the tables in Section 17.10 and Section 17.12.

The first value from the firmware routine is the offset step identifier, STEP, with integer values 0 to 15 (i.e. the 16 offset steps). The other value is the ADC10 data, A_{ZCODE} or A_{XCODE} with integer values 0 to 511. A_{ZCODE}/A_{XCODE}values 1 through 510 are usable; whereas values 0 and 511 indicate fault conditions. The Z-axis acceleration is scaled for ~80g range within each of the 16 offset steps. The steps are at ~40g increments, allowing for adequate overlaps.

The combined Z- and X-axis accelerations are scaled for ~60g and ~20g ranges respectively within each of the 16 offset steps. The steps are at ~30g and ~10g increments respectively, allowing for adequate overlaps.

Section 17.10 or Section 17.12 provide tables of acceleration values resulting from characterizations.

Acceleration sensitivity, $\Delta A_{Z\text{-}STEP}$ or $\Delta A_{X\text{-}STEP}$ varies between each offset step, and should be calculated by dividing the range of g's for each offset step by the usable A_{ZCODE} or A_{XCODE} range (i.e. 510):

$$\Delta A_{a\text{-}STEP} = (A_{a\text{-}STEP} @ A_{aCODE}510 - A_{a\text{-}STEP} @ A_{aCODE}1) / 510 \quad \textit{Eqn. 6}$$

Once the sensitivity $\Delta A_{Z\text{-}STEP}$ or $\Delta A_{X\text{-}STEP}$ has been calculated, the acceleration A_Z or A_X can be calculated by the re-using the $A_{a\text{-}STEP}$ @ A_{aCODE} 1 value of the offset step and the returned A_{ZCODE} or A_{XCODE} value with the following transfer function:

$$A_Z = \Delta A_{Z\text{-}STEP} \times A_{ZCODE} + (A_{Z\text{-}STEP} @ A_{ZCODE}1 - \Delta A_{Z\text{-}STEP}) \quad \textit{Eqn. 7}$$

or

$$A_X = \Delta A_{X\text{-}STEP} \times A_{XCODE} + (A_{X\text{-}STEP} @ A_{XCODE}1 - \Delta A_{X\text{-}STEP}) \quad \textit{Eqn. 8}$$

10.4.1 Optional Z-axis acceleration measurements

Z-axis -210 to +240 g range:

$$A_Z = \Delta A_{Z\text{-}7} \times A_{ZCODE} + (A_{Z\text{-}7} @ A_{ZCODE}1 - \Delta A_{Z\text{-}7}) \quad \textbf{\textit{Eqn. 9}}$$

Z-axis -270 to +350 g range:

$$A_Z = \Delta A_{Z\text{-}6} \times A_{ZCODE} + (A_{Z\text{-}6} @ A_{ZCODE}1 - \Delta A_{Z\text{-}6}) \quad \textbf{\textit{Eqn. 10}}$$

10.4.2 Optional X-axis acceleration measurements

X-axis -70 to +80 g range:

$$A_X = \Delta A_{X\text{-}7} \times A_{XCODE} + (A_{X\text{-}7} @ A_{XCODE}1 - \Delta A_{X\text{-}7}) \quad \textbf{\textit{Eqn. 11}}$$

NOTE

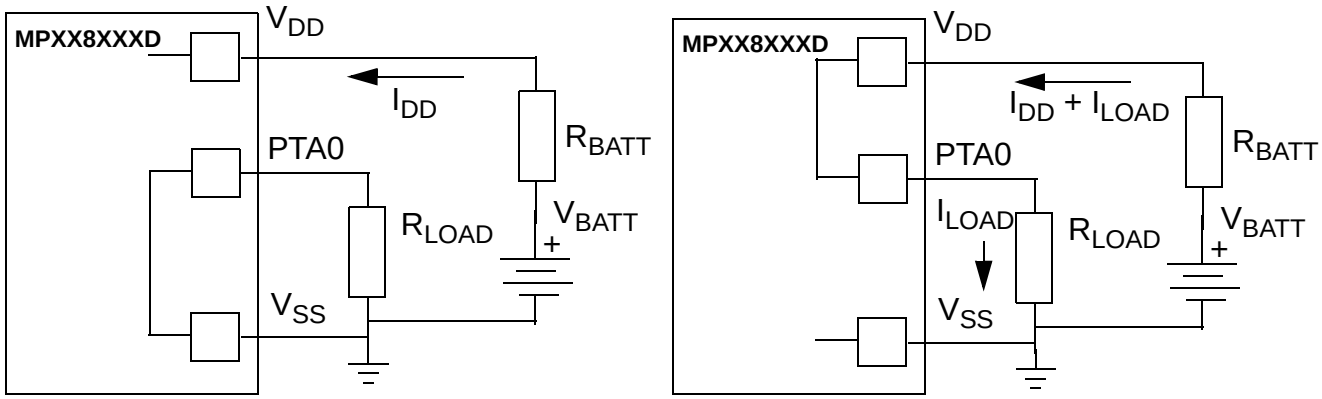
The pressure, X-axis and Z-axis accelerometer all share the same signal path in the Transducer interface and all the sensors share the same ADC. Therefore only one of the sensors can be accessed at a given moment.

NOTE

The accelerometers can be cross checked against each other to determine any permanent faults. The X-axis accelerometer should be cycling ±1 g at a rate up to 50 Hz, if the Z-axis accelerometer is indicating any continuous acceleration over 5 g. Similarly, the Z-axis accelerometer should indicate less than 5 g if the X-axis accelerometer indicates no cycling greater than 1 g.

10.5 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 20 (any of the PTA[3:0] can be used for this purpose).



Port Pin Driven Low

$$V_{DD0} = V_{BATT} - I_{DD0} \times R_{BATT}$$

Port Pin Driven High

$$V_{DD1} = V_{BATT} - (I_{DD1} + I_{LOAD}) \times R_{BATT}$$

$$V_{DD1} = V_{BATT} - \left(I_{DD1} + \frac{V_{DD1}}{R_{LOAD}} \right) \times R_{BATT}$$

Figure 20. Battery check circuit

The battery voltage can first be checked using the method given in Section 10.3 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:

$$R_{BATT} = \frac{V_{DD1} - V_{DD0}}{I_{DD0} - I_{DD1} + \frac{V_{DD1}}{R_{LOAD}}} \qquad \textit{Eqn. 12}$$

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:

$$R_{BATT} = \frac{V_{DD1} - V_{DD0}}{\frac{V_{DD1}}{R_{LOAD}}} = \frac{R_{LOAD} \times (V_{DD1} - V_{DD0})}{V_{DD1}} \qquad \textit{Eqn. 13}$$

where:

V_{DD0} is the voltage determined with the external load resistor connected to V_{SS}

V_{DD1} is the voltage determined with the external load resistor connected to V_{DD}

R_{LOAD} is the resistance of the external load resistance in ohms

R_{BATT} is the implied battery impedance in ohms

It is recommended that this calculation be performed with a reasonable current load on the battery of approximately 3 mA (R_{LOAD} approximately 1000 ohms).

10.6 Measurement firmware

The firmware for making measurements is comprised of two function calls as described in Section 14. 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 21.

The user therefore has the option to decide how often each measurement (and its component terms) are made. The resulting power consumption is then the sum of using these components are defined in the electrical specifications in Section 17.

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 Section 17.

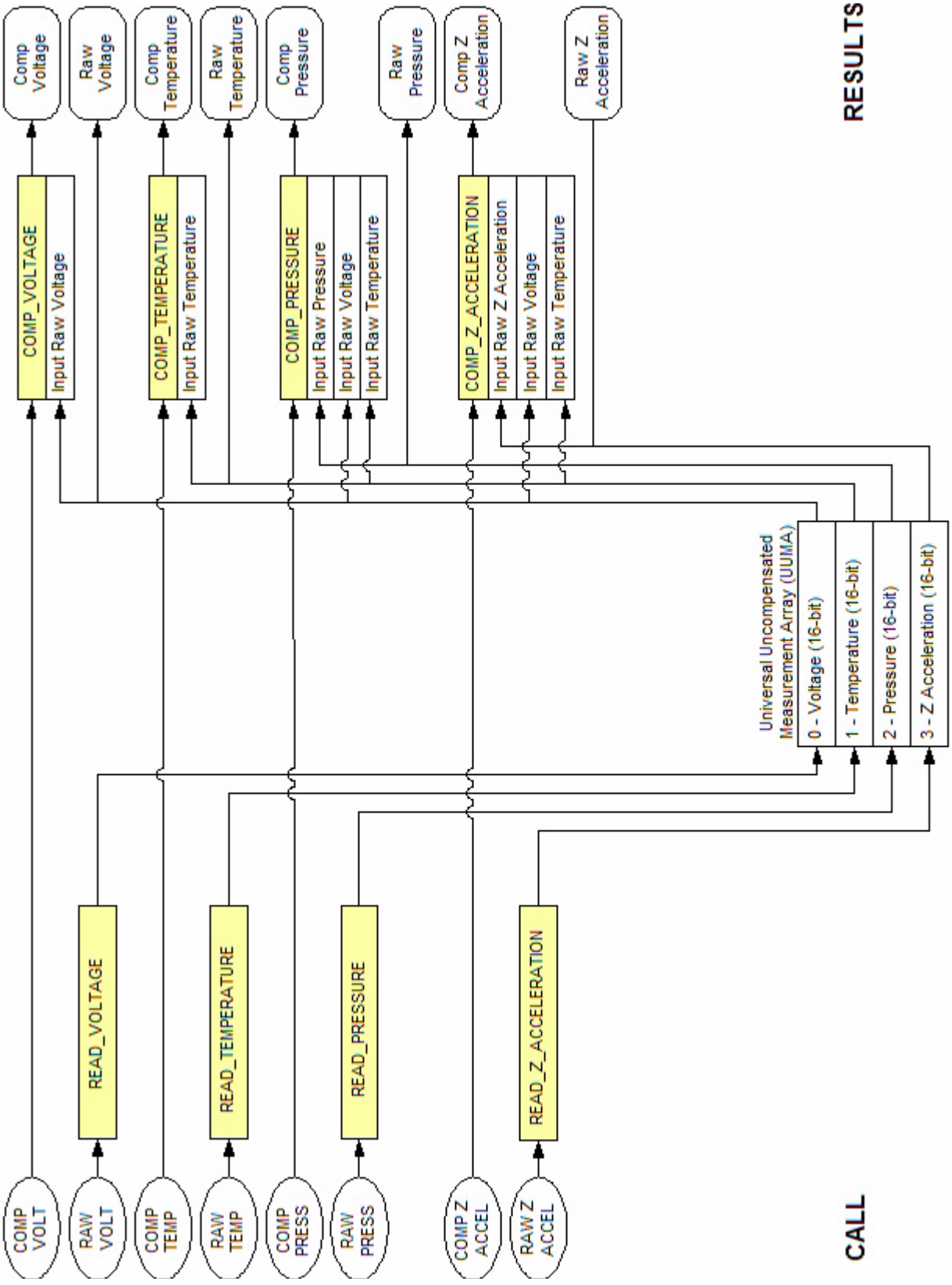


Figure 21. Data flow for measurements

10.7 Thermal shutdown

When the package temperature becomes too low or too high the MCU can be placed into a stop mode to suspend operation and prevent transmission of RF signals which may be corrupted at the temperature extremes. Return to normal operation after the temperature falls back within the recovery temperature range. The presence of either the low or high temperature shutdown will disable the PWU from causing either a periodic wake-up or a periodic reset. The MCU, temperature sensor and ADC10 are all functional over the full temperature range from T_L to T_H.

10.7.1 Low temperature shutdown

Low temperature shutdown is achieved using temperature readings taken by the ADC10 as described in Section 10.2 and enabling the thermal restart circuit by setting the TRE bit and selecting the low temperature threshold by clearing the TRH bit. When the software programmed low temperature is reached the MCU will turn off all operating functions and enter the Stop1 mode.

10.7.2 High temperature shutdown

The high-temperature shutdown level is determined from a measurement of the temperature sensor by the ADC10 as described in Section 10.2 and enabling the thermal restart circuit by setting the TRE bit and selecting the high temperature threshold by setting the TRH bit. When the software programmed high temperature is reached the MCU will turn off all operating functions and enter the Stop1 mode.

10.7.3 Temperature shutdown recovery

The MCU can be restarted by the Temperature Restart (TR) module when the temperature returns within the normal temperature range, T_{RESET}. When this occurs the MCU will be reset and begin execution from the reset vector located at $DFFE/$DFFF. The TR module can be enabled using the TRE bit in the SIMOPT1 register. The TR module can be powered on and off by setting or clearing the TRE bit located at bit 3 in the SIMOPT1 register at address $1802. The TRE bit is cleared by an MCU reset.

When the TRE bit is set the TR module can then be set to detect a recovery from either a high temperature or a low temperature using the TRH in the SIMOPT1 register. The TRH bit is cleared by an MCU reset.

The TR module does not activate an MCU restart and reset unless it has first moved outside the re-arming temperature range, T_{REARM}, as shown in Figure 22. The status of the TR can be checked by reading the TRO bit located at bit 0 in the SIMTST register at address $180F. The TRO bit is set high by an MCU reset. The state of the TRO bit is as follows:

1 = TR module is outside the T_{REARM} temperature range and will restart the MCU if the TRE bit is set and temperature falls back within the T_{RESET} temperature range.

0 = TR module is within the T_{RESET} temperature range and the MCU cannot be armed to restart when temperature falls back to the T_{RESET} range. The TRE bit cannot be set.

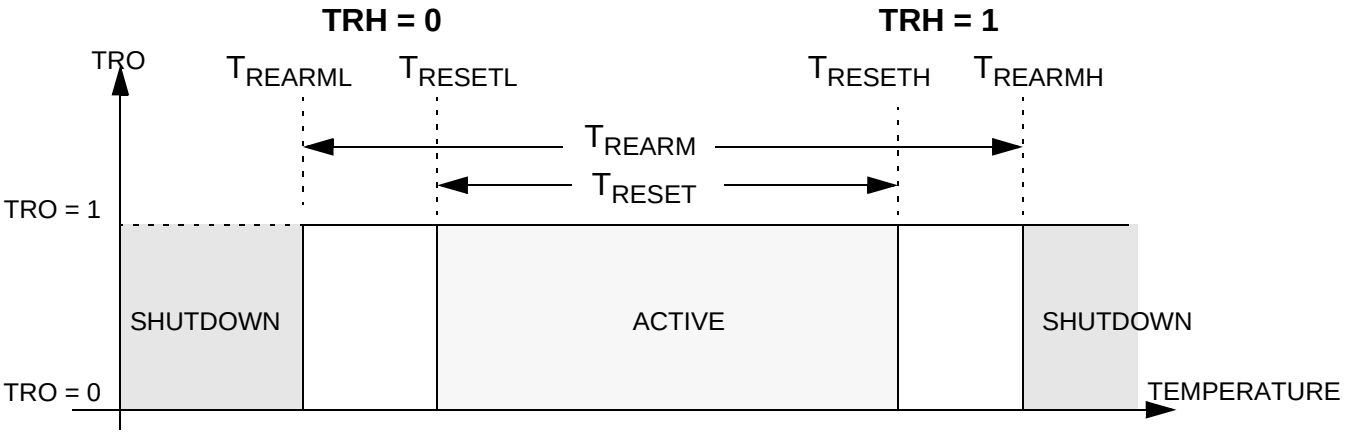


Figure 22. Temperature restart response

This sequence is further explained by the user software flowchart in Figure 23.

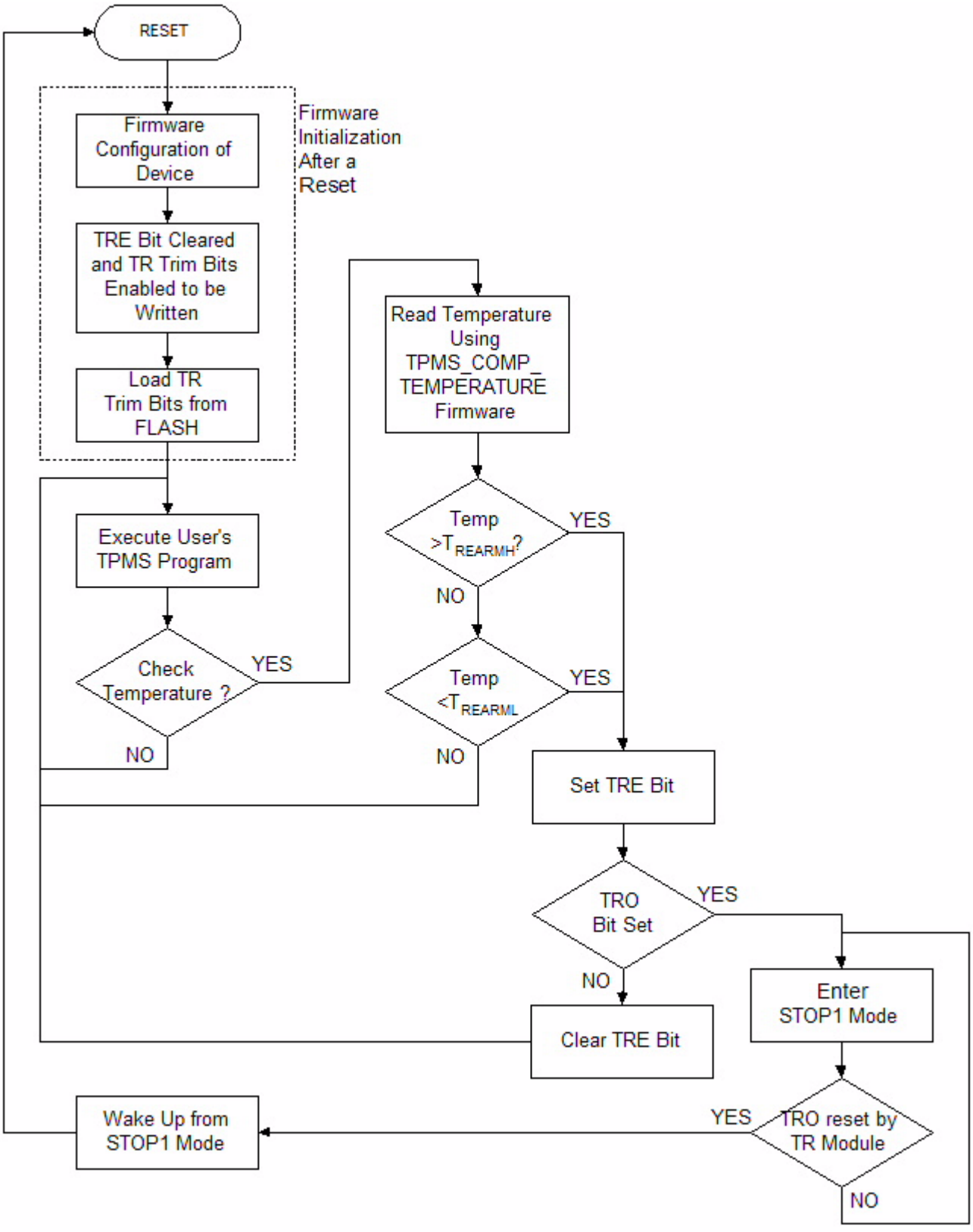


Figure 23. Flowchart for using TR module

11 Periodic Wake-up Timer

The periodic wake-up timer (PWU) generates a periodic interrupt to wake 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.

11.1 Block diagram

The block diagram of the wake-up timer is shown in Figure 24. This consists of a programmable pre-scaler with 64 steps that can be used to adjust for variations in the value of the LFO period. Finally there are two cascaded programmable 6-bit dividers to set wake-up and/or reset time intervals.

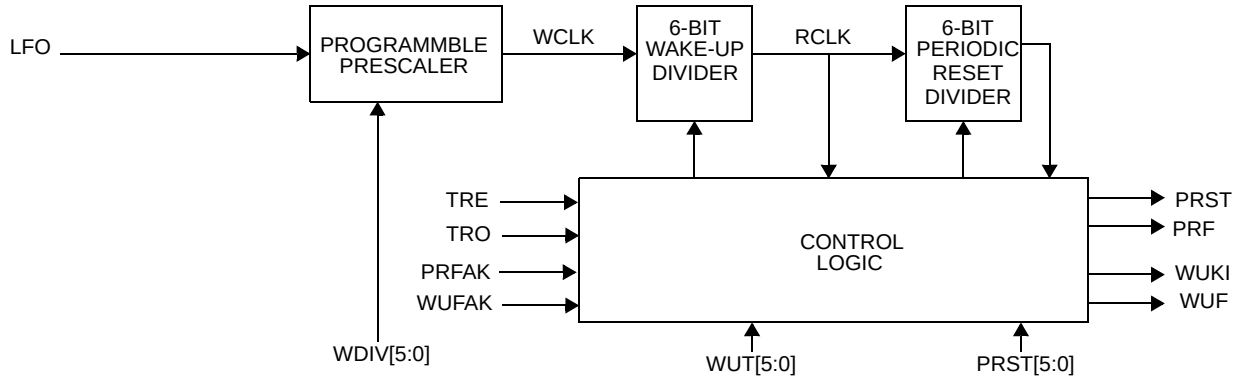


Figure 24. Wake-up timer block diagram

The wake-up divider (PWUDIV) register selects a division of the incoming 1 msec clock to generate a wake-up clock, WCLK. The WCLK frequency can be calibrated against the more precise external oscillator using the TCAL firmware subroutine as described in Section 14. This subroutine turns on the RFM crystal oscillator and feeds a 500 kHz clock to the TPM1 for one cycle of the LFO. The measured time is used to calculate the correct value for the WDIV[5:0] bits for a WCLK period of 1 second. The TCAL subroutine cannot be used while the RFM is transmitting or the TPM1 is being used for another task.

The wake-up time register (PWUSC0) selects the number of WCLK pulses that are needed to generate a wake-up interrupt to the MCU. The periodic reset register (PWUSC1) selects the number of wake-up pulses that are needed to generate a periodic reset of the MCU. Both the wake-up time counter and the periodic reset timer are incrementing counters that generate their interrupt or reset when the desired count is reached and are then reset to zero. Reading the status of either of these counters will return a zero content if done immediately after the interrupt or reset is generated.

If both the reset and the interrupt occur on the same clock cycle the reset will have precedence and the interrupt will not be generated.

In order to prevent wake-up or reset from an extreme temperature event both the wake-up interrupt or periodic reset are disabled if the thermal restart is activated and the TRO bit indicates that the device is still outside of the T_{RESET} range. The wake-up and periodic reset counters will still run. The state of these counters can be read using the PSEL bit in the PWUS register.

NOTE

The wake-up interrupt (WUKI) cannot be masked by clearing the I-bit.

11.2 Wake-up Divider register - PWUDIV

The PWUDIV register contains six bits to select the division of the incoming 1 msec clock period as described in Table 88.

Table 88. PWU Divider register (PWUDIV)

$0038	Bit 7	6	5	4	3	2	1	Bit 0
R	0	0	WDIV[5:0]					
W	—	—						
RESET:	—	—	—	—	—	—	—	—
POR:	0	0	0	1	1	1	1	1

Table 89. PWUDIV Register Field Descriptions

Field	Description
7:6 Unused	**Unused**
5:0 WDIV[5:0]	**Wake-up Divider Value** — The WDIV[5:0] bits select an incoming pre-scaler for the incoming 1 msec clock period from 504 to 1512. This results in a clocking of the 6-bit wake-up divider at rates from a nominal 0.504 to 1.512 sec per wake-up clock, WCLK. The user can use this pre-scaler to fine tune the wake-up time based on the variation in the LFO frequency. The conversion from the decimal value of the WDIV bits to the nominal WCLK period is given as: $t_{WCLK} = \frac{(504 + 16 \times WDIV)}{f_{LFO}}$ A power on reset presets these bits to a value of $1F (decimal 31) which yields a nominal 1 second output period for WCLK. Other resets have no effect on these bits.

11.3 PWU Control/Status 0 register - PWUCS0

The PWUCS0 register contains six bits to select the division of the incoming WCLK clock period and provide interrupt flag and acknowledge bits as described in Table 90. The period of the resulting interrupt also generates the clock, RCLK, for the periodic reset timing.

Table 90. PWU Control/Status 0 register (PWUCS0)

$0039	Bit 7	6	5	4	3	2	1	Bit 0
R	WUF	0	WUT[5:0]					
W		WUFAK						
RESET:	0	—	1	1	1	1	1	1

Table 91. PWUSC0 register field descriptions

Field	Description
7 WUF	**Wake-up Interrupt Flag** — The 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. Reset clears this bit. 0 Wake-up interrupt not generated or was previously acknowledged. 1 Wake-up interrupt generated.
6 WUFAK	**Acknowledge WUF Interrupt Flag** — The 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 No effect. 1 Clear WUF bit.
5:0 WUT[5:0]	**WUF Time Interval** — These control bits select the number of WCLK clocks that are needed before the next wake-up interrupt is generated. The count gives a range of wake-up times from 1 to 63 WCLK clocks. Depending on the value of the bits for the WDIV[5:0] this time interval can nominally be from 1 to 63 seconds in 1 second steps. Whenever the WUT[5:0] bits are changed the timeout period is restarted. Writing the same data to the WUT[5:0] bits has no effect. Writing zeros to all of the WUT[5:0] bits forces the wake-up divider to a value of $3F and disables the wake-up interrupt. However, writing all zeros to the WUT[5:0] bits is inhibited if all of the PRST[5:0] bits are already cleared to zero. This prevents disabling both the periodic wake-up and the periodic reset at the same time. See Table 92. The WUT[5:0] bits are preset to a value of $3F (decimal 63) by any resets.

Table 92. Limitations on clearing WUT/PRST

Control Bits	State of Control Bits	Control Bits to be Cleared	Resulting Action	Resulting Wake-up Interrupt	Resulting Periodic Reset
WUT[5:0]	non-zero	PRST[5:0]	Allowed	Enabled(1)	Disabled
	all zero		Inhibited	Disabled(2)	Enabled(1)
PRST[5:0]	non-zero	WUT[5:0]	Allowed	Disabled	Enabled(1)
	all zero		Inhibited	Enabled(1)	Disabled

1. Using previous values.
2. Wake-up divider preset to $3F.

11.4 PWU Control/Status 1 register - PWUCS1

The PWUSC1 register contains six bits to select the division of the incoming RCLK clock period and provide interrupt flag and acknowledge bits as described in Table 93.

Table 93. PWU Control/Status 1 register (PWUCS1)

$003A	Bit 7	6	5	4	3	2	1	Bit 0
R	PRF	0	PRST[5:0]					
W	—	PRFAK						
RESET:	0	0	1	1	1	1	1	1

Table 94. PWUSC1 register field descriptions

Field	Description
7 PRF	**Periodic Reset Flag**— The 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. This bit is cleared by a power on reset, but is unaffected by other resets. 0 Periodic reset not generated or previously acknowledged. 1 Periodic reset generated.
6 PRFAK	**Acknowledge PRF Interrupt Flag** — 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. Reset has no effect on this bit. 0 No effect. 1 Clear PRF bit.
5:0 PRST[5:0]	**Periodic Reset Time Interval** — These control bits select the number of wake-up interrupts that are needed before the next periodic reset is generated. The decimal count gives a range of periodic reset times from 1 to 63 wake-up interrupts. Depending on the value of the bits for the WDIV[5:0] and WUT[5:0] this time interval can nominally be from 1 second to 66 minutes with steps from 1 to 63 seconds. Whenever the PRST[5:0] bits are changed the timeout period is restarted. Writing the same data to the PRST[5:0] bits has no effect. Writing zeros to all of the PRST[5:0] bits forces the periodic reset to be disabled if at least one of the WUT[5:0] 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[5:0] bits is inhibited if all of the WUT[5:0] bits are already cleared to zero. This prevents disabling both the periodic wake and the periodic reset at the same time. See Table 92. The PRST[5:0] bits are preset to a value of 63 by any resets.

11.5 PWU Wake-up Status register - PWUS

The PWUS register shows the current status of the two PWU counters as described in Table 93. The counter contents are captured when the register is read.

Table 95. PWU Wake-up Status register (PWUS)

$001F	Bit 7	6	5	4	3	2	1	Bit 0
R	PSEL	0	CSTAT					
W		—						
RESET:	0	-	-	-	-	-	-	-

Table 96. PWUS register field descriptions

Field	Description
7 PSEL	**Page Selection** — The PSEL read/write bit selects whether the other bits are read from the WUT or PRST 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 = WUT counter status 1 CSTAT = PRST counter status
6 unused	**Unused** — An unused bit that always reads as a logical zero.
5:0 CSTAT	**Counter Status** — These read-only bits show the status of the counter selected by the PSEL bit. The effects 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.

11.6 Functional modes

PWU module will work in each of the MCU operating modes as follows:

11.6.1 Run mode

If the module generates a wake-up interrupt the PC (Program Counter) will be redirected to the wake-up timer interrupt vector. The WUF flag will be set to indicate wake-up timer interrupt; write 1 to WUFACK to clear this flag.

If the module generates a reset the PC will be redirected to the reset vector. The PRF flag will be set to indicate periodic reset; write 1 to PRFACK to clear this flag.

All registers will continue to hold their programmed values after interrupt or reset is taken.

11.6.2 Stop4 mode

If the module generates a wake-up interrupt the bus and core clocks will be restarted and the PC will be redirected to the wake-up timer interrupt vector. The WUF flag will be set to indicate wake-up timer interrupt, write 1 to WUFACK to clear this flag.

If the module generates a periodic reset the bus and core clocks will be restarted and the PC will be redirected to the reset vector. The PRF flag will be set to indicate periodic reset; write 1 to PRFACK to clear this flag.

All registers will continue to hold their programmed values after interrupt or reset is taken.

11.6.3 Stop1 mode

If the module generates a wake-up interrupt the module will cause the MCU to exit the power saving mode as a POR. MCU will have the wake-up interrupt pending and once CLI opcode is executed PC will be redirected to wake-up interrupt vector address. The WUF flag will be set to indicate wake-up timer interrupt, write 1 to WUFACK to clear this flag.

If the module generates a periodic reset the module will cause the MCU to exit the power saving mode as a POR. The PRF flag will be set to indicate periodic reset; write 1 to PRFACK to clear this flag. The SRS register will have just the POR bit set.

In this stop mode exit all registers will continue to hold their programmed values.

11.6.4 Active BDM/Foreground commands

The PWU is frozen in Active Background Mode or executing foreground commands, so PWU counters will also be stopped. Normal PWU operation will resume as MCU exits BDM or foreground command is finished.

12 Low-Frequency Receiver (LFR)

NOTE

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.

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 can be a period of continuous LF carrier or it can be a series of Manchester coded bits at the nominal 3.906 kbps 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.

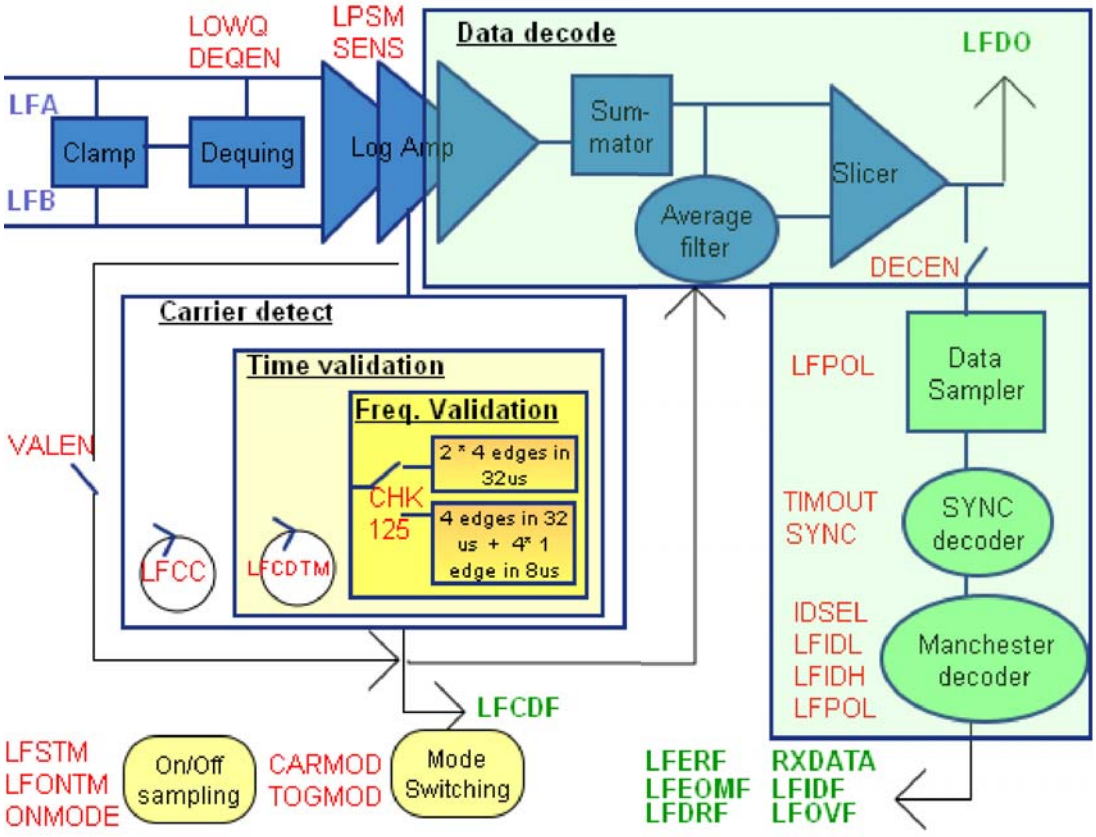


Figure 25. Block diagram

12.1 Features

Major features of the LFR module include:

- Differential input LF detector (two dedicated pins):
  - Selectable sensitivity (three levels).
  - Thresholds trimmed at the factory with trim setting saved in nonvolatile memory.
  - LFR has own 125 kHz 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 preamble requirements
    Continuous carrier
    Manchester zeros preamble
    No preamble
  - 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.

12.2 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.

12.3 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 1kHz 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 1uA 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 1kHz 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 two LFO period before any active operations. In the ON/OFF cycle, those 2 msec are hidden in the sampling time during the off time.

12.4 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 three 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.

12.5 LFR DATA mode states

The modes of operation the LFR state machine will sequence as shown in Figure 25.

12.6 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 three 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.

12.7 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 μsec. 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 μsec 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.

12.8 Data recovery

Rectified signals from the amplifier output are connected to the input of an averaging filter and data slicer. The slicer thus 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 mVpp 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.

12.9 Data clock recovery and synchronization

Data clock recovery and synchronization takes place before or during the SYNC portion of an incoming message. The preamble can be nothing, a carrier only or 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 kbps 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.

12.10 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 26. 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.

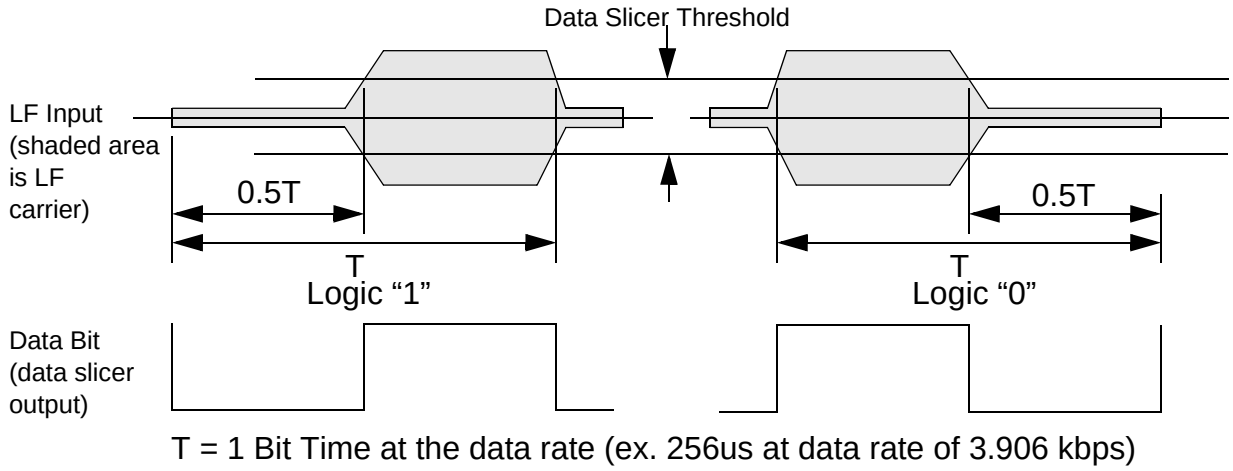


Figure 26. Manchester Encoded Datagram for LFPOL = 0

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 27.

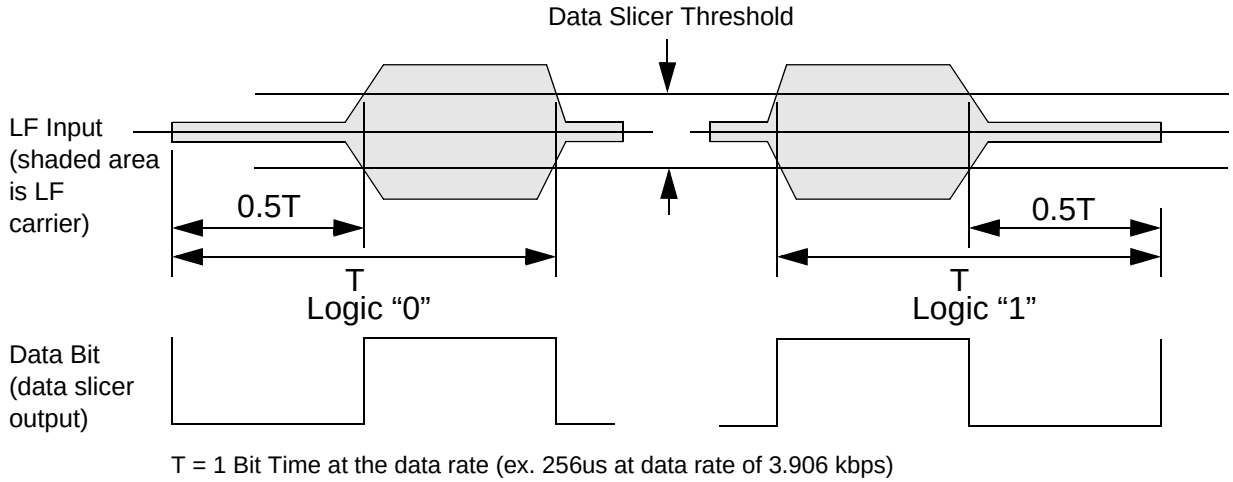


Figure 27. Manchester encoded datagram for LFPOL = 1

12.11 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 28.

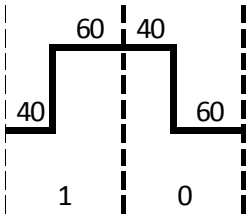


Figure 28. Definition of duty-cycle of 40%

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 29.

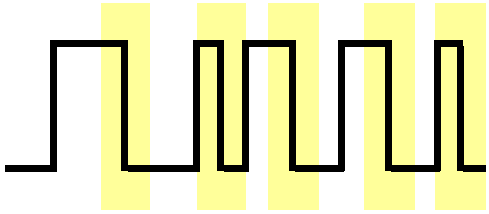


Figure 29. Impact of duty-cycle on SYNC pattern

12.12 Input signal envelope

The combination of the external LF antenna and any external components as shown in Figure 30 should not significantly filter the envelope of the LF carrier as shown in Figure 31. Excessive filtering will cause the received message error rate (MER) to increase.

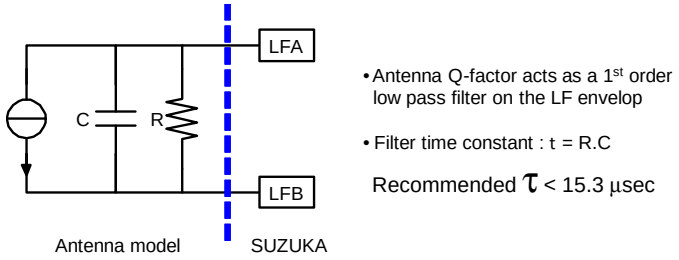


Figure 30. Antenna Q-factor equivalent model for the LF envelope

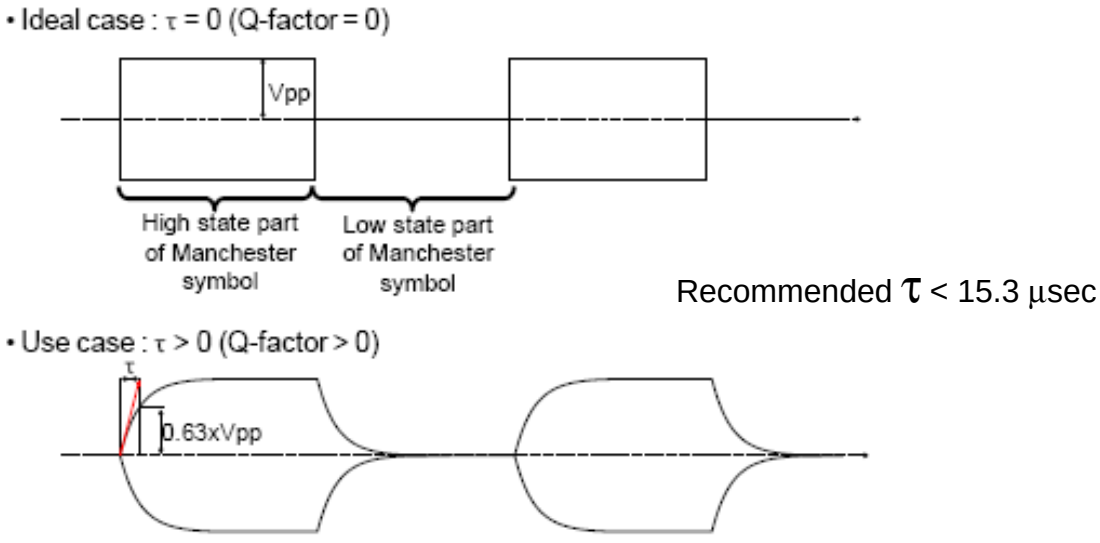


Figure 31. LF envelope filtering

12.13 Telegram verification

The LFR has control bits to allow flexibility in the telegram format and protocol to allow the LFR to adapt to a variety of 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 an optional preamble, 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 32.

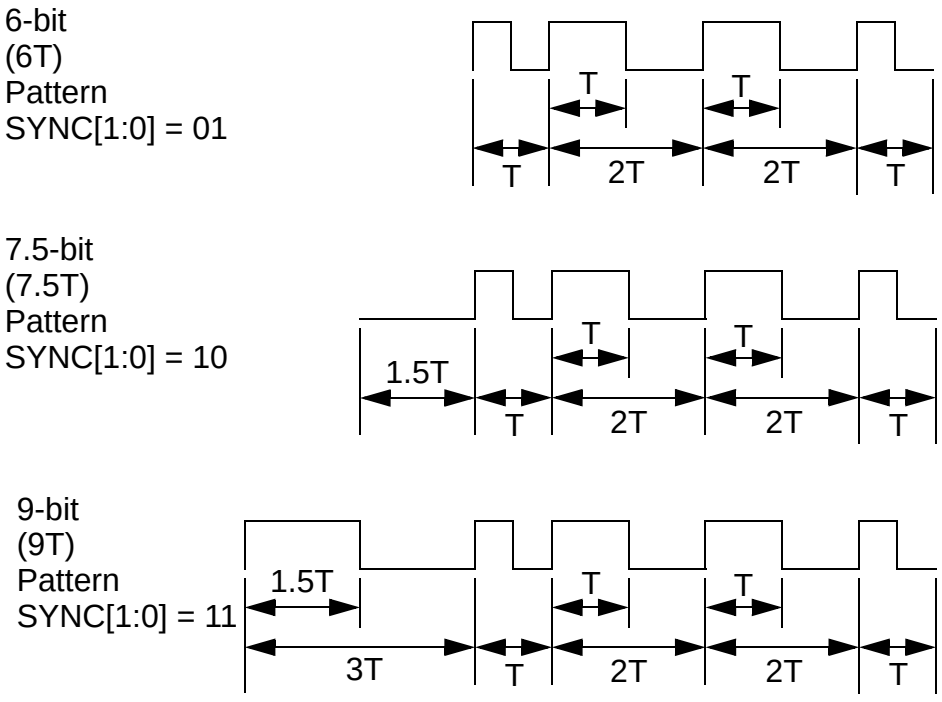


Figure 32. SYNC patterns

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 thus started after a carrier detection and will stop the receiver is reaching the programmed value selected by the TIMOUT[1:0] bits in the LFCTL4 register. This time out counter is clocked by the internal LFRO 125 kHz 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 before the SYNC pattern as shown in Figure 33. 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.

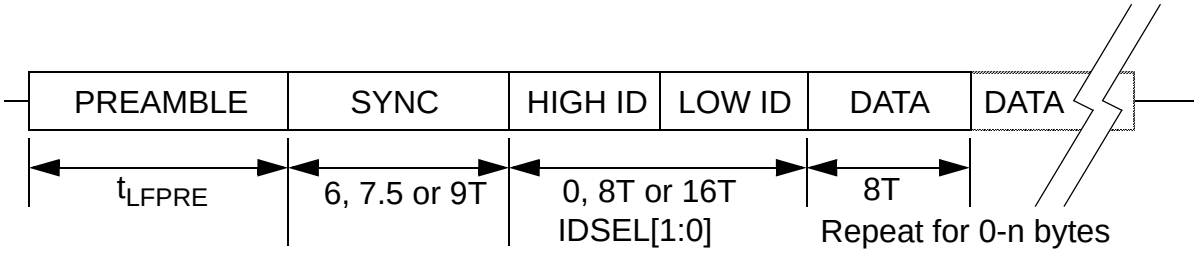


Figure 33. Telegram format (carrier preamble)

12.14 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.

12.15 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.

12.16 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.

12.17 LFR register definition

The LFR module uses eight addresses in the MCU memory map for data, control, and status registers.This section consists of register descriptions.

NOTE

Each control register (LFCTLx) should be modify when the LF is off (LFEN = 0). Modification of the control registers “on-the-fly” might lead to unknown state.

NOTE

Each turn off of the LFR (LFEN = 0) should be followed by at least two LFO cycles before trying to restart the LFR (LFEN = 1).

12.17.1 LF Control 1 register (LFCTL1)

LFCTL1 contains the main LF enable control, detection protocol format controls, and input sensitivity controls. The LFCTL1 register also contains a register select bit, LPAGE.

Table 97. LFR Control 1 register (LFCTL1)

$0020	Bit 7	6	5	4	3	2	1	Bit 0
R	LFEN	0	CARMOD	LPAGE	IDSEL[1:0]		SENS[1:0]	
W		SRES						
Reset:	0	0	0	0	0	0	0	0

Table 98. LFCTL1 register field descriptions

Field	Description
7 LFEN	**LF 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 µsec and then the LFR is ready to receive signals. 0 LF receiver in standby. 1 LF receiver active.
6 SRES	**Soft Reset**— 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	**Carrier Mode**— This read/write control bit selects the basic operating mode for the LFR. 0 Data receive mode. 1 Carrier detect mode - wake the MCU when a carrier signal is detected if LFCDIE is set.
4 LPAGE	**LFR Page Select** — This read/write bit is used is used to select the register page access. The LPAGE bit has no effect on the LFCTL1 and LFCTL2 registers. This bit is cleared by LFR reset. 0 Access page 0. 1 Access page 1.
3:2 IDSEL[1:0]	**Wake-up ID selection** — Selects the existence and length of the wake-up ID. Reset clears these bits. 00 No ID expected 01 8-bit ID based on the contents of the LFIDL register 10 16-bit ID based on the contents of the LFIDH and LFIDL registers 11 8-bit ID matches the contents of either the LFIDH or LFIDL registers

Table 98. LFCTL1 register field descriptions (continued)

Field	Description
1:0 SENS[1:0]	**Sensitivity control** — These two read/write control bits 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 S_{NODET_x} level, no signal will be detected. If the level is above S_{DET_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. 00 Very low sensitivity (S_{DET_VL}; S_{NODET_VL}) 01 Low sensitivity (S_{DET_L}; S_{NODET_L}) 10 High sensitivity (S_{DET_H}; S_{NODET_H}) 11 Performance not guaranteed - not recommended.

12.17.2 LF Control 2 register (LFCTL2)

LFCTL2 contains the selection bits for the length of the LF sampling ON time and the time interval between samples as shown in Figure 99.

Table 99. LFR control 2 register (LFCTL2)

$0021	Bit 7	6	5	4	3	2	1	Bit 0
R W	LFSTM[3:0]				LFONTM[3:0]			
Reset:	0	1	1	0	0	0	0	0

Table 100. LFCTL2 register Field Descriptions

Field	Description
7:4 LFSTM [3:0]	**LF Sampling Time Interval Select**— These read/write control bits 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. A reset of the LFR results in the value being set to binary 0110. 0000 Continuous ON mode (see Section 12.15) 0001 Sampled decoding mode every 16 LFO clock periods (16 milliseconds nominal) 0010 Sampled decoding mode every 32 LFO clock periods (32 milliseconds nominal) 0011 Sampled decoding mode every 64 LFO clock periods (64 milliseconds nominal) 0100 Sampled decoding mode every 128 LFO clock periods (128 milliseconds nominal) 0101 Sampled decoding mode every 250 LFO clock periods (250 millisecond nominal) 0110 Sampled decoding mode every 500 LFO clock periods (0.5 seconds nominal) 0111 Sampled decoding mode every 1000 LFO clock periods (1 seconds nominal) 1000 Sampled decoding mode every 2000 LFO clock periods (2 seconds nominal) 1001 Sampled decoding mode every 4000 LFO clock periods (4 seconds nominal) 1010-1xxxContinuous ON mode (see Section 12.15)

Table 100. LFCTL2 register Field Descriptions (continued)

Field	Description
3:0 LFONTM [3:0]	**LF Sampling ON Time Select** — These read/write control bits 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 msec) included in the OFF time prior to the sample ON time (see Figure 34). 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. Reset forces the LFONTM bits to 0:0:0. 0000 1 LFO clock cycle (1 millisecond nominal) 0001 2 LFO clock cycle (2 milliseconds nominal) 0010 4 LFO clock cycle (4 milliseconds nominal) 0011 8 LFO clock cycle (8 milliseconds nominal) 0100 16 LFO clock cycle (16 milliseconds nominal) 0101 32 LFO clock cycle (32 milliseconds nominal) 0110 64 LFO clock cycle (64 milliseconds nominal) 0111 128 LFO clock cycle (128 milliseconds nominal) 1000 256 LFO clock cycle (256 milliseconds nominal) 1001 512 LFO clock cycle (512 milliseconds nominal) 1010 1024 LFO clock cycles (1024 milliseconds nominal) 1011 1024 LFO clock cycles (1024 milliseconds nominal) 11xx 1024 LFO clock cycles (1024 milliseconds nominal) **NOTE:** The LFONTM selected time must be less than the LFSTM selected time, otherwise the Continuously ON mode is present.

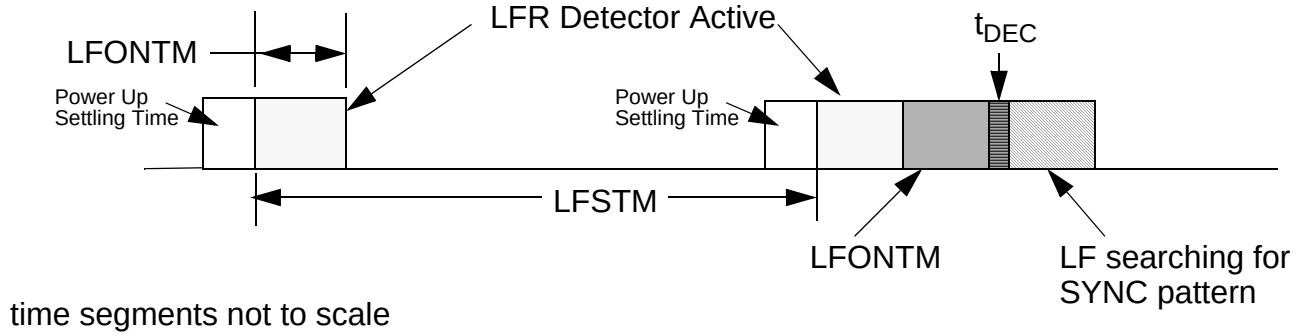


Figure 34. LF detector sampling timing

12.17.3 LF Control 3 register (LFCTL3)

LFCTL3 contains the control bits for the LF sampling interval and the minimum required carrier detection time when using the carrier detect mode.

Table 101. LFR Control 3 register (LFCTL3)

$0022	Bit 7	6	5	4	3	2	1	Bit 0
R	LFDO	TOGMOD	SYNC[1:0]		LFCDTM[3:0]			
W	—							
Reset:	—	0	0	1	0	0	0	0

Table 102. LFCTL3 register field descriptions

Field	Description
7 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. This bit is read only and unaffected by any reset. 0 LF detector output low (no signal above threshold) 1 LF detector output high (received signal above threshold)
6 TOGMOD	**LFR Mode Toggle** — This read/write bit enables the toggling of the CARMOD bit at each new LFON sequence. Reset clears this bit. 0 CARMOD bit does not change and determines detector mode. 1 CARMOD bit will be toggled every LFON detection sequence, starting by CARMOD selection. Therefore the reception chain will alternately look for a carrier frame or for a data frame.
5:4 SYNC[1:0]	**LF SYNC Selection** — Selects the type of SYNC pattern as described in Figure 32. Reset presets these bits to the 01 (6T SYNC) option. 00 For factory test purposes, not intended for use in any application. 01 6T SYNC pattern required (compatible with no preamble, carrier preamble, or Manchester 0 or 1 preamble). 10 7.5T SYNC pattern required (compatible with carrier preamble or Manchester 0 or 1 preamble). 11 9T SYNC pattern required (compatible with Manchester 0 or 1 preamble).
3:0 LFCDTM [3:0]	**LF Carrier Detect Time** — These read/write control bits 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). 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. Reset of the LFR results in LFCDTM[3:0] being reset to 0:0:0:0. The resulting carrier detect times are defined by the following number of carrier periods needed to validate the carrier, with the corresponding time for a carrier at 125 kHz in parenthesis: 0000 Carrier detect = 8 (64 µsec)Data mode detect = 8 (64 µsec) 0001 Carrier detect = 16 (128 µsec)Data mode detect = 8 (64 µsec) 0010 Carrier detect = 32 (256 µsec)Data mode detect = 8 (64 µsec) 0011 Carrier detect = 64 (512 µsec)Data mode detect = 8 (64 µsec) 0100 Carrier detect = 128 (1024 µsec)Data mode detect = 8 (64 µsec) 0101 Carrier detect = 256 (2048 µsec)Data mode detect = 8 (64 µsec) 0110 Carrier detect = 512 (4096 µsec)Data mode detect = 8 (64 µsec) 0111 Carrier detect = 1024 (8192 µsec0Data mode detect = 8 (64 µsec) 1000 Carrier detect = 8 (64 µsec)Data mode detect = 8 (64 µsec) 1001 Carrier detect = 16 (128 µsec)Data mode detect = 16 (128 µsec) 1010 Carrier detect = 32 (256 µsec)Data mode detect = 32 (256 µsec) 1011 Carrier detect = 64 (512 µsec)Data mode detect = 64 (512 µsec) 1100 Carrier detect = 128 (1024 µsec)Data mode detect = 128 (1024 µsec)(1) 1101 Carrier detect = 256 (2048 µsec)Data mode detect = 256 (2048 µsec)(1) 1110 Carrier detect = 512 (4096 µsec)Data mode detect = 512 (4096 µsec)(1) 1111 Carrier detect = 1024 (8192 µsec)Data mode detect = 1024 (8192 µsec)(1)

1. The auto-zero sequence needs to be performed every 1 msec. Therefore LFR detection times of 1024, 2048, 4096 and 8192 msec the auto-zero sequence will be done at each 1 msec interval. This auto-zero sequence lasts for 64 msec. If the carrier is detected again at the end of the auto-zero sequence it is assumed that the carrier was there for the complete 64 msec period of the auto-zero.

12.17.4 LFR Control 4 register (LFCTL4)

LFCTL4 contains local interrupt enable control bits. The provided I-interrupts are not globally masked by the I bit in the CPU's CCR, setting one or more of these interrupt enable control bits will cause a CPU interrupt to be requested whenever the flag bit associated with the corresponding LFR interrupt source becomes set. It is good practice to clear any flag bits in the LFS register before setting interrupt enable bits in this register in order to avoid an immediate interrupt request.

Table 103. LFR Control 4 register (LFCTL4)

$0023	Bit 7	6	5	4	3	2	1	Bit 0
R W	LFDRIE	LFERIE	LFCDIE	LFIDIE	DECEN	VALEN	TIMOUT[1:0]	
Reset:	0	0	0	0	1	1	0	0

Table 104. LFCTL4 register field descriptions

Field	Description
7 LFDRIE	**LFR Data Register Full Interrupt Enable** — This read/write bit enables interrupts to be requested when the LFR data register is full. Reset clears LFDRIE. 0 LFDRF interrupts disabled. Use software polling. 1 LFR Data Register Full interrupts are enabled. If LFDRIE is set, then an interrupt is requested when LFDRF = 1.
6 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. Reset clears LFERIE. 0 LFERF interrupts disabled. Use software polling. 1 LFERF interrupts are enabled. If LFERIE is set, then an interrupt is requested when LFERF = 1.
5 LFCDIE	**LFR Carrier Detect Interrupt Enable** — This read/write bit enables the LFCDF interrupt when the LFR detects the number of samples with an LF signal defined by the LFCDTM bits in the LFCTL3 register. The LFCDIE is ignored when the LFR is operating in the data mode (CARMOD = 0), except when DECEN is cleared. Reset clears LFCDIE. 0 LFCDF interrupts disabled. Use software polling. 1 LFR LFCDF interrupts are enabled. If LFCDIE is set, then an interrupt is requested when LFCDF = 1.
4 LFIDIE	**LFR ID Detect Interrupt Enable** — This read/write bit enables interrupts to be requested when the LFR detects a match to the ID code selected in the LFIDH:L registers. Reset clears LFIDIE. 0 LFIDF interrupts disabled. 1 LFIDF interrupts are enabled. If LFIDIE is set, then an interrupt is requested when LFIDF = 1.
3 DECEN	**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). The MCU should then poll the LFDO bit to extract from the analog detector the bit stream (may also referred to as the MCU Direct Mode in the Reims product). Reset sets the DECEN bit. 0 Digital decoder is disabled. 1 Digital decoder is enabled.
2 VALEN	**LF Validation Enable** — This read/write bit enables the carrier validation process. Reset sets this bit. 0 Carrier Validation disabled. 1 Carrier Validation enabled.
1:0 TIMOUT [1:0]	**SYNC Time Out Select** — These two read/write bits 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 125 kHz LFRO clock. Reset clears TIMOUT bit. 00 SYNC word is continuously search, no timeout. 01 SYNC search time set to nominal 8 milliseconds. 10 SYNC search time set to nominal 24 milliseconds. 11 SYNC search time set to nominal 48 milliseconds.

12.17.5 LFR Status register (LFS, LPAGE = 0)

LFS contains the data ready status flags. It is only accessible when the LPAGE bit is clear.

.

Table 105. LFR Status register (LFS, LPAGE =0)

$0024	Bit 7	6	5	4	3	2	1	Bit 0
R	LFDRF	LFERF	LFCDF	LFIDF	LFOVF	LFEOMF	LPSM	0
W	—	—	—	—	—	—		LFIAK
Reset:	0	0	0	0	0	0	1	0

Table 106. LFS register field descriptions

Field	Description
7 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. LFDRF is also cleared by reset. 0 No new data in LFDATA register. 1 A new byte of data has been received and can be read from the LFDATA register.
6 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. 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. LFERF is also cleared by reset. 0 Normal operation. 1 Error detected in the Manchester data mode.
5 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. Clear LFCDF by writing a one to the LFIAK bit. LFCDF is also cleared by reset. 0 Normal operation. 1 Carrier detection has occurred.
4 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. LFIDF is also cleared by reset. 0 Normal operation. 1 Wake-up ID has been detected.
3 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. LFOVF is also cleared by reset. 0 Normal operation. 1 Previous data over-written before MCU read it.
2 LFEOMF	**LF Receive Data EOM 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. LFEOMF is also cleared by reset. 0 No EOM detected. 1 EOM detected.
1 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 200 µs in transition from carrier to data mode. LPSM is set by reset. 0 Low time transition from carrier to data mode 1 Low consumption during sniff mode
0 LFIAK	**LF Interrupt Acknowledge** — 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. Reset has no effect on this bit. 0 No effect. 1 Clears the LFDRF, LFERF, LFCDF, LFIDF, LFOVF and LFEOMF flag bits.

12.17.6 LFR Data register (LFDATA, LPAGE = 0)

The LFDATA is a read-only register that contains the most recent received data value. It is only accessible when the LPAGE bit is clear. 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.

Table 107. LFR Data register (LFDATA) when LPAGE = 0

$0025	Bit 7	6	5	4	3	2	1	Bit 0
R	RXDATA[7:0]							
W	—	—	—	—	—	—	—	—
Reset:	0	0	0	0	0	0	0	0

Table 108. LFDATA register field descriptions

Field	Description
7:0 RXDATA[7:0]	**Receive Data [7:0]** — This is the received data from the LFR when in the data mode. All bits are read-only and any writes to these bits will be ignored. Reading this register will clear the LFDRF.

12.17.7 LFR ID registers (LFIDH:LFIDL, LPAGE = 0)

These two 8-bit read/write registers hold one of two ID values for LF messages. They are only accessible when the LPAGE bit is clear. 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, the ID value received through the LFR must match the contents of the LFIDH and/or LFIDL registers depending on the IDSEL bits or the message will be ignored and the MCU will remain in standby mode to minimize power consumption. All these bits are cleared by a reset.

Table 109. LFR ID Low Byte (LFIDL)

$0026	Bit 7	6	5	4	3	2	1	Bit 0
R W	ID[0:7]							
Reset:	0	0	0	0	0	0	0	0

Table 110. LFR ID High Byte (LFIDH)

$0027	Bit 7	6	5	4	3	2	1	Bit 0
R W	ID[15:8]							
Reset:	0	0	0	0	0	0	0	0

Table 111. LFR ID register field description

Field	Description
ID[15:0]	**ID bits 15 through 0** — These read/write bits contain bits 15 through 0 of the 16-bit ID value.

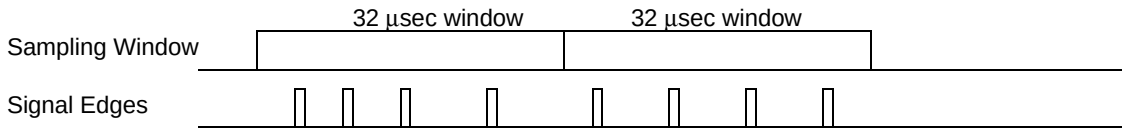
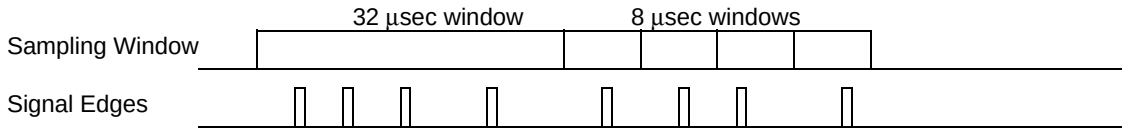
12.17.8 LFR Control D register (LFCTRLD, LPAGE =1)

The LFCTRLD register contains two control bits for the LF detector and decoder. It is only accessible when the LPAGE bit is set.

Table 112. LFR Control D register (LFCTRLD, LPAGE =1)

$0022	Bit 7	6	5	4	3	2	1	Bit 0
R W	—	—	—	—	—	—	ONMODE	CHK125
Reset:	0	0	0	0	0	0	0	0

Table 113. LFCTRLD register field descriptions

Field	Description
7-2 Reserved	**Reserved bits** — Not for user access.
1 ONMODE	**ON Mode Behavior** — This read/write bits selects how an error will affect the ON time. This bit is cleared by reset. 0 Any error will stop the ON time and will wait for a complete OFF time to restart. 1 If there is any remaining ON time, any error will turn the LFR off for up to 7 LFO cycles. before going back to the sniff mode for the remainder of the selected ON time.
0 CHK125	**125 kHz Carrier Check** — This read/write bits controls the CARVAL frequency check method. This bit is cleared by reset. 0 CARVAL validates on 2 times 4 edges inside a 32 µsec window.  1 CARVAL validates on 1 times 4 edges inside a 32 µsec window and then 4 times 1 edge inside an 8 µsec window. 

12.17.9 LFR Control C register (LFCTRLC, LPAGE =1)

The LFCTRLC register contains control bits for the LF detector and decoder. It is only accessible when the LPAGE bit is set.

Table 114. LFR Control C register (LFCTRLC, LPAGE =1)

$0023	Bit 7	6	5	4	3	2	1	Bit 0
R W	—	—	—	—	—	LOWQ[1:0]		DEQEN
Reset:	0	1	1	0	1	0	0	0

Table 115. LFCTRLC register field descriptions

Field	Description
7-3 Reserved	**Reserved bits** — Not for user access. The LF Enable firmware routine needs to be executed to set the correct reset value.
2-1 LOWQ [1:0]	**High Level Attenuator Select** — These read/write bits select the resistor added in parallel to the LFR input network. These bits are cleared by reset. 00 4.2 kΩ input loading 01 2.2 kΩ input loading 10 1.2 kΩ input loading 11 700 Ω input loading

Table 115. LFCTRLC register field descriptions (continued)

Field	Description
0 DEQEN	**High Level Attenuator System Enable** — This bit controls the High Level Attenuator system. The bit is cleared by a reset. 0 High Level Attenuator system disabled. 1 High Level Attenuator system enabled.

12.17.10 LFR Control B register (LFCTRLB, LPAGE = 1)

The LFCTRLB register contains control bits for the LF detector and decoder. It is only accessible when the LPAGE bit is set

.

Table 116. LFR Control B register (LFCTRLB, LPAGE =1)

$0024	Bit 7	6	5	4	3	2	1	Bit 0
R W	—	—	—	—	LFPOL	—	—	—
Reset:	1	1	0	0	0	1	0	0

.

Table 117. LFCTRLB register field descriptions

Field	Description
7-4 Reserved	**Reserved bits** — Not for user access.
3 LFPOL	**LF Manchester Polarity Select** — 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. Reset clears LFPOL bit. 0 Zero is falling edge in middle of a bit time, one is a rising edge in the middle of bit time. 1 Zero is rising edge in middle of a bit time, one is a falling edge in the middle of bit time.
2-0 Reserved	**Reserved bits** — Not for user access.

12.17.11 LFR Control A register (LFCTRLA, LPAGE = 1)

The LFCTRLA register contains control bits for the LF detector and factory test selects. It is only accessible when the LPAGE bit is set.

Table 118. LFR Control A register (LFCTRLA, LPAGE =1)

$0025	Bit 7	6	5	4	3	2	1	Bit 0
R W	—	—	—	—	LFCC[3:0]			
Reset:	0	0	0	0	0	0	0	0

Table 119. LFCTRLA register field descriptions

Field	Description
7-4 Reserved	**Reserved bits** — Not for user access.
3-0 LFCC[3:0]	**LF Successive Carrier Validations Counter**— 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 start up 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 TOGMOD = 1.

13 RF Module

NOTE

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 MPXx85/86xxD consists of an RF module (RFM) with crystal 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 overall block diagram is shown in Figure 35.

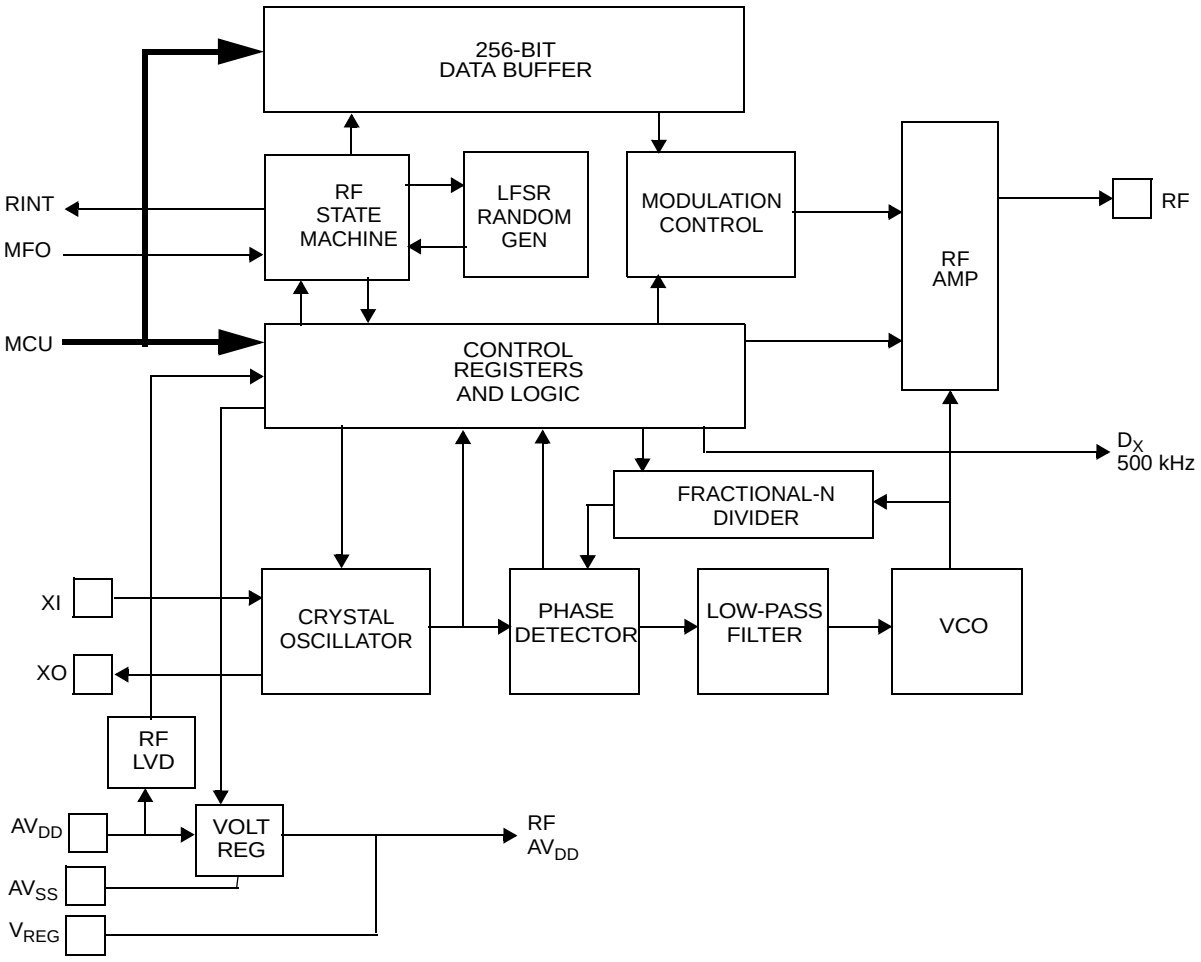


Figure 35. RF transmitter block diagram

13.1 RF Data Modes

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

13.1.1 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 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 as described in Section 13.17.

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.

NOTE

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.

13.1.2 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.

NOTE

The accuracy of the data rate in the MCU direct mode will depend on the accuracy of the HFO clock.

13.2 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 2 trailing bits for an end-of-message, EOM, as shown in Figure 36. 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.

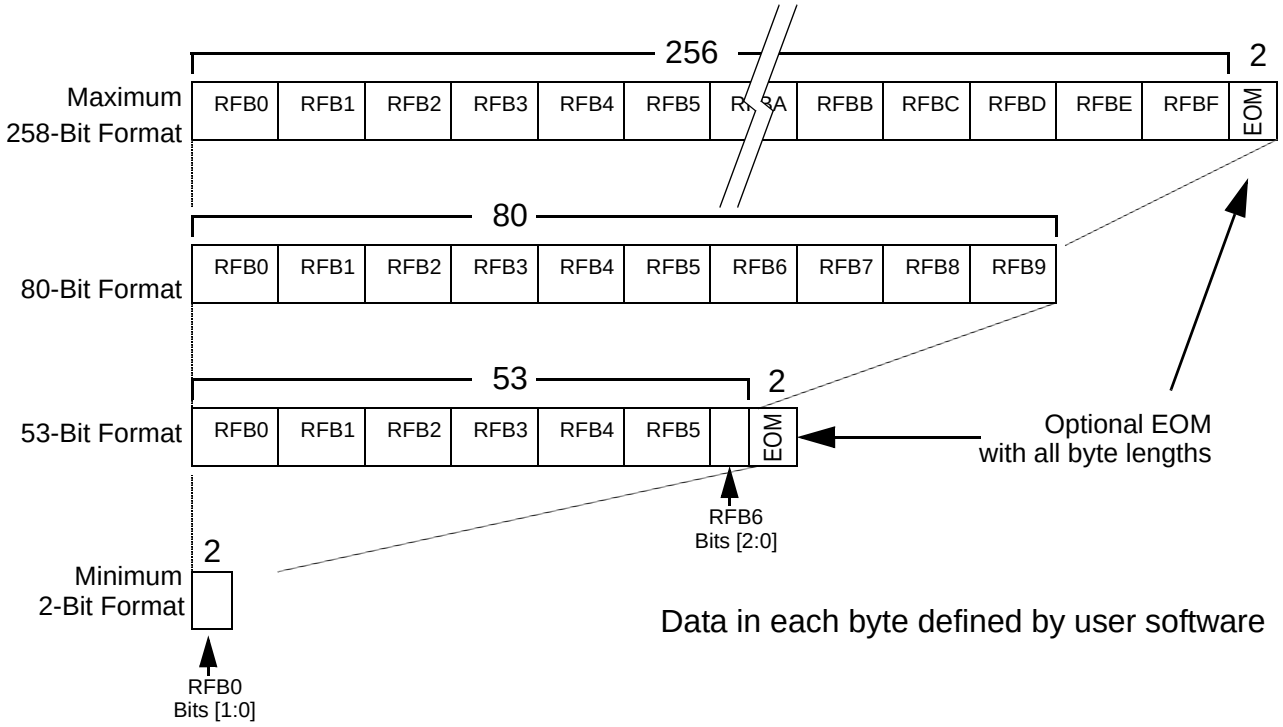


Figure 36. Data frame formats

13.2.1 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.

13.2.2 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 2 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 will be added to the transmission.

13.3 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 as shown in Figure 37.

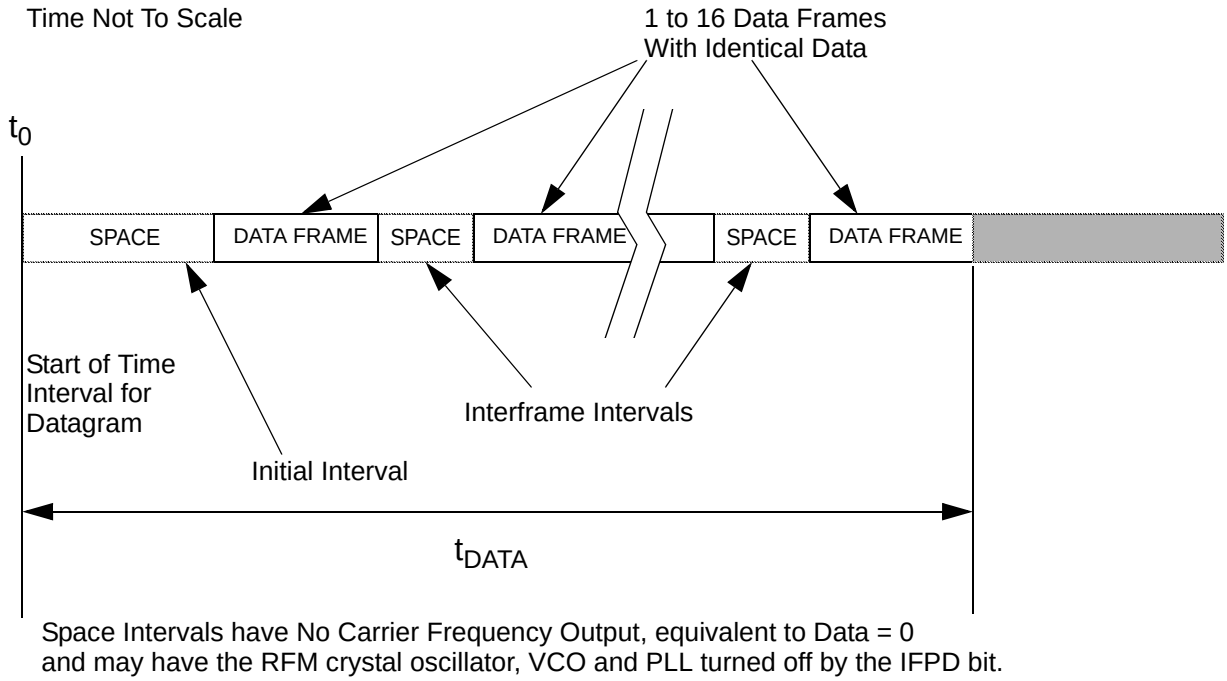


Figure 37. Datagram overview

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 as shown in Figure 38. The initial time interval can be done by adjusting the start time using the MCU or using this interval timing generator.

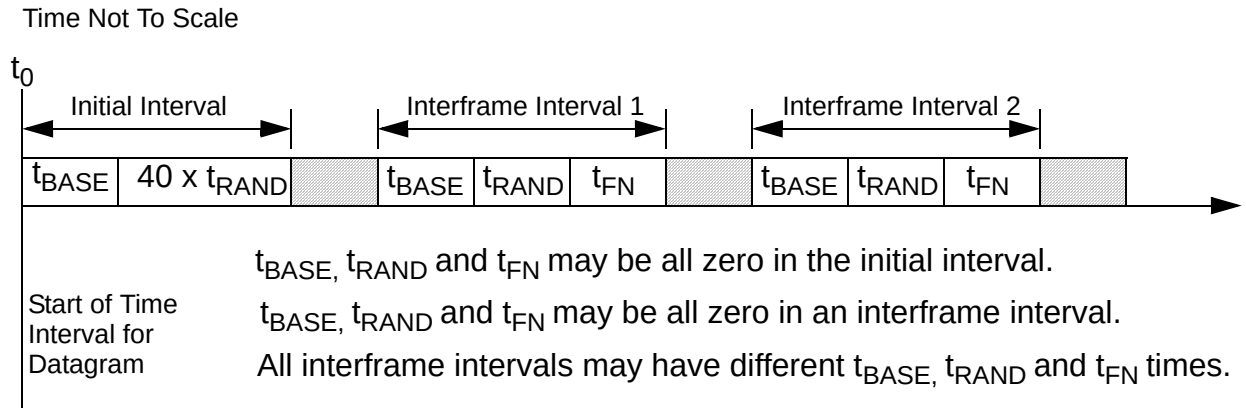


Figure 38. Initial and interframe timing

13.3.1 Initial time interval

When generating an initial time interval the MCU loads the RFM interval generator variables and then goes into the Stop1 mode. When the initial time interval ends the data in the RFM data buffer is automatically sent and the MCU will wake-up at the end of the transmission. The initial time interval is made up of two components:

$$t_{INIT} = t_{BASE} + 40 \times t_{RAND} \quad \textit{Eqn. 14}$$

where:

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

t_{BASE} = Base time in msec

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

The components of this time are described in the following sections.

13.3.2 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-up at the end of the transmission. The interframe time interval is made up of two components:

$$t_{IFRM} = t_{BASE} + t_{RAND} + t_{FN} \quad \textit{Eqn. 15}$$

where:

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

t_{BASE} = Base time in msec

t_{FN} = Time adder in msec for frame number

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

The components of this time are described in the following sections.

13.3.3 Base time interval

The base time interval, t_{BASE}, is used in the initial time interval and in datagram transmissions with 2 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 is 0 to 255 msec using a clock generated from the MFO divided by 125.

13.3.4 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 as shown in Figure 39.

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 msec 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.

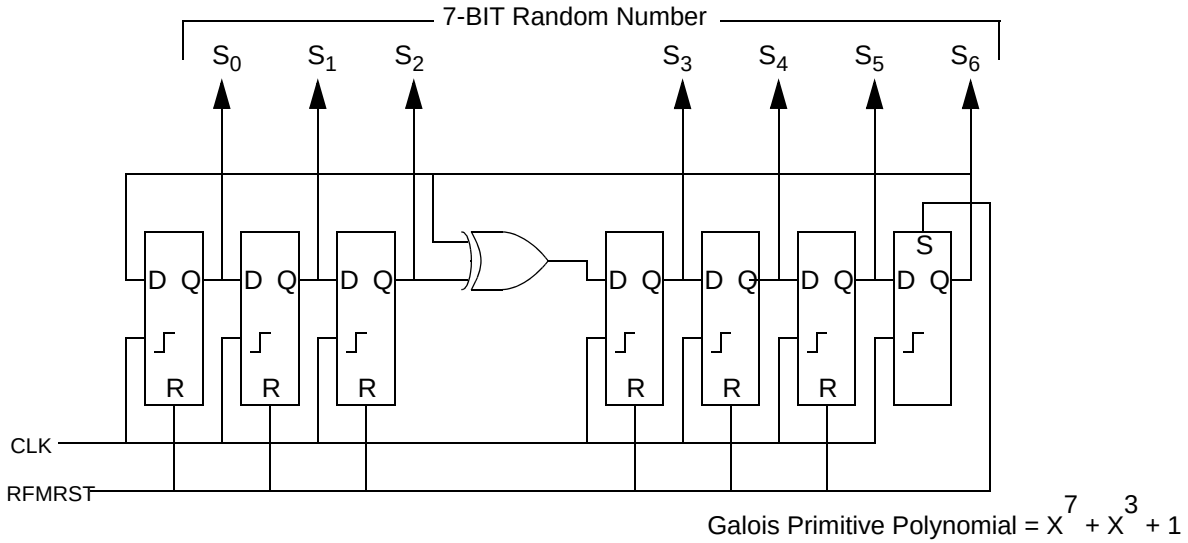


Figure 39. LFSR implementation

NOTE

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 120 for both the design center and the variation resulting from the tolerance of the MFO clock.

Table 120. Randomization interval times

Time interval	Randomization number	Ideal time interval (msec)	Time interval including MFO tolerance (msec)	
			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

13.3.5 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 msec and the number of the frame that was just transmitted as given in Table 121. 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 msec 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.

Table 121. Frame number interval times

Value of FNUM[3:0]	Number of frames	Frame interval where time added	Nominal frame number time interval added (msec)	
			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

Table 121. Frame number interval times (continued)

Value of FNUM[3:0]	Number of frames	Frame interval where time added	Nominal frame number time interval added (msec)	
			Minimum	Maximum
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

13.4 RFM in Stop1 mode

The entire RF transmitter digital section can remain powered up, if enabled by the RFEN bit (see Section 5.3), when the MCU goes into the Stop1 mode.

13.5 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.

13.5.1 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 40. When the POL bit is set, then a logical LOW is defined as an decrease in signal in the middle of a bit time and a logical HIGH is defined as a increase in signal in the middle of a bit time as shown in Figure 41. 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.

13.5.2 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 42. 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 43. 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.

13.5.3 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.

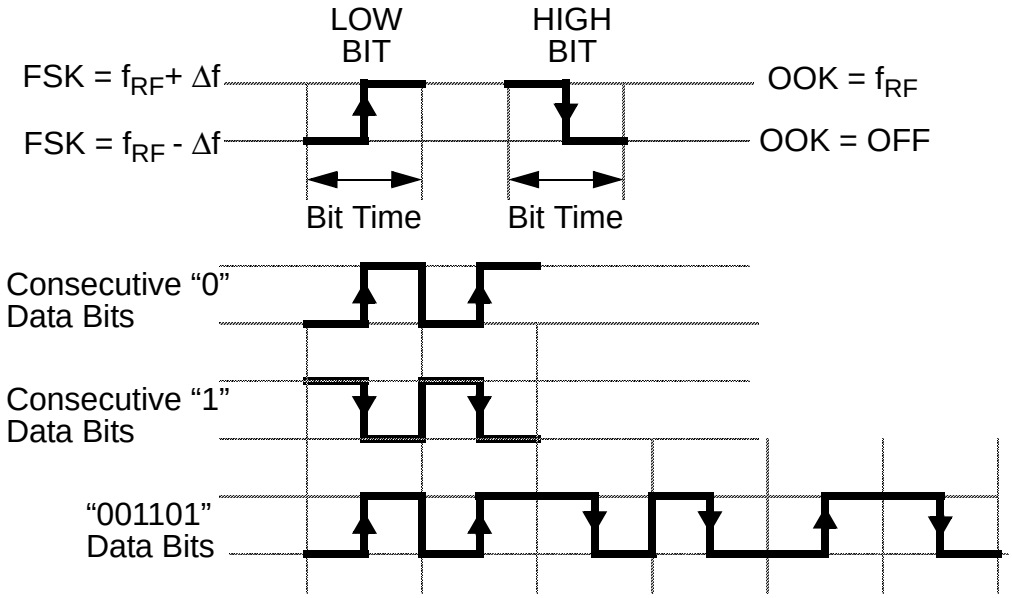


Figure 40. Manchester data bit encoding (POL = 0)

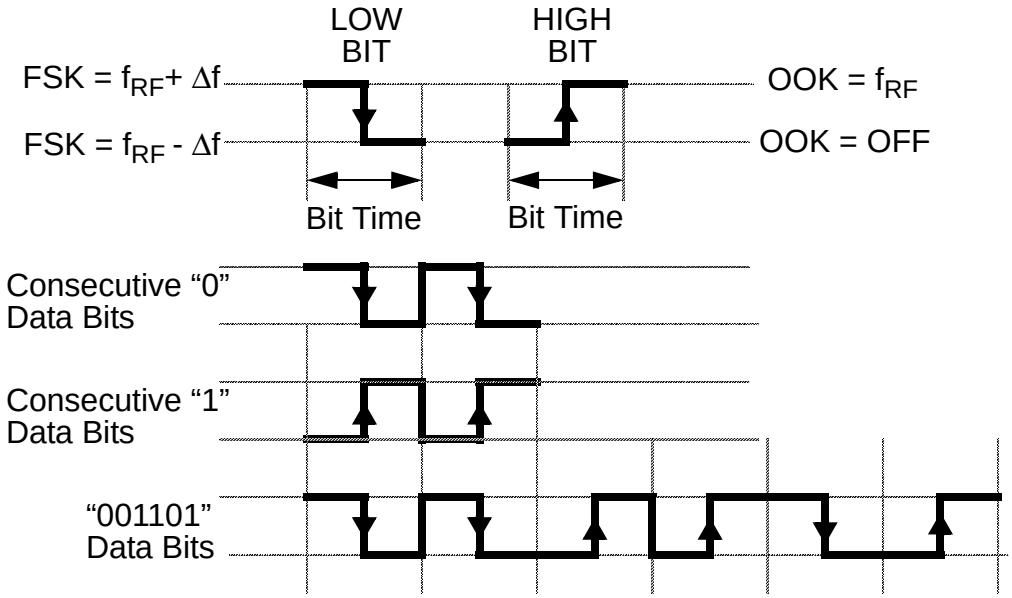


Figure 41. Manchester data bit encoding (POL = 1)

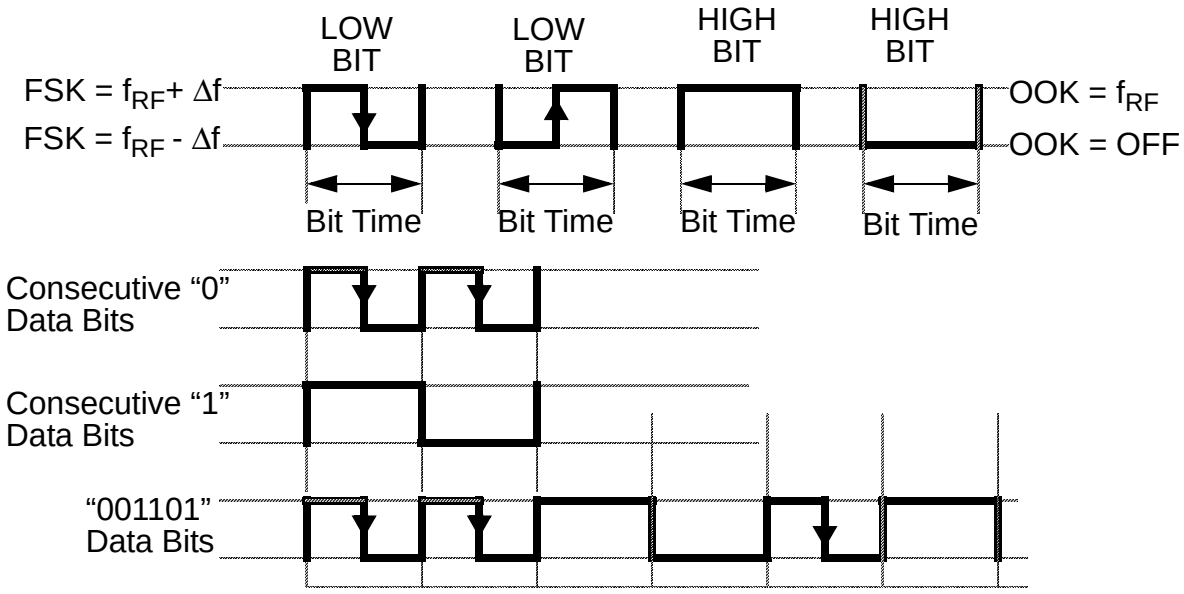


Figure 42. Bi-Phase data bit encoding (POL = 0)

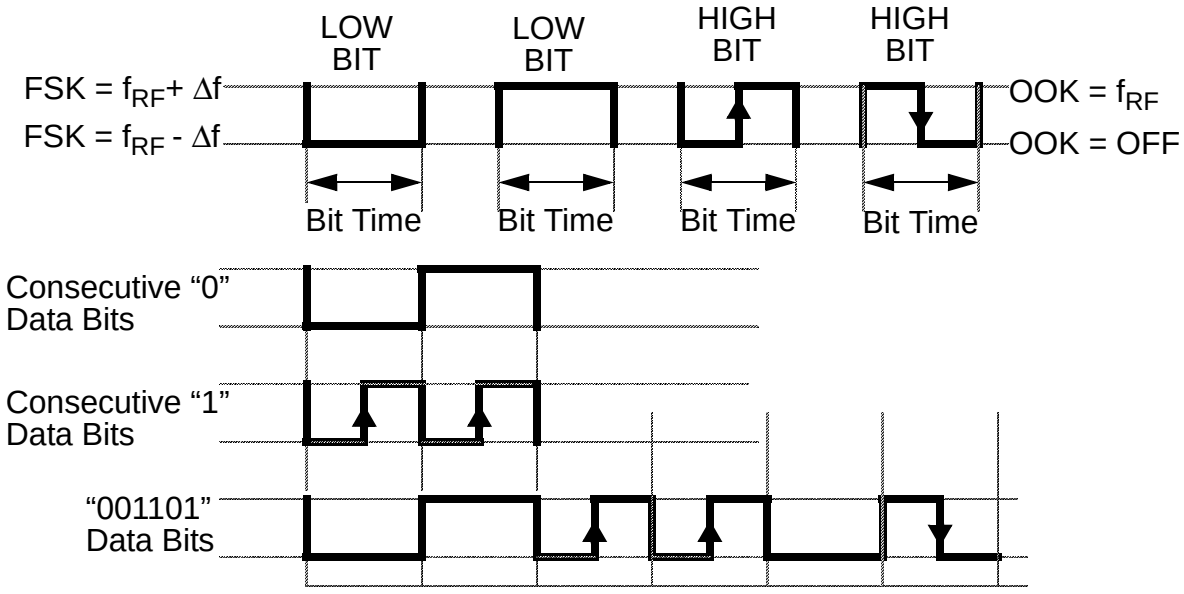


Figure 43. Bi-Phase data bit encoding (POL = 1)

13.6 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 13.1.

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.

13.6.1 Modulation method

The modulation control bit, MOD, described in Section 13.18**,** 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.

13.6.2 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.

13.6.3 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.

13.6.4 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.

13.6.5 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.

13.6.6 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.

13.7 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 the 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.

13.8 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.

13.9 RFM registers

The RFM contains twelve registers to control its functions and thirty-two registers to provide access to the output data buffer.

13.9.1 RFM Control Register 0 - RFCR0

The RFCR0 register contains eight control bits for setting the output data rate of the RFM as described in Table 122.

Table 122. RFM Control Register 0 (RFCR0)

$0030	Bit 7	6	5	4	3	2	1	Bit 0
R W	BPS[7:0]							
RFMRST:	0	0	1	1	0	1	0	0

Table 123. RFCR0 field descriptions

Field	Description
7–0 BPS[7:0]	**Data Rate -** The BPS[7:0] control bits select the data rate for the transmitted datagrams as described by the following equation: $$f_{DATA} = \frac{f_{XTAL}}{52 \times (BPS + 1)} = \frac{5 \times (10)^5}{(BPS + 1)}$$ where: f_{DATA} = Data rate in bits/second f_{XTAL} = External crystal frequency in Hz = 26 MHz BPS = Value of data rate code (BPS[7:0]) Examples of the value for common data rates are given in Table 124. The BPS[7:0] control bits are set to $34 by the RFMRST signal which results in a default data rate of 9600 bits/sec.

Table 124. 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 bps	2000.0	249	4800 bps	4807.7	103
2400 bps	2403.8	207	5000 bps	5000.0	99
4000 bps	4000.0	124	9600 bps	9615.4	51
4500 bps	4504.5	110	19200 bps	19230.8	25

The BPS[7:0] bits are set to $34 by an RFMRST signal which results in a default data rate of approximately 9600 bps.

13.10 RFM Control Register 1 - RFCR1

The RFCR1 register contains eight control bits for the RFM as described in Table 125.

Table 125. RFM Control Register 1 (RFCR1)

$0031	Bit 7	6	5	4	3	2	1	Bit 0
R W	FRM[7:0]							
RFMRST:	0	0	0	0	0	0	0	0

Table 126. RFCR1 Field Descriptions

Field	Description
7-0 FRM[7:0]	**Frame Bit Length** - The FRM[7:0] control bits 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.

13.11 RFM Control Register 2 - RFCR2

The RFCR2 register contains eight control bits for the RFM as described in Figure 127.

Table 127. RFM Control Register 2 (RFCR2)

$0032	Bit 7	6	5	4	3	2	1	Bit 0
R W	SEND	RPAGE	EOM	PWR[4:0]				
RFMRST:	0	0	0	0	0	0	0	0

Table 128. RFCR2 field descriptions

Field	Description
7 SEND	**Transmission Start 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. 1 Start data transmission or transmission in progress.
6 RPAGE	**Buffer Page Select** — The RPAGE bit will select the lower or upper 16 bytes of the RFM data buffer when writing/reading to the RFD0-RD15 registers. This bit also selects between the lower and upper banks of RFM registers at addresses $0038 through $003B. This bit is cleared by a reset of the MCU. 0 Select the lower 16 bytes of the RFM data buffer. 1 Select the upper 16 bytes of the RFM data buffer.
5 EOM	**End Of Message** — 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 a RFMRST. 0 EOM bit times not added. 1 EOM bit times added.

Table 128. RFCR2 field descriptions (continued)

Field	Description
4:0 PWR[4:0]	**RF Amplifier Power Level** — 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 a RFM reset. The PWR control bits are initially set to 0x00. This setting targets -10 dBm typical power output. The PWR control bits scale the typical output power level from -1.5 to 8 dBm in steps of 0.5 dB and fixes the low power level mode to -10 dBm, The power control range is defined as follows: 00000 set output power level to -10 dBm (Default Value) 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.

13.11.1 Power working domains

The working areas of the RF transmitter are divided into several domains as defined in Figure 44.

13.11.1.1 P_{TYP}

T_A = 25°C to 60°C and V_{DD} = 2.5 V to 3.6 V

P_{TYP} is where the power step is adjusted to guarantee 5 dBm. The power consumption in this domain is specified at 5 dBm output power step at nominal conditions of T_A = 25 °C and V_{DD} = 3 VDC.

13.11.1.2 P_{MIN}

P_{MIN} is where the power step is adjusted to guarantee a minimum of 3 dBm as shown in Figure 44. The power consumption in this domain is given as the maximum consumption at whatever temperature of supply voltage condition. The P_{MIN} domain is subdivided into two areas according to the lowest supply voltage encountered (1.8 or 2.5 VDC).

P_{MIN_COLD}

T_A = -40°C to 0°C and V_{DD} = 1.8 V to 3.6 V

P_{MIN_HOT}

T_A = 0°C to 25°C and V_{DD} = 2.5 V to 3.6 V

T_A = 60°C to 125°C and V_{DD} = 2.5 V to 3.6 V

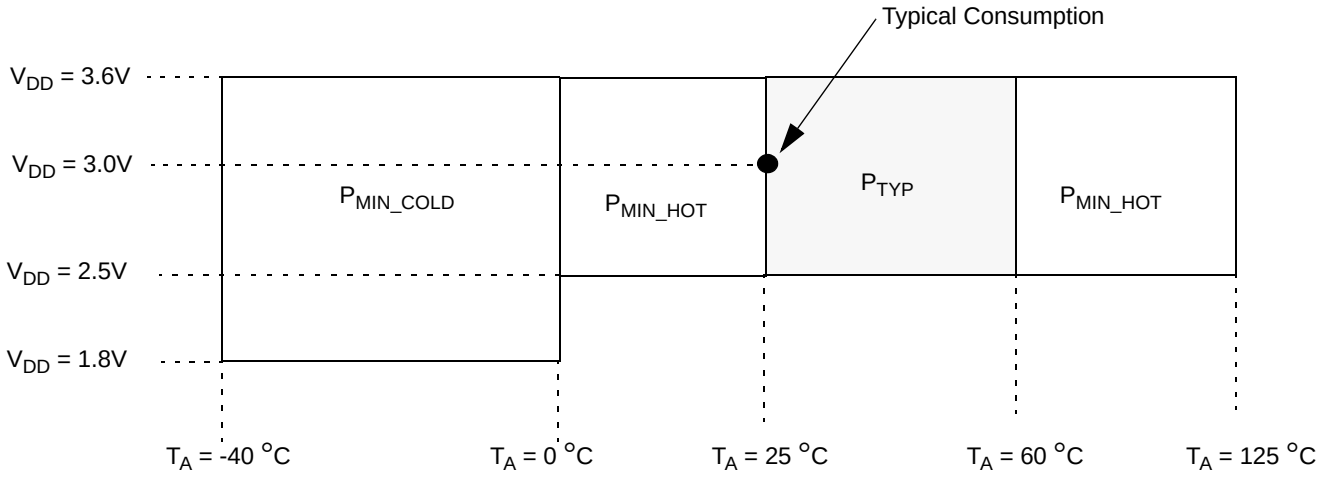


Figure 44. RF power domains

13.12 RFM Control Register 3 - RFCR3

The RFCR3 register contains five control bits for the RFM as described in Table 129 which sets the number of frames in each RF datagram.

Table 129. RFM Control Register 3 (RFCR3)

$0033	Bit 7	6	5	4	3	2	1	Bit 0
R W	DATA	IFPD	ISPC	IFID	FNUM[3:0			
RFMRST	0	0	0	0	0	0	0	0

Table 130. RFCR3 field descriptions

Field	Description
7 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. 1 RF output state high.
6 IFPD	**Interframe Power Down** — The IFPD control bit selects whether the XCO, VCO and PLL are powered down during interframe timing caused by the RFM. The IFPD control bit is cleared by the RFMRST signal. 0 The XCO, VCO and PLL remained powered up as long as the SEND bit is set. 1 The XCO, VCO and PLL are powered down during RFM controlled interframe timing events. The restart of these functions will start 1 msec before the end of the timing interval if another frame is to be transmitted.
5 ISPC	**Initial Random Space**— 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. 1 Initial time interval enabled.
4 IFID	**Interframe Interrupt Delay** — The IFID control bit selects whether the RFIF bit is set and the MCU is interrupted at the end of each frame sent or at the end of the last frame in a multiple frame message. 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. 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]** — The FNUM[3:0] bits set the number of frames transmitted in each RF datagram. The frames will be randomly spaced apart as described it Section 13.3.These bits are cleared by an RFM reset. The number of frame transmitted is the binary number plus one.

13.13 RFM Control Register 4 - RFCR4

The RFCR4 register contains eight control bits to set the initial and interframe timing base timing variable as described in Table 131. A RFMRST signal clears the RFBT[7:0] bits.

Table 131. RFCR4 register - base time variable

$0034	Bit 7	6	5	4	3	2	1	Bit 0
R W	RFBT[7:0]							
RFMRST:	0	0	0	0	0	0	0	0

Table 132. RFCR4 field descriptions

Field	Description
7:0 RFBT[7:0]	**Base Timer** — The RFBT[7:0] control bits 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.

13.14 RFM Control Register 5 - RFCR5

The RFCR5 register contains eight control bits to set the initial and interframe random timing variable as described in Table 133. A RFMRST signal clears the LFSR[6:0] bits causing the random time variable to be ignored.

Table 133. RFCR5 register - pseudo-random time variable

$0035	Bit 7	6	5	4	3	2	1	Bit 0
R W	BOOST	LFSR[6:0]						
RFMRST:	0	0	0	0	0	0	0	0

Table 134. RFCR5 field descriptions

Field	Description
7 BOOST	**BOOST** — 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). 1 The VCO runs at its higher power consumption level (lower phase noise).
6:0 LFSR[6:0]	**Pseudo-Random Timer**— The LFSR[6:0] bits 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.

13.15 RFM Control Register 6 - RFCR6

The RFCR6 register contains eight control bits to set the initial and interframe frame number timing variable as described in Table 135. A RFMRST signal clears the RFFT[5:0] bits.

Table 135. RFCR6 register - frame number time - RFTS[1:0] = 1:0

$0036	Bit 7	6	5	4	3	2	1	Bit 0
R W	VCO_GAIN[1:0]		RFFT[5:0}					
RFMRST:	1	0	0	0	0	0	0	0

Table 136. RFCR6 field descriptions

Field	Description
7:6 VCO_GAIN [1:0]	**VCO Gain Selection** — These bits control the VCO gain. The VCO_GAIN[1] bit is set and the VCO_GAIN[0] bit is cleared by the RFMRST signal. Not normally need to be adjusted by the end user.
5:0 RFFT[5:0]	**Frame Number Timer** — The RFFT[5:0] control bits 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.

13.16 RFM Control Register 7 - RFCR7

The RFCR7 register contains four control bits and four status bits for the RFM as described in Table 137.

Table 137. RFM transmit control registers (RFCR7)

$0037	Bit 7	6	5	4	3	2	1	Bit 0
R	RFIF	RFEF	RFVF	0	RFIEN	RFLVDEN	RCTS	0
W	—	—	—	RFIAK				RFMRST
RFMRST:	0	0	0	0	0	0	0	0

Table 138. RFCR7 field descriptions

Field	Description
7 RFIF	**RF Interrupt Flag** — The read-only RFIF status bit indicates if the RF transmission has ended 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. 1 RF transmission completed in the data buffer mode.
6 RFEF	**RF Transmission Error Flag** — The read-only RFEF status bit indicates if there was an error in the current or prior RF transmission as described in Section 13.6.4. 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. 1 RF transmission error occurred.
5 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. This bit is cleared by the RFMRST signal. 0 Voltage is and has been above RF LVD rising threshold or the RF LVD is disabled. 1 Voltage has dropped below the RF LVD falling threshold since last reset of this bit.
4 RFIAK	**Acknowledge RF Interrupt Flags** — 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. 1 Clear the RFIF and RFEF bits.
3 RFIEN	**RF Interrupt Enable** — The RFIEN bit enables the RFIF and RFEF bits to generate an interrupt to the MCU. The RFMRST signal clears this bit. 0 RF interrupts disabled. 1 RF interrupts enabled.
2 RFLVDEN	**RF LVD Enable** — When the RFLVDEN bit is set the RF LVD circuit will be enabled. If the MCU is to be stopped the Stop4 mode should be used to allow the RF LVD to function properly. This bit is cleared by the RFMRST signal. 0 RF LVD disabled. 1 RF LVD enabled.
1 RCTS	**RF Clear To Send Status**— 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. 1 RFM ready to send.
0 RFMRST	**RFM Reset** — 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. 1 Reset RFM.

13.17 PLL Control Registers A- PLLCR[1:0], RPAGE = 0

The PLLCR[1:0] registers contain 16 control bits for the RFM as described in Figure 139. These bits are only accessible when the RPAGE bit is cleared.

Table 139. PLL Control Registers A (PLLCR[1:0], RPAGE = 0)

$0038	Bit 7	6	5	4	3	2	1	Bit 0
R W	AFREQ[12:5]							
RFMRST:	0	0	0	0	0	0	0	0
$0039								
R W	AFREQ[4:0]					POL	CODE	
RFMRST:	0	0	0	0	0	0	0	0

Table 140. PLLCR[1:0] Field Descriptions

Field	Description
PLLCR0 7:0 AFREQ 12:5 PLLCR1 7:3 AFREQ 4:0	**PLL Divider Ratio A** — The AFREQ[12:0] control bits select the PLL divider ratio for a data zero in the FSK mode of modulation as described by the following equation: $$f_{DATA0} = f_{XTAL} \times \left((12 + 4 \times CF) + \frac{AFREQ}{8192}\right)$$ where: f_{DATA0} = RF Carrier frequency for a data zero in MHz f_{XTAL} = External crystal frequency in MHz, 26 Mhz CF = State of the CF carrier select bit AFREQ = Decimal value of the AFREQ[12:0] binary weighted bits The AFREQ[12:0] control bits are cleared by the RFMRST signal. 1 LSB of AFREQ[12:0] = 3.17 kHz.
2 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 as in Figure 41 and Bi-Phase encoding polarity as in Figure 43. 1 NRZ and MCU direct DATA bit data inverted and Manchester encoding polarity as in Figure 40 and Bi-Phase encoding polarity as in Figure 42.
1:0 CODE [1:0]	**Data Encoding and Source** — The 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. 00 Manchester encoded data from the RFM data buffer. 01 Bi-Phase encoded data from the RFM data buffer. 10 NRZ direct data from the RFM data buffer (can be mixed NRZ and Manchester at 2X the data rate). 11 MCU direct mode with RF output driven by the state of the DATA bit.

13.18 PLL Control Registers B- PLLCR[3:2], RPAGE = 0

The PLLCR[3:2] registers contain 16 control bits for the RFM as described in Table 141. These bits are only accessible when the RPAGE bit is cleared.

Table 141. PLL Control Registers B (PLLCR[3:2], RPAGE = 0)

$003A	Bit 7	6	5	4	3	2	1	Bit 0
R W	BFREQ[12:5]							
RFMRST:	0	0	0	0	0	0	0	0
$003B								
R W	BFREQ[4:0]					CF	MOD	CKREF
RFMRST:	0	0	0	0	0	0	0	0

Table 142. PLLCR[3:2] Field Descriptions

Field	Description
PLLCR2 7:0 BFREQ 12:5 PLLCR3 7:3 BFREQ 4:0	**PLL Divider Ratio B** — The BFREQ[12:0] control bits select the PLL divider ratio for a data one in either the OOK or FSK modes of modulation as described by the following equation: $$f_{DATA1} = f_{XTAL} \times \left((12 + 4 \times CF) + \frac{BFREQ}{8192} \right)$$ where: $f_{CARRIER}$ = RF Carrier frequency in MHz f_{XTAL} = External crystal frequency in MHz CF = State of the CF carrier select bit BFREQ = Decimal value of the BFREQ[12:0] binary weighted bits The BFREQ[12:0] control bits are cleared by the RFMRST signal. 1 LSB of BFREQ[12:0] = 3.17 kHz.
2 CF	**Carrier Frequency** — The CF control bit selects the optimal VCO setup and correct divider for the 500 kHz reference clock to the MCU on D_X 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. 1 Configured for 434 MHz, 16.6923 PLL divider using a 26.000 MHz external crystal.
1 MOD	**RF Modulation Method** — 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. 1 Configured for FSK.
0 CKREF	**Generated Clock Reference** — Generates the DX signal to the TPM1 module for determining the other clock frequencies: 0 D_X signal not generated. 1 D_X 500 kHz signal connected to the TPM1 module.

13.19 EPR register - EPR (RPAGE = 1)

The EPR register contains eight control bits for the RFM as described in Table 143. The function of the upper 4 bits depends on the state of the VCD_EN bit.

Table 143. RFM EPR registers (EPR, RPAGE = 1, VCD_EN = 0)

$0038	Bit 7	6	5	4	3	2	1	Bit 0
R W	—	PLL_LPF_[2:0]			—	—	PA_SLOPE	VCD_EN
RFMRST:	0	0	1	1	0	0	1	0

Table 144. RFM EPR registers (EPR, RPAGE = 1, VCD_EN = 1)

$0038	Bit 7	6	5	4	3	2	1	Bit 0
R	VCD[3:0]				—	—	PA_SLOPE	VCD_EN
W								
RFMRST:	—	—	—	—	0	0	1	0

Table 145. EPR field descriptions

Field	Description
7 Reserved	**Reserved bit** — Not for user access if the VCD_EN bit is clear.
6-4 PLL_LPF_ [2:0]	**Low-Pass Filter Selection** — These read/write bits select the PLL low-pass filter. A reset sets these bits to $03. These bits are only accessible if the VCD_EN bit is clear.
7-4 VCD[3:0]	**VCO Calibration Count Difference** — These read-only bits show the count difference from “ideal” when the VCO calibration machine is finished (see Section 13.21). These bits are only accessible when the VCD_EN bit is set. Writing to these bits when the VCD_EN bit is set has no effect. The reset state is undefined.
3-2 Reserved	**Reserved bits** — Not for user access.
1 PA_SLOPE	**PA Output Slope Selection** — This read/write bit controls the output slope of the RFM PA output. This bit is set by the RFMRST signal.
0 VCD_EN	**VCD Enable bit** — This bit allows access to the VCD[3:0] bits. This bit is cleared by the RFMRST signal. 0 PLL_LPF_[2:0] bits accessed. 1 VCD[3:0] bits accessed.

13.20 RF data registers - RFD[31:0]

The RFD registers contain 256 read/write bits for the RFM to use when outputting data as described in Section 13.2. The 256-bit buffer is divided into two pages of 128 bits as selected by the RPAGE bit in the RFCR2. These as described in Table 146. These bits are unaffected by any reset.

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 (RFD255) in the most significant byte (RFB31). This is often referred to as “little-endian” data ordering. The output of this data by the RFM in all 256 bits locations is not dependent on the state of the RPAGE bit.

Table 146. RF data registers (RFD[31:0])

	Bit 7	6	5	4	3	2	1	Bit 0
$003C	RFD[7:0] for RPAGE=0, RFD[135:128] for RPAGE=1							
$003D	RFD[15:8] for RPAGE=0, RFD[143:136] for RPAGE=1							
$003E	RFD[23:16] for RPAGE=0, RFD[151:144] for RPAGE=1							
$003F	RFD[31:24] for RPAGE=0, RFD[159:152] for RPAGE=1							
$0040	RFD[39:32] for RPAGE=0, RFD[167:160] for RPAGE=1							
$0041	RFD[47:40] for RPAGE=0, RFD[175:168] for RPAGE=1							
$0042	RFD[55:48] for RPAGE=0, RFD[183:176] for RPAGE=1							
$0043	RFD[63:56] for RPAGE=0, RFD[191:184] for RPAGE=1							
$0044	RFD[71:64] for RPAGE=0, RFD[199:192] for RPAGE=1							
$0045	RFD[79:72] for RPAGE=0, RFD[207:200] for RPAGE=1							
$0046	RFD[87:80] for RPAGE=0, RFD[215:208] for RPAGE=1							
$0047	RFD[95:88] for RPAGE=0, RFD[223:216] for RPAGE=1							
$0048	RFD[103:96] for RPAGE=0, RFD[231:224] for RPAGE=1							
$0049	RFD[111:104] for RPAGE=0, RFD[239:232] for RPAGE=1							
$004A	RFD[119:112] for RPAGE=0, RFD[247:240] for RPAGE=1							
$004B	RFD[127:120] for RPAGE=0, RFD[255:248] for RPAGE=1							

r

Table 147. RFD[31:0] field descriptions

Field	Description
RFD 15:0 RPAGE=0 RFD [127:0]	**RF Data Registers Lower 128 bits** — These are read/write bits that hold the lower 128 bits of possible data to be sent by the RFM. Access to the lower 128 bits occurs when the RPAGE bit is clear. These bits are unaffected by any reset.
RFD 31:16 RPAGE=1 RFD [255:128]	**RF Data Registers Upper 128 bit** — These are read/write bits that hold the upper 128 bits of possible data to be sent by the RFM. Access to the lower 128 bits occurs when the RPAGE bit is clear. These bits are unaffected by any reset.

13.21 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 maxband VCO sub-bands. maxband 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 45.

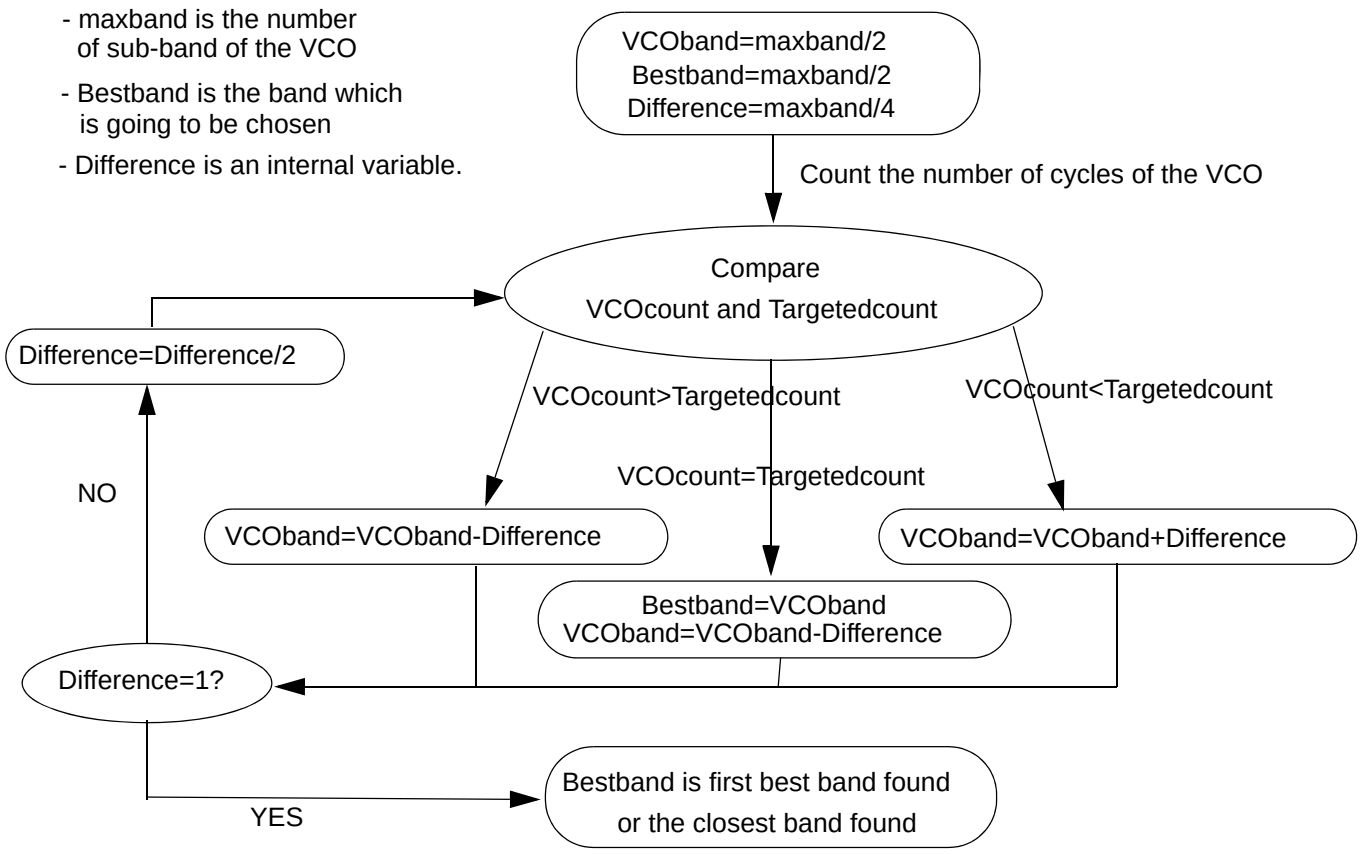


Figure 45. VCO calibration state machine

14 Firmware

This section describes the software subroutines contained in the firmware section of the FLASH memory that the user can call for various tasks and to reduce the software development time for the main internal operations.

14.1 Software jump table

All subroutines are accessed through a jump table located at the bottom of the firmware FLASH memory as described in Table 148. This allows upgrades in firmware without changing the software code of the user. All subroutines should be accessed with the JSR instruction.

14.2 Function documentation

The following subsections describe the details of the firmware routines. Further details can be found in the latest version of the MPXx85/86xxD Embedded Firmware User Guide.

14.2.1 General rules

1. No output parameter can use the extreme codes (all zero's or all one's).
2. The all zero's output code will always indicate a fault and the status byte will indicate the source of the error.
3. While firmware is processing, CPU resources are unavailable for application.
4. Each measured parameter will return a limit code ($00, $FF or $1FF) if an error occurs in its acquisition, except for the external ADC voltage measurements on the PTA[1:0] pins.
5. External ADC voltage measurements on the PTA[1:0] pins will return a full range code that is ratiometric to the supply voltage.

14.2.2 Firmware routines

The details on the use and execution of each firmware routine is documented in the Code Warrior project file that is supplied by Freescale. Any future updates to these firmware routines will be contained in that file. A summary of the firmware routines available is given in Table 148.

The firmware table is comprised of three-byte entries where the first byte is the operational code for the JMP instruction, and the following two bytes are the absolute address pointing to the location of the firmware function.

Table 148. Firmware summary and jump table

Address	Routine	Description
E000	TPMS_RESET	Master reset of complete device
E003	TPMS_READ_VOLTAGE	10-bit uncompensated bandgap voltage reading
E006	TPMS_COMP_VOLTAGE	8-bit compensation of 10-bit voltage reading
E009	TPMS_READ_TEMPERATURE	10-bit uncompensated temperature reading
E00C	TPMS_COMP_TEMPERATURE	8-bit compensation of 10-bit temperature reading
E00F	TPMS_READ_PRESSURE	10-bit uncompensated pressure reading
E012	TPMS_COMP_PRESSURE	9-bit compensation of 10-bit pressure reading
E015	TPMS_READ_ACCELERATION_X	10-bit uncompensated X-axis accel reading
E018	TPMS_READ_DYNAMIC_ACCEL_X	10-bit uncompensated X-axis accel reading with dynamic offset adjustment.
E01B	TPMS_COMP_ACCELERATION_X	9-bit compensation of 10-bit X-axis accel reading
E01E	TPMS_READ_ACCELERATION_Z	10-bit uncompensated Z-axis accel reading
E021	TPMS_READ_DYNAMIC_ACCEL_Z	10-bit uncompensated Z-axis accel reading with dynamic offset adjustment.
E024	TPMS_COMP_ACCELERATION_Z	9-bit compensation of 10-bit Z-axis accel reading
E027	TPMS_READ_ACCELERATION_XZ	10-bit uncompensated X-axis and Z-axis accel readings
E02A	TPMS_READ_DYNAMIC_ACCEL_XZ	10-bit uncompensated X-axis and Z-axis accel readings with dynamic offset adjustment.
E02D	TPMS_COMP_ACCELERATION_XZ	9-bit compensation of 10-bit X-axis and Z-axis accel readings

Table 148. Firmware summary and jump table (continued)

Address	Routine	Description
E030	TPMS_READ_V0	10-bit uncompensated voltage reading on PTA0 pin
E033	TPMS_READ_V1	10-bit uncompensated voltage reading on PTA1 pin
E036	TPMS_LFOCAL	LFO clock calibration
E039	TPMS_MFOCAL	MFO clock calibration
E03C	TPMS_RF_ENABLE	Enable and set up RFM
E03F	TPMS_RF_RESET	Master reset of RFM
E042	TPMS_RF_READ_DATA	Read RFM data buffer
E045	TPMS_RF_READ_DATA_REVERSE	Read RFM data buffer in reverse bit order
E048	TPMS_RF_WRITE_DATA	Write RFM data buffer
E04B	TPMS_RF_WRITE_DATA_REVERSE	Write RFM data buffer in reverse bit order
E04E	TPMS_RF_CONFIG_DATA	Configure RFM
E051	Reserved	Reserved
E054	TPMS_RF_SET_TX	Initiate RF transmission
E057	TPMS_RF_DYNAMIC_POWER	Adjusts PA for uniform power output
E05A	TPMS_MSG_INIT	Initialization of the emulated serial communication
E05D	TPMS_MSG_READ	Reading data from emulated serial interface
E060	TPMS_MSG_WRITE	Writing data on emulated serial interface
E063	TPMS_CHECKSUM_XOR	Calculates a checksum for given buffer in XOR
E066	TPMS_CRC8	Calculates CRC8 on portion of memory
E069	TPMS_CRC16	Calculates CRC16 on portion of memory
E06C	TPMS_SQUARE_ROOT	Calculates square root
E06F	TPMS_READ_ID	Reads device ID stored in FLASH
E072	TPMS_LF_ENABLE	Enable/Disable LF for Carrier or Data
E075	TPMS_LF_READ_DATA	Reading LF data
E078	TPMS_WIRE_AND_ADC_CHECK	Performs checks of internal bond wires
E07B	TPMS_FLASH_WRITE	Write to FLASH
E07E	TPMS_FLASH_CHECK	Performs checksum on Freescale firmware FLASH
E081	TPMS_FLASH_ERASE	Erases one page (512 bytes) of FLASH at a time
E084	TPMS_FLASH_PROTECTION	Lock out FLASH
E087	Reserved	Reserved
E08A	TPMS_MULT_SIGN_INT16	Multiple two signed 16-bit numbers together
E08D	TPMS_WAVG	Weighted average
E090	TPMS_RDE_ADJUST_PRESSURE	Check to determine if pressure reading should be adjusted following a rapid decompression. Refer to Appendix C, “Rapid Decompression Events” on page 180.

14.2.3 Device identification

The bytes assigned to identify the device and its options are described below. These data can be read by use of the TPMS_READ_ID routine.

Table 149. Device ID coding summary

ID Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
00	**CODE0**	Reserved - Firmware Revision/Software Information							
01	**CODE1**	Reserved		ES0	PRESS	ACC1	ACC0	SPCLA	SPCLP
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

Table 150. Device ID coding descriptions

Field	Description
CODE0 7:0 Reserved	Reserved for Freescale firmware description.
CODE1 7 ES	Revision number for the multiple-chip-module silicon. 0 - 1-hole lid 1 - 5-hold lid
CODE1 6:5 Reserved	Reserved for Freescale firmware description.
CODE1 4 PRESS	Calibrated range for pressure. The range is a combination of this bit and the SPCLP bit, below. SPCLP = 0, PRESS = 0 indicates a 100-450 kPa range SPCLP = 0, PRESS = 1 indicates a 100-900 kPa range SPCLP = 1, PRESS = 0 indicates a 100-550 kPa range SPCLP = 1, PRESS = 1 is unused
CODE1 3:2 ACC	Type of accelerometer. 00 - None 01 - One accelerometer with X-axis orientation 10 - One accelerometer with Z-axis orientation 11 - Two accelerometers with X- and Z-axis orientations
CODE1 1 SPCLA	Special calibration for accelerometer. 0 - Normal trim 1 - Special trim
CODE1 0 SPCLP	Calibrated range for pressure. The range is a combination of this bit and the PRESS bit, above. SPCLP = 0, PRESS = 0 indicates a 100-450 kPa range SPCLP = 0, PRESS = 1 indicates a 100-900 kPa range SPCLP = 1, PRESS = 0 indicates a 100-550 kPa range SPCLP = 1, PRESS = 1 is unused
CODE2:4 7:0 CODE5 3:0 ID27:0	28-bit serial number for each device. All numbers to be unique with numbering sequence being a sequential counter for each product type.
CODE5 7:4 ID31:28	4-bit number assigned to vendor type. If these bits are unspecified as part of the complete FLASH programming by the customer, then these 4 bits are programmed to $8

14.2.4 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 46. 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 all cases the signal will be between the supply rails, so that the ADC10 will convert it to a range of digital numbers between 0 and 1023. These digital numbers will 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.

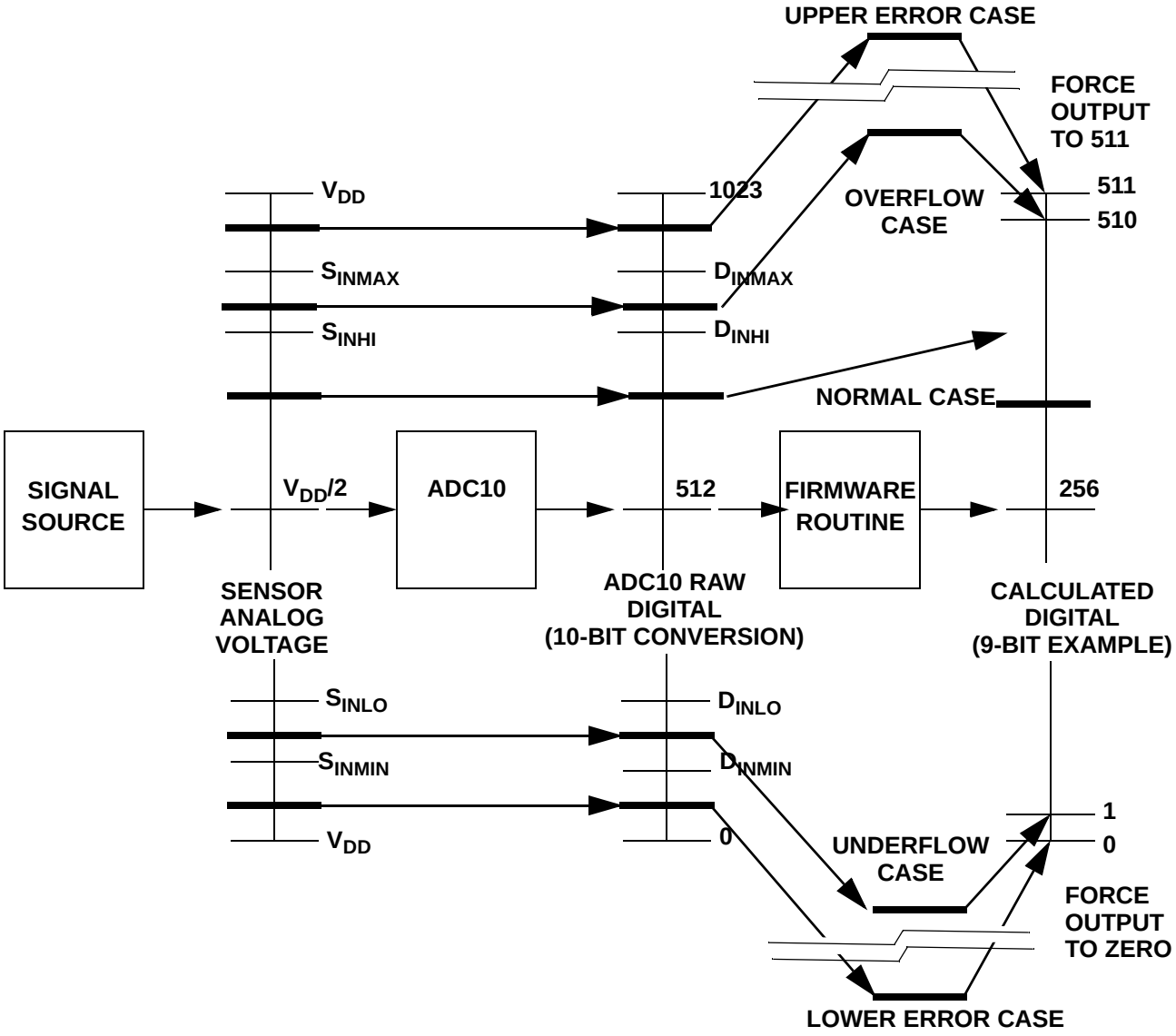


Figure 46. Measurement signal range definitions

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.

14.3 Memory resource usage

The firmware uses the top 8192 bytes of the FLASH memory map.

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

The firmware uses one byte ($008F) of the parameter registers for global flags for all routines.

Software stack

The RESET firmware function sets the SP register to the last address in the RAM. The user can change the default stack location to meet its own application needs.

15 Development Support

15.1 Introduction

This chapter describes the single-wire background debug mode (BDM), which uses the on-chip background debug controller (BDC) module, and the independent on-chip real-time in-circuit emulation (ICE) system, which uses the on-chip debug (DBG) module.

15.1.1 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

Features of the ICE system include:

- Two trigger comparators: Two address + read/write (R/W) or one full address + data + R/W
- Flexible 8-word by 16-bit FIFO (first-in, first-out) buffer for capture information:
  - Change-of-flow addresses or
  - Event-only data
- Two types of breakpoints:
  - Tag breakpoints for instruction opcodes
  - Force breakpoints for any address access
- Nine trigger modes:
  - Basic: A-only, A OR B
  - Sequence: A then B
  - Full: A AND B data, A AND NOT B data
  - Event (store data): Event-only B, A then event-only B
  - Range: Inside range ($A \leq address \leq B$), outside range (address < A or address > B)

15.2 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 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.

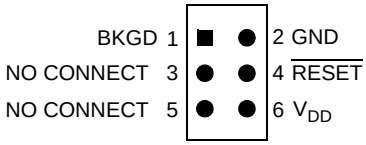


Figure 47. BDM tool connector

15.2.1 BKGD pin description

BKGD 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, refer to Section 15.2.2.

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 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. Refer to Section 15.2.2, for more detail.

When no debugger pod is connected to the 6-pin BDM interface connector, the internal pullup on BKGD chooses normal operating mode. When a debug pod is connected to BKGD 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 (refer to the introduction to this Development Support section). It is not necessary to reset the target MCU to communicate with it through the background debug interface.

15.2.2 Communication details

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

BKGD 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 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 48 shows an external host transmitting a logic 1 or 0 to the BKGD 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 pin. Typically, the host actively drives the pseudo-open-drain BKGD pin during host-to-target transmissions to speed up rising edges. Because the target does not drive the BKGD pin during the host-to-target transmission period, there is no need to treat the line as an open-drain signal during this period.

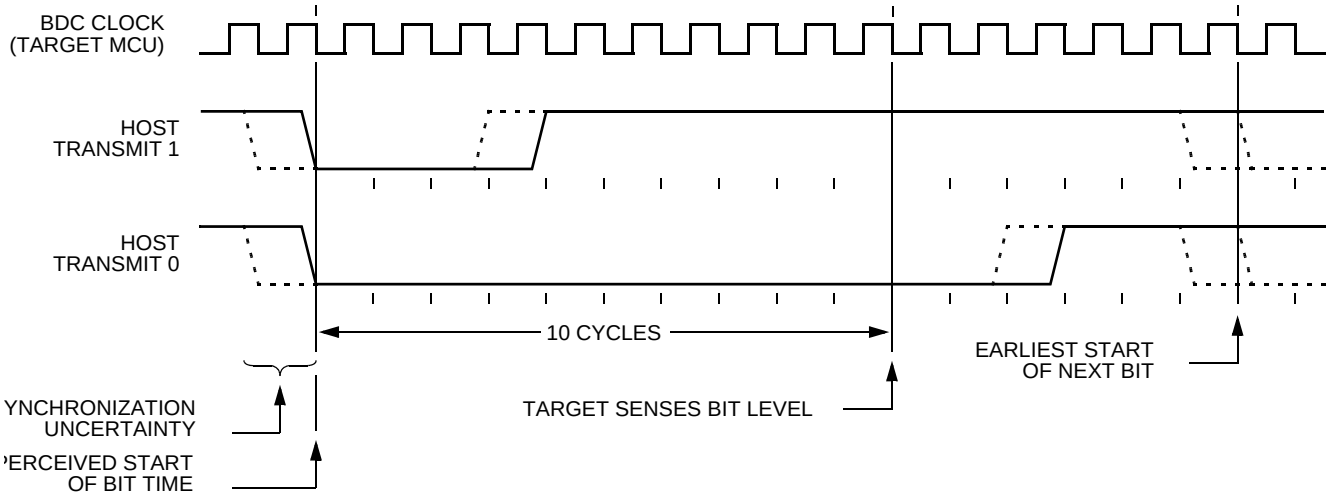


Figure 48. BDC host-to-target serial bit timing

Figure 49 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 to the perceived start of the bit time in the target MCU. The host holds the BKGD 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.

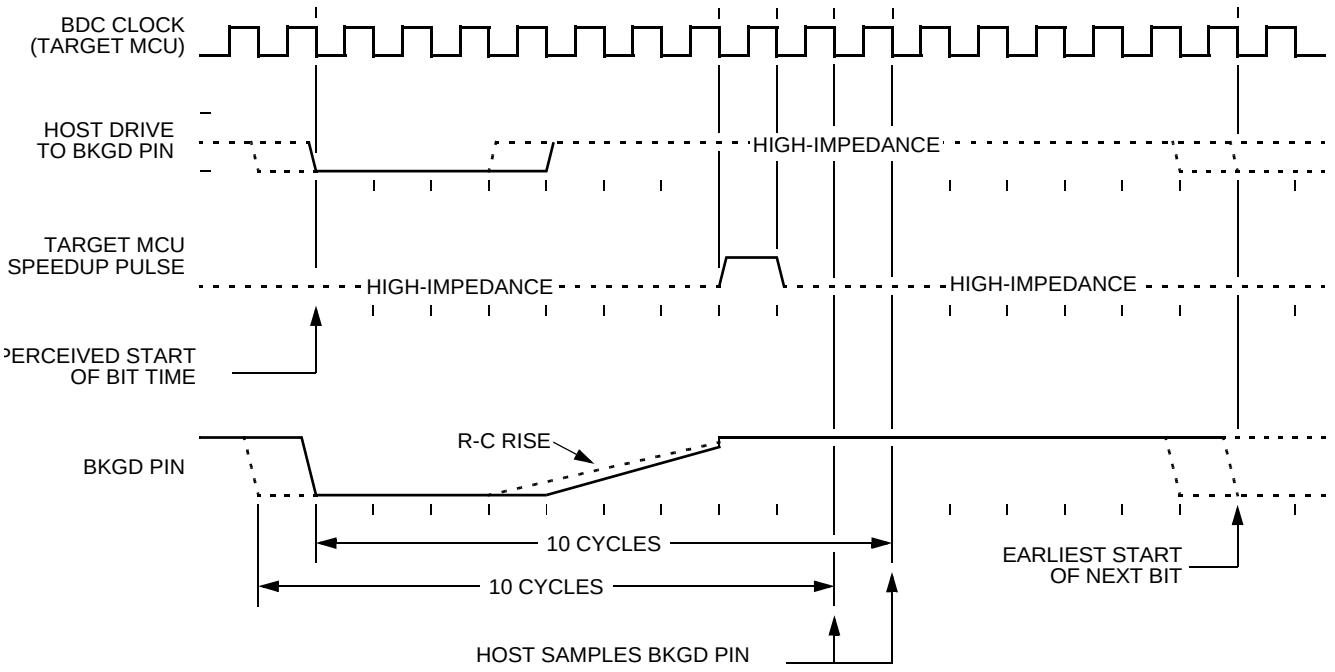


Figure 49. BDC target-to-host serial bit timing (Logic 1)

Figure 50 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 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 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.

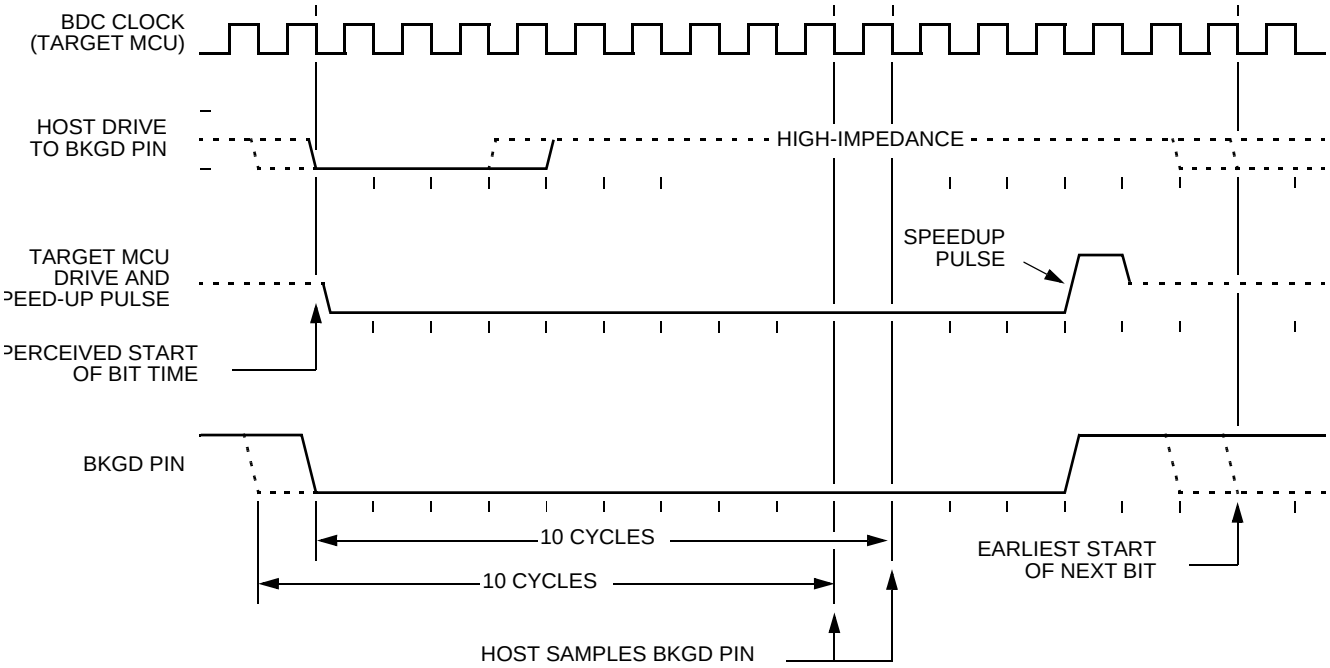


Figure 50. BDM target-to-host serial bit timing (Logic 0)

15.2.3 BDC commands

BDC commands are sent serially from a host computer to the BKGD 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 151 shows all HCS08 BDC commands, a shorthand description of their coding structure, and the meaning of each command.

Coding structure nomenclature

This nomenclature is used in Table 151 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
RD16	= 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 151. 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 Freescale document order no. HCS08RMv1/D.
ACK_DISABLE	Non-intrusive	D6/d	Disable acknowledge protocol. Refer to Freescale 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)
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 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 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 pin so it reverts to high impedance
- Monitors the BKGD 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 to return to a logic high
- Delays 16 cycles to allow the host to stop driving the high speedup pulse
- Drives BKGD low for 128 BDC clock cycles
- Drives a 1-cycle high speedup pulse to force a fast rise time on BKGD
- Removes all drive to the BKGD 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.

15.2.4 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 will enter 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.

The on-chip debug module (DBG) includes circuitry for two additional hardware breakpoints that are more flexible than the simple breakpoint in the BDC module.

15.3 On-chip debug system (DBG)

Because HCS08 devices do not have external address and data buses, the most important functions of an in-circuit emulator have been built onto the chip with the MCU. The debug system consists of an 8-stage FIFO that can store address or data bus information, and a flexible trigger system to decide when to capture bus information and what information to capture. The system relies on the single-wire background debug system to access debug control registers and to read results out of the eight stage FIFO.

The debug module includes control and status registers that are accessible in the user's memory map. These registers are located in the high register space to avoid using valuable direct page memory space.

Most of the debug module's functions are used during development, and user programs rarely access any of the control and status registers for the debug module. The one exception is that the debug system can provide the means to implement a form of ROM patching. This topic is discussed in greater detail in Section 15.3.6.

15.3.1 Comparators A and B

Two 16-bit comparators (A and B) can optionally be qualified with the R/W signal and an opcode tracking circuit. Separate control bits allow you to ignore R/W for each comparator. The opcode tracking circuitry optionally allows you to specify that a trigger will occur only if the opcode at the specified address is actually executed as opposed to only being read from memory into the instruction queue. The comparators are also capable of magnitude comparisons to support the inside range and outside range trigger modes. Comparators are disabled temporarily during all BDC accesses.

The A comparator is always associated with the 16-bit CPU address. The B comparator compares to the CPU address or the 8-bit CPU data bus, depending on the trigger mode selected. Because the CPU data bus is separated into a read data bus and a write data bus, the RWAEN and RWA control bits have an additional purpose, in full address plus data comparisons they are used to decide which of these buses to use in the comparator B data bus comparisons. If RWAEN = 1 (enabled) and RWA = 0 (write), the CPU's write data bus is used. Otherwise, the CPU's read data bus is used.

The currently selected trigger mode determines what the debugger logic does when a comparator detects a qualified match condition. A match can cause:

- Generation of a breakpoint to the CPU
- Storage of data bus values into the FIFO
- Starting to store change-of-flow addresses into the FIFO (begin type trace)
- Stopping the storage of change-of-flow addresses into the FIFO (end type trace)

15.3.2 Bus capture information and FIFO operation

The usual way to use the FIFO is to setup the trigger mode and other control options, then arm the debugger. When the FIFO has filled or the debugger has stopped storing data into the FIFO, you would read the information out of it in the order it was stored into the FIFO. Status bits indicate the number of words of valid information that are in the FIFO as data is stored into it. If a trace run is manually halted by writing 0 to ARM before the FIFO is full (CNT = 1:0:0:0), the information is shifted by one position and the host must perform (8 – CNT) – 1) dummy reads of the FIFO to advance it to the first significant entry in the FIFO.

In most trigger modes, the information stored in the FIFO consists of 16-bit change-of-flow addresses. In these cases, read DBGFH then DBGFL to get one coherent word of information out of the FIFO. Reading DBGFL (the low-order byte of the FIFO data port) causes the FIFO to shift so the next word of information is available at the FIFO data port. In the event-only trigger modes (see Section 15.3.5), 8-bit data information is stored into the FIFO. In these cases, the high-order half of the FIFO (DBGFH) is not used and data is read out of the FIFO by simply reading DBGFL. Each time DBGFL is read, the FIFO is shifted so the next data value is available through the FIFO data port at DBGFL.

In trigger modes where the FIFO is storing change-of-flow addresses, there is a delay between CPU addresses and the input side of the FIFO. Because of this delay, if the trigger event itself is a change-of-flow address or a change-of-flow address appears during the next two bus cycles after a trigger event starts the FIFO, it will not be saved into the FIFO. In the case of an end-trace, if the trigger event is a change-of-flow, it will be saved as the last change-of-flow entry for that debug run.

The FIFO can also be used to generate a profile of executed instruction addresses when the debugger is not armed. When ARM = 0, reading DBGFL causes the address of the most-recently fetched opcode to be saved in the FIFO. To use the profiling feature, a host debugger would read addresses out of the FIFO by reading DBGFH then DBGFL at regular periodic intervals. The first eight values would be discarded because they correspond to the eight DBGFL reads needed to initially fill the FIFO. Additional periodic reads of DBGFH and DBGFL return delayed information about executed instructions so the host debugger can develop a profile of executed instruction addresses.

15.3.3 Change-of-flow information

To minimize the amount of information stored in the FIFO, only information related to instructions that cause a change to the normal sequential execution of instructions is stored. With knowledge of the source and object code program stored in the target system, an external debugger system can reconstruct the path of execution through many instructions from the change-of-flow information stored in the FIFO.

For conditional branch instructions where the branch is taken (branch condition was true), the source address is stored (the address of the conditional branch opcode). Because BRA and BRN instructions are not conditional, these events do not cause change-of-flow information to be stored in the FIFO.

Indirect JMP and JSR instructions use the current contents of the H:X index register pair to determine the destination address, so the debug system stores the run-time destination address for any indirect JMP or JSR. For interrupts, RTI, or RTS, the destination address is stored in the FIFO as change-of-flow information.

15.3.4 Tag vs. force breakpoints and triggers

Tagging is a term that refers to identifying an instruction opcode as it is fetched into the instruction queue, but not taking any other action until and unless that instruction is actually executed by the CPU. This distinction is important because any change-of-flow from a jump, branch, subroutine call, or interrupt causes some instructions that have been fetched into the instruction queue to be thrown away without being executed.

A force-type breakpoint waits for the current instruction to finish and then acts upon the breakpoint request. The usual action in response to a breakpoint is to go to Active Background mode rather than continuing to the next instruction in the user application program.

The tag vs. force terminology is used in two contexts within the debug module. The first context refers to breakpoint requests from the debug module to the CPU. The second refers to match signals from the comparators to the debugger control logic. When a tag-type break request is sent to the CPU, a signal is entered into the instruction queue along with the opcode so that if/when this opcode ever executes, the CPU will effectively replace the tagged opcode with a BGND opcode so the CPU goes to Active Background mode rather than executing the tagged instruction. When the TRGSEL control bit in the DBGT register is set to select tag-type operation, the output from comparator A or B is qualified by a block of logic in the debug module that tracks opcodes and only produces a trigger to the debugger if the opcode at the compare address is actually executed. There is separate opcode tracking logic for each comparator so more than one compare event can be tracked through the instruction queue at a time.

15.3.5 Trigger modes

The trigger mode controls the overall behavior of a debug run. The 4-bit TRG field in the DBGT register selects one of nine trigger modes. When TRGSEL = 1 in the DBGT register, the output of the comparator must propagate through an opcode tracking circuit before triggering FIFO actions. The BEGIN bit in DBGT chooses whether the FIFO begins storing data when the qualified trigger is detected (begin trace), or the FIFO stores data in a circular fashion from the time it is armed until the qualified trigger is detected (end trigger).

A debug run is started by writing a 1 to the ARM bit in the DBGC register, which sets the ARMF flag and clears the AF and BF flags and the CNT bits in DBGS. A begin-trace debug run ends when the FIFO gets full. An end-trace run ends when the selected trigger event occurs. Any debug run can be stopped manually by writing a 0 to ARM or DBGEN in DBGC.

In all trigger modes except event-only modes, the FIFO stores change-of-flow addresses. In event-only trigger modes, the FIFO stores data in the low-order eight bits of the FIFO.

The BEGIN control bit is ignored in event-only trigger modes and all such debug runs are begin type traces. When TRGSEL = 1 to select opcode fetch triggers, it is not necessary to use R/W in comparisons because opcode tags would only apply to opcode fetches that are always read cycles. It would also be unusual to specify TRGSEL = 1 while using a full mode trigger because the opcode value is normally known at a particular address.

The following trigger mode descriptions only state the primary comparator conditions that lead to a trigger. Either comparator can usually be further qualified with R/W by setting RWAEN (RWBEN) and the corresponding RWA (RWB) value to be matched against R/W. The signal from the comparator with optional R/W qualification is used to request a CPU breakpoint if BRKEN = 1 and TAG determines whether the CPU request will be a tag request or a force request.

- **A-Only** — Trigger when the address matches the value in comparator A
- **A OR B** — Trigger when the address matches either the value in comparator A or the value in comparator B
- **A Then B** — Trigger when the address matches the value in comparator B but only after the address for another cycle matched the value in comparator A. There can be any number of cycles after the A match and before the B match.
- **A AND B Data (Full Mode)** — This is called a full mode because address, data, and R/W (optionally) must match within the same bus cycle to cause a trigger event. Comparator A checks address, the low byte of comparator B checks data, and R/W is checked against RWA if RWAEN = 1. The high-order half of comparator B is not used.
  In full trigger modes it is not useful to specify a tag-type CPU breakpoint (BRKEN = TAG = 1), but if you do, the comparator B data match is ignored for the purpose of issuing the tag request to the CPU and the CPU breakpoint is issued when the comparator A address matches.
- **A AND NOT B Data (Full Mode)** — Address must match comparator A, data must not match the low half of comparator B, and R/W must match RWA if RWAEN = 1. All three conditions must be met within the same bus cycle to cause a trigger.

  In full trigger modes it is not useful to specify a tag-type CPU breakpoint (BRKEN = TAG = 1), but if you do, the comparator B data match is ignored for the purpose of issuing the tag request to the CPU and the CPU breakpoint is issued when the comparator A address matches.
- **Event-Only B (Store Data)** — Trigger events occur each time the address matches the value in comparator B. Trigger events cause the data to be captured into the FIFO. The debug run ends when the FIFO becomes full.
- **A Then Event-Only B (Store Data)** — After the address has matched the value in comparator A, a trigger event occurs each time the address matches the value in comparator B. Trigger events cause the data to be captured into the FIFO. The debug run ends when the FIFO becomes full.
- **Inside Range (A ≤ Address ≤ B)** — A trigger occurs when the address is greater than or equal to the value in comparator A and less than or equal to the value in comparator B at the same time.
- **Outside Range (Address < A or Address > B)** — A trigger occurs when the address is either less than the value in comparator A or greater than the value in comparator B.

15.3.6 Hardware breakpoints

The BRKEN control bit in the DBGC register may be set to 1 to allow any of the trigger conditions described in Section 15.3.5, to be used to generate a hardware breakpoint request to the CPU. TAG in DBGC controls whether the breakpoint request will be treated as a tag-type breakpoint or a force-type breakpoint. A tag breakpoint causes the current opcode to be marked as it enters the instruction queue. If a tagged opcode reaches the end of the pipe, the CPU executes a BGND instruction to go to Active Background mode rather than executing the tagged opcode. A force-type breakpoint causes the CPU to finish the current instruction and then go to Active Background mode.

If the background mode has not been enabled (ENBDM = 1) by a serial WRITE_CONTROL command through the BKGD pin, the CPU will execute an SWI instruction instead of going to Active Background mode.

15.4 Register definitions

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

Refer to the high-page register summary in the device overview chapter of this data sheet for the absolute address assignments for all DBG registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

15.4.1 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.

15.4.2 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 152. BDC Status and Control Register (BDCSCR)

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

= Unimplemented or Reserved

Table 153. 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

Table 153. BDCSCR register field descriptions (continued)

Field	Description
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

15.4.3 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, refer to Section 15.2.4.

15.4.4 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 154. System Background Debug Force Reset register (SBDFR)

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

= Unimplemented or Reserved

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

Table 155. 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.

15.4.5 DBG Registers and Control Bits

The debug module includes nine bytes of register space for three 16-bit registers and three 8-bit control and status registers. These registers are located in the high register space of the normal memory map so they are accessible to normal application programs. These registers are rarely if ever accessed by normal user application programs with the possible exception of a ROM patching mechanism that uses the breakpoint logic.

15.4.5.1 Debug Comparator A High register (DBGCAH)

This register contains compare value bits for the high-order eight bits of comparator A. This register is forced to 0x00 at reset and can be read at any time or written at any time unless ARM = 1.

15.4.5.2 Debug Comparator A Low register (DBGCAL)

This register contains compare value bits for the low-order eight bits of comparator A. This register is forced to 0x00 at reset and can be read at any time or written at any time unless ARM = 1.

15.4.5.3 Debug Comparator B High register (DBGCBH)

This register contains compare value bits for the high-order eight bits of comparator B. This register is forced to 0x00 at reset and can be read at any time or written at any time unless ARM = 1.

15.4.5.4 Debug Comparator B Low register (DBGCBL)

This register contains compare value bits for the low-order eight bits of comparator B. This register is forced to 0x00 at reset and can be read at any time or written at any time unless ARM = 1.

15.4.5.5 Debug FIFO High register (DBGFH)

This register provides read-only access to the high-order eight bits of the FIFO. Writes to this register have no meaning or effect. In the event-only trigger modes, the FIFO only stores data into the low-order byte of each FIFO word, so this register is not used and will read 0x00.

Reading DBGFH does not cause the FIFO to shift to the next word. When reading 16-bit words out of the FIFO, read DBGFH before reading DBGFL because reading DBGFL causes the FIFO to advance to the next word of information.

15.4.5.6 Debug FIFO Low register (DBGFL)

This register provides read-only access to the low-order eight bits of the FIFO. Writes to this register have no meaning or effect.

Reading DBGFL causes the FIFO to shift to the next available word of information. When the debug module is operating in event-only modes, only 8-bit data is stored into the FIFO (high-order half of each FIFO word is unused). When reading 8-bit words out of the FIFO, simply read DBGFL repeatedly to get successive bytes of data from the FIFO. It isn't necessary to read DBGFH in this case.

Do not attempt to read data from the FIFO while it is still armed (after arming but before the FIFO is filled or ARMF is cleared) because the FIFO is prevented from advancing during reads of DBGFL. This can interfere with normal sequencing of reads from the FIFO.

Reading DBGFL while the debugger is not armed causes the address of the most-recently fetched opcode to be stored to the last location in the FIFO. By reading DBGFH then DBGFL periodically, external host software can develop a profile of program execution. After eight reads from the FIFO, the ninth read will return the information that was stored as a result of the first read. To use the profiling feature, read the FIFO eight times without using the data to prime the sequence and then begin using the data to get a delayed picture of what addresses were being executed. The information stored into the FIFO on reads of DBGFL (while the FIFO is not armed) is the address of the most-recently fetched opcode.

15.4.5.7 Debug Control register (DBGC)

This register can be read or written at any time.

Table 156. Debug Control register (DBGC)

	7	6	5	4	3	2	1	0
R W	DBGEN	ARM	TAG	BRKEN	RWA	RWAEN	RWB	RWBEN
Reset	0	0	0	0	0	0	0	0

Table 157. DBGC register field descriptions

Field	Description
7 DBGEN	**Debug Module Enable** — Used to enable the debug module. DBGEN cannot be set to 1 if the MCU is secure. 0 DBG disabled 1 DBG enabled
6 ARM	**Arm Control** — Controls whether the debugger is comparing and storing information in the FIFO. A write is used to set this bit (and ARMF) and completion of a debug run automatically clears it. Any debug run can be manually stopped by writing 0 to ARM or to DBGEN. 0 Debugger not armed 1 Debugger armed
5 TAG	**Tag/Force Select** — Controls whether break requests to the CPU will be tag or force type requests. If BRKEN = 0, this bit has no meaning or effect. 0 CPU breaks requested as force type requests 1 CPU breaks requested as tag type requests
4 BRKEN	**Break Enable** — Controls whether a trigger event will generate a break request to the CPU. Trigger events can cause information to be stored in the FIFO without generating a break request to the CPU. For an end trace, CPU break requests are issued to the CPU when the comparator(s) and R/W meet the trigger requirements. For a begin trace, CPU break requests are issued when the FIFO becomes full. TRGSEL does not affect the timing of CPU break requests. 0 CPU break requests not enabled 1 Triggers cause a break request to the CPU
3 RWA	**R/W Comparison Value for Comparator A** — When RWAEN = 1, this bit determines whether a read or a write access qualifies comparator A. When RWAEN = 0, RWA and the R/W signal do not affect comparator A. 0 Comparator A can only match on a write cycle 1 Comparator A can only match on a read cycle
2 RWAEN	**Enable R/W for Comparator A** — Controls whether the level of R/W is considered for a comparator A match. 0 R/W is not used in comparison A 1 R/W is used in comparison A
1 RWB	**R/W Comparison Value for Comparator B** — When RWBEN = 1, this bit determines whether a read or a write access qualifies comparator B. When RWBEN = 0, RWB and the R/W signal do not affect comparator B. 0 Comparator B can match only on a write cycle 1 Comparator B can match only on a read cycle
0 RWBEN	**Enable R/W for Comparator B** — Controls whether the level of R/W is considered for a comparator B match. 0 R/W is not used in comparison B 1 R/W is used in comparison B

15.4.5.8 Debug Trigger Register (DBGT)

This register can be read any time, but may be written only if ARM = 0, except bits 4 and 5 are hard-wired to 0s.

Table 158. Debug Trigger Register (DBGT)

	7	6	5	4	3	2	1	0
R	TRGSEL	BEGIN	0	0	TRG3	TRG2	TRG1	TRG0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 159. DBGT register field descriptions

Field	Description
7 TRGSEL	**Trigger Type** — Controls whether the match outputs from comparators A and B are qualified with the opcode tracking logic in the debug module. If TRGSEL is set, a match signal from comparator A or B must propagate through the opcode tracking logic and a trigger event is only signalled to the FIFO logic if the opcode at the match address is actually executed. 0 Trigger on access to compare address (force) 1 Trigger if opcode at compare address is executed (tag)
6 BEGIN	**Begin/End Trigger Select** — Controls whether the FIFO starts filling at a trigger or fills in a circular manner until a trigger ends the capture of information. In event-only trigger modes, this bit is ignored and all debug runs are assumed to be begin traces. 0 Data stored in FIFO until trigger (end trace) 1 Trigger initiates data storage (begin trace)
3:0 TRG[3:0]-+-	**Select Trigger Mode** — Selects one of nine triggering modes, as described below. 0000A-only 0001 A OR B 0010A Then B 0011Event-only B (store data) 0100A then event-only B (store data) 0101A AND B data (full mode) 0110A AND NOT B data (full mode) 0111Inside range: A ≤ address ≤ B 1000Outside range: address < A or address > B 1001 – 1111 (No trigger)

15.4.5.9 Debug Status register (DBGS)

This is a read-only status register.

Table 160. Debug Status register (DBGS)

	7	6	5	4	3	2	1	0
R	AF	BF	ARMF	0	CNT3	CNT2	CNT1	CNT0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 161. DBGS register field descriptions

Field	Description
7 AF	**Trigger Match A Flag** — AF is cleared at the start of a debug run and indicates whether a trigger match A condition was met since arming. 0 Comparator A has not matched 1 Comparator A match
6 BF	**Trigger Match B Flag** — BF is cleared at the start of a debug run and indicates whether a trigger match B condition was met since arming. 0 Comparator B has not matched 1 Comparator B match
5 ARMF	**Arm Flag** — While DBGEN = 1, this status bit is a read-only image of ARM in DBGC. This bit is set by writing 1 to the ARM control bit in DBGC (while DBGEN = 1) and is automatically cleared at the end of a debug run. A debug run is completed when the FIFO is full (begin trace) or when a trigger event is detected (end trace). A debug run can also be ended manually by writing 0 to ARM or DBGEN in DBGC. 0 Debugger not armed 1 Debugger armed

Table 161. DBGS register field descriptions (continued)

Field	Description
3:0 CNT[3:0]	**FIFO Valid Count** — These bits are cleared at the start of a debug run and indicate the number of words of valid data in the FIFO at the end of a debug run. The value in CNT does not decrement as data is read out of the FIFO. The external debug host is responsible for keeping track of the count as information is read out of the FIFO. 0000Number of valid words in FIFO = No valid data 0001Number of valid words in FIFO = 1 0010Number of valid words in FIFO = 2 0011Number of valid words in FIFO = 3 0100Number of valid words in FIFO = 4 0101Number of valid words in FIFO = 5 0110Number of valid words in FIFO = 6 0111Number of valid words in FIFO = 7 1000Number of valid words in FIFO = 8

16 Battery Charge Consumption Modeling

The supply current consumed by the MPXx85/86xxD can be estimated using the following basic model.

16.1 Standby current

The overall charge consumed by the standby features is:

$$Q_{STDBY} = t_{TOT} \times \frac{(I_{STDBY} + I_{LF})}{1000} \qquad \textit{Eqn. 16}$$

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

16.2 Measurement events

The overall charge consumed by the measurements is:

$$Q_{MEAS} = \frac{1}{1000} \times (n_{PRESS} \times Q_{PRESS} + n_{TEMP} \times Q_{TEMP} + n_{VOLT} \times Q_{VOLT}) \qquad \textit{Eqn. 17}$$

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

16.3 Transmission events

The overall charge consumed by the transmissions is:

$$Q_{XMT} = \frac{Q_{FRM}}{1000} \times F \times n_{XMT} \qquad \textit{Eqn. 18}$$

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

16.4 Total consumption

The overall charge consumed is:

$$Q_{TOT} = \frac{(Q_{STDBY} + Q_{MEAS} + Q_{XMT})}{1 - Y \times (SD/100)} \qquad \textit{Eqn. 19}$$

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}.

17 Electrical Specifications

17.1 Maximum ratings

Maximum ratings are the extreme limits to which the device can be exposed without permanently damaging it. The device contains circuitry to protect the inputs against damage from high static voltages; however, do not apply voltages higher than those shown in the table below. Keep V_{IN} and V_{OUT} within the range $V_{SS} \le (V_{IN}$ or $V_{OUT}) \le V_{DD}$.

Table 162. Maximum ratings

#	Rating	Symbol	Value	Unit	
1	Supply Voltage (V_{DD}, AV_{DD})	V_{DD}	-0.3 to +3.8	V	(1)
	Input Voltage				
2	X1	V_{IN}	-0.3 to V_{DD}+0.3	V	(1)
3	LFA, LFB, PTA0, PTA1, PTA2, PTA3	V_{IN}	-0.3 to V_{DD}+0.3	V	(1)
4	BKGD, $\overline{\text{RESET}}$	V_{IN}	-0.3 to V_{DD}+0.3	V	(1)
	Input Current				
5	X1	I_{IN}	±10	mA	(1)
6	LFA, LFB, PTA0, PTA1, PTA2, PTA3	I_{IN}	±10	mA	(1)
7	BKGD, $\overline{\text{RESET}}$	I_{IN}	±10	mA	(1)
	Substrate Current Injection				
	Current from any pin to V_{SS} - 0.3 VDC				
8	XI, PTA0, PTA1, PTA2, PTA3, BKGD, $\overline{\text{RESET}}$	I_{SUB}	600	µA	(1)
9	LFA, LFB	I_{SUB}	2	mA	(1)
	Latchup Current				
10	Current to/from any pin to supply rails + 0.3 VDC	I_{LATCH}	±100	mA	(1)
	Electrostatic Discharge				
11	Human Body Model (HBM), all pins other than RF	V_{ESD}	±2000	V	(1)
12	Human Body Model (HBM), RF pin	V_{ESD}	±3000	V	(1)
	Charged Device Model (CDM),				
13	Pins 1, 8, 9, 10, 16, 17, 24, 25, 32	V_{ESD}	±750	V	(1)
14	Pins 2-7, 10-15, 18-23, 26-31	V_{ESD}	±500	V	(1)
15	Machine Model (MM)	V_{ESD}	±150	V	(1)
16	Maximum Storage Temperature Range	T_{stg}	-50 to +150	°C	(1)

Note: Refer to page 173 for description of notes.

17.2 Operating range

The limits normally expected in the application which define range of operation.

Table 163. Operating range

#	Characteristic	Symbol	Min	Typ	Max	Units	
	Operating Supply Voltage (V_{DD} = AV_{DD})		V_L		V_H		
	Measurements						
17	Pressure, Temperature, Acceleration	V_{DD}	2.3	3.0	3.6	V	(2)
18	Voltage	V_{DD}	1.8	3.0	3.6	V	(2)
19	LFR operation (-20 to +85°C)	V_{DD}	2.3	3.0	3.6	V	(2)
20	RF Transmissions	V_{DD}	1.8	3.0	3.6	V	(2)
	MCU operation (CPU, ADC10, RAM, TPM1)						
21	FLASH write (-40 to +125°C)	V_{DD}	2.3	3.0	3.6	V	(2)
22	FLASH write (-20 to +85°C)	V_{DD}	2.1	3.0	3.6	V	(2)
23	FLASH read	V_{DD}	1.8	3.0	3.6	V	(2)
24	Parameter registers data retention	V_{DD}	1.2	—	—	V	(2)
	Operating Temperature Range		T_L		T_H		
25	Continuous Temperature Range	T_A	-40	—	+125	°C	(1)
26	Stop1 mode (5)	T_A	-40	—	+150	°C	(1)

Note: Refer to page 173 for description of notes.

17.3 Electrical characteristics

$1.8 \le V_{DD} \le 3.6$, $T_L \le T_A \le T_H$, unless otherwise specified.

Table 164. Electrical characteristics

#	Characteristic	Symbol	Min	Typ	Max	Units	
27	Output High Voltage (I_{Load} = 5 mA) LFA, LFB, PTA0, PTA1, PTA2, PTA3	V_{OH}	V_{DD}-0.35	—	—	V	(1)
28	Output Low Voltage (I_{Load} = -5 mA) LFA, LFB, PTA0, PTA1, PTA2, PTA3	V_{OL}	—	—	0.35	V	(1)
29	Input High Voltage ($2.3 < V_{DD} \ge V_H$) LFA, LFB, PTA0, PTA1, PTA2, PTA3, BKGD	V_{IH}	0.7 x V_{DD}	—	—	V	(1)
30	Input High Voltage ($V_L \le V_{DD} \le 2.3$) LFA, LFB, PTA0, PTA1, PTA2, PTA3	V_{IH}	0.85 x V_{DD}	—	—	V	(1)
31	Input Low Voltage ($2.3 < V_{DD} \le V_H$) LFA, LFB, PTA0, PTA1, PTA2, PTA3, BKGD	V_{IL}	V_{SS}	—	0.35 x V_{DD}	V	(1)
32	Input Low Voltage ($V_L \le V_{DD} \le 2.3$) LFA, LFB, PTA0, PTA1, PTA2, PTA3	V_{IL}	V_{SS}	—	0.28 x V_{DD}	V	(1)
	Input High Current (at V_{DD})						
33	LFA, LFB	I_{IH}	+1	—	+4	µA	(1)
34	BKGD	I_{IH}	-1	—	+1	µA	(1)
35	PTA0, PTA1, PTA2, PTA3 (pullup off)	I_{IH}	-1	—	+1	µA	(1)
36	PTA0, PTA1, PTA2, PTA3 (pullup active)	I_{IH}	-1	—	+1	µA	(1)
	Input Low Current (at V_{SS})						
37	LFA, LFB	I_{IL}	-0.8	—	0	µA	(1)
38	BKGD	I_{IL}	-0.8	—	0	µA	(1)
39	PTA0, PTA1, PTA2, PTA3 (pulldown off)	I_{IL}	-1	—	+1	µA	(1)
40	PTA0, PTA1, PTA3 (pulldown active)	I_{IL}	0	—	+120	µA	(1)
41	PTA2 (pulldown active)	I_{IL}	-10	—	+10		
	Pin Capacitance (3 V)						
42	BKGD, PTA0, PTA1, PTA2, PTA3	C	0	—	15	pF	(1)

Note: Refer to page 173 for description of notes.

17.4 Power consumption (MCU)

$1.8 \le V_{DD} \le 3.6$, $T_A = -40$ to 125°C unless otherwise specified.

Table 165. Power consumption

#	Characteristic	Symbol	Min	Typ	Max	Units	
	Standby supply current						
	Stop1 mode, LFR, LVD and TR all off						
43	T_A = -40°C, V_{DD} = 3.0 V	I_{STDBY}	—	—	0.9	µA	(1)
44	T_A = 0°C, V_{DD} = 3.0 V	I_{STDBY}	—	—	0.9	µA	(1)
45	T_A = 25°C, V_{DD} = 1.8 V	I_{STDBY}	—	—	0.8	µA	(1)
46	T_A = 25°C, V_{DD} = 3.0 V	I_{STDBY}	—	0.5	0.7	µA	(1)
47	T_A = 70°C, V_{DD} = 3.0 V	I_{STDBY}	—	—	1.5	µA	(1)
48	T_A = 125°C, V_{DD} = 3.0 V	I_{STDBY}	—	—	13	µA	(1)
	Standby supply current						
	Stop4 mode, LFR and TR off, LVD or RFLVD on						
49	T_A = -40°C, V_{DD} = 3.0 V	I_{STDBY}	—	—	100	µA	(1)
50	T_A = 0°C, V_{DD} = 3.0 V	I_{STDBY}	—	—	100	µA	(1)
51	T_A = 25°C, V_{DD} = 1.8 V	I_{STDBY}	—	—	95	µA	(1)
52	T_A = 25°C, V_{DD} = 3.0 V	I_{STDBY}	—	73	95	µA	(1)
53	T_A = 70°C, V_{DD} = 3.0 V	I_{STDBY}	—	—	100	µA	(1)
54	T_A = 125°C, V_{DD} = 3.0 V	I_{STDBY}	—	—	120	µA	(1)
	MCU Operate Current (V_{DD}=1.8V, T_A=25°C to 125°C)						
	Instruction Speed = 0.333 MIP/f_{BUS}						
55	0.5 MHz f_{BUS}, BUSCLKS[1:0] = 11	I_{DD}	—	—	0.8	mA	(1)
56	1 MHz f_{BUS}, BUSCLKS[1:0] = 10	I_{DD}	—	—	1.1	mA	(1)
57	2 MHz f_{BUS}, BUSCLKS[1:0] = 01	I_{DD}	—	—	1.7	mA	(1)
58	4 MHz f_{BUS}, BUSCLKS[1:0] = 00	I_{DD}	—	—	2.9	mA	(1)
	MCU Operate Current (V_{DD}=3V, T_A= -40°C to 25°C)						
	Instruction Speed = 0.333 MIP/f_{BUS}						
59	0.5 MHz f_{BUS}, BUSCLKS[1:0] = 11	I_{DD}	—	—	0.8	mA	(1)
60	1 MHz f_{BUS}, BUSCLKS[1:0] = 10	I_{DD}	—	—	1.1	mA	(1)
61	2 MHz f_{BUS}, BUSCLKS[1:0] = 01	I_{DD}	—	—	1.7	mA	(1)
62	4 MHz f_{BUS}, BUSCLKS[1:0] = 00	I_{DD}	—	—	2.9	mA	(1)
	MCU Operate Current (V_{DD}=3V, T_A=25°C to 125°C)						
	Instruction Speed = 0.333 MIP/f_{BUS}						
63	0.5 MHz f_{BUS}, BUSCLKS[1:0] = 11	I_{DD}	—	0.68	0.8	mA	(2)
64	1 MHz f_{BUS}, BUSCLKS[1:0] = 10	I_{DD}	—	0.94	1.1	mA	(2)
65	2 MHz f_{BUS}, BUSCLKS[1:0] = 01	I_{DD}	—	1.46	1.7	mA	(2)
66	4 MHz f_{BUS}, BUSCLKS[1:0] = 00	I_{DD}	—	2.50	2.9	mA	(2)
	Standby Current Adder for Temperature Restart						
67	V_{DD} = 1.8V, T_A = 125°C	I_{DD}	—	10	15	µA	(1)
68	V_{DD} = 3.0V, T_A= 125°C	I_{DD}	—	10	15	µA	(1)
	MCU Wake-Up Consumption (+25°C, 3 V)						
69	From Stop1 to first instruction, f_{BUS} = 4 MHz	Q_{WAKE}	0	0.05	0.1	µA-sec	(1)
	External Battery Model						
70	Series impedance at end-of-life	Z_{EOL}	—	—	60	ohm	(4)
71	Open circuit voltage at end-of-life	V_{EOL}	2.7	—	—	V	(4)

Note: Refer to page 173 for description of notes.

17.5 Control timing

$1.8 \le V_{DD} \le 3.6$, $T_L \le T_A \le T_H$, unless otherwise specified.

Table 166. Control timing

#	Characteristic	Symbol	Min	Typ	Max	Units	
	Internal Clock Frequency						
72	Initial startup frequency	$f_{OSCINIT}$	6	8	10	MHz	(2)
73	Final frequency	f_{OSC}	7	8	8.1	MHz	(2)
74	FLL stabilization time (see Figure 51)	t_{OSCSU}	—	300	1000	µSec	(1)
75	MCU Bus Frequency	f_{BUS}	—	0.5 f_{OSC}	—	MHz	(2)
76	Medium frequency clock (MFO)	f_{MFO}	110	—	130	kHz	(2)
	Low-frequency clock (LFO)						
77	Limited temperature range, 0°C to +70°C	f_{LFO}	833	—	1250	Hz	(1)
78	Full temperature range, -40°C to +125°C	f_{LFO}	770	—	1429	Hz	(1)
79	LFR clock (derived from 125 kHz LFRO)	f_{LFRO}	116.8	—	134.4	kHz	(1)
	Power-On Reset Response						
80	Supply voltage rise time	t_{VDDR}	—	—	1	sec	(1)
81	Recovery time below V_{DD} = 0.5 V	t_{VDDOFF}	0	—	70	µsec	(1)
	MCU Wake-up Time (see Figure 51)						
82	From Stop1 to first instruction, f_{BUS} = 4 MHz	$t_{MCUWAKE}$	—	50	70	µSec	(1)
83	From Stop4 to first instruction, f_{BUS} = 4 MHz	$t_{MCUWAKE}$	—	100	140	µSec	(1)
84	FLASH Data Retention Time	t_{DR}	10	—	—	year	(20)

Note: Refer to page 173 for description of notes.

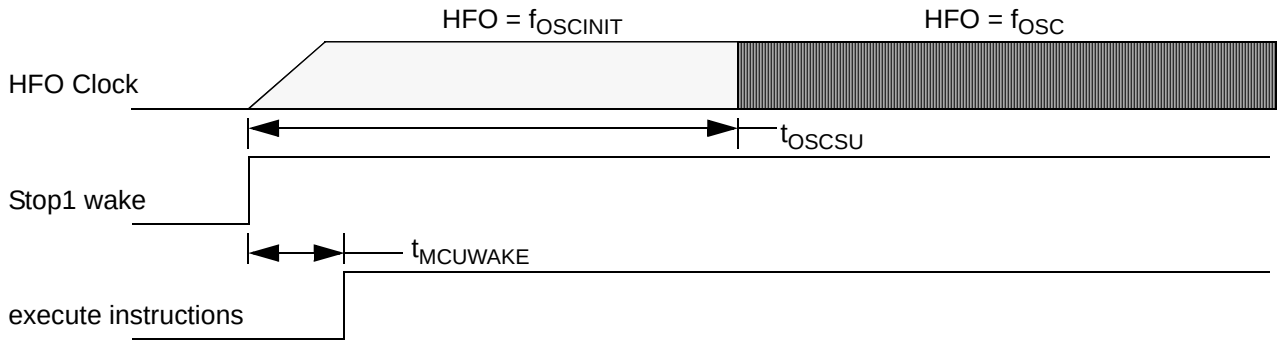


Figure 51. MCU startup delays

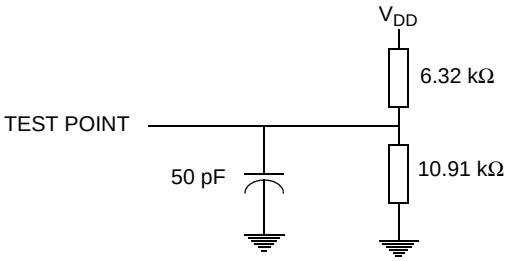


Figure 52. Control timing test load for digital Pins

17.6 Voltage measurement characteristics

$1.8 \le V_{DD} \le 3.6$, $T_L \le T_A \le T_H$, unless otherwise specified.

Table 167. Voltage measurement characteristics

#	Characteristic	Symbol	Min	Typ	Max	Units	
	Lower LVD detect threshold(15)						
85	V_{DD} falling	V_{LVDL}	1.79	1.88	1.96	V	(2)
86	V_{DD} rising	V_{LVDL}	1.87	—	2.03	V	(2)
87	Voltage drop detection time(16)	t_{LVDL}	—	—	10	µsec	(1)
	Higher LVD detect threshold(15)						
88	V_{DD} falling	V_{LVDH}	2.05	—	2.3	V	(2)
89	V_{DD} rising	V_{LVDH}	2.12	—	2.3	V	(2)
90	Voltage drop detection time(16)	t_{LVDH}	—	—	10	µsec	(1)
	Lower LVW detect threshold(15) (LVWV = 0)						
91	V_{DD} falling	V_{LVWL}	2.05	—	2.3	V	(2)
92	V_{DD} rising	V_{LVWL}	2.12	—	2.3	V	(2)
93	Voltage drop detection time(16)	t_{LVWL}	—	—	10	µsec	(1)
	Higher LVW detect threshold(15) (LVWV = 1)						
94	V_{DD} falling	V_{LVWH}	2.28	—	2.54	V	(2)
95	V_{DD} rising	V_{LVWH}	2.34	—	2.61	V	(2)
96	Voltage drop detection time(16)	t_{LVWH}	—	—	10	µsec	(1)
	RF LVD detect threshold(15)						
97	V_{DD} falling	V_{LVDRF}	1.72	1.88	1.95	V	(2)
98	Voltage drop detection time	t_{LVWRF}	—	—	10	µsec	(1)
	Power-on reset voltage						
99	Rising voltage to recover	V_{PORR}	—	—	2.1	V	(1)
100	Falling voltage to reset	V_{PORF}	0.9	—	—	V	(1)
	Internal voltage (V_{DD}, monotonic response)						
101	V_{CODE} = 0	V_{INT}	—	FAULT	—	V	(1)
102	V_{CODE} = 58	V_{INT}	1.6	1.8	2.0	V	(1)
103	V_{CODE} = 88	V_{INT}	2.0	2.1	2.2	V	(1)
104	V_{CODE} = 108	V_{INT}	2.2	2.3	2.4	V	(1)
105	V_{CODE} = 128	V_{INT}	2.4	2.5	2.6	V	(3)
106	V_{CODE} = 158	V_{INT}	2.7	2.8	2.9	V	(1)
107	V_{CODE} = 208	V_{INT}	3.2	3.3	3.4	V	(1)
108	V_{CODE} = 238	V_{INT}	3.5	3.6	3.7	V	(1)
109	V_{CODE} = 255	V_{INT}	—	FAULT	—	V	(1)
110	Voltage sensitivity at 25°C, 3.0 V	ΔV	—	10	—	mV/count	(1)
	External voltage (PTA[1:0], monotonic response, conversion is ratiometric to V_{DD})						
111	Output code	n	—	$1023(V_{IN}/V_{DD})$	—	count	(4)
112	Voltage sensitivity at 25°C, 3.0 V	ΔV_{EXT}	—	$V_{DD}/1023$	—	V/count	(1)
113	ADC INL	INL	-1	—	+1	LSB	(1)
114	ADC DNL	DNL	-1	—	+1	LSB	(1)
	Voltage measurement (internal voltage or external pin)						
115	Sensor measurement time (7)	t_{VM}	—	0.26	—	mSec	(1)
116	Peak current (9)	I_V	—	3.4	—	mA	(1)
	Power consumption						
117	Raw measurement, 10-bit	Q_V	—	0.13	0.3	µA-sec	(1)
118	Compensation, 8-bit	Q_V	—	0.40	0.64	µA-sec	(1)
119	Basic compensated reading, 8-bit	Q_V	—	0.53	0.9	µA-sec	(1)
120	Full compensated reading, 8-bit	Q_V	—	0.53	0.9	µA-sec	(1)

Note: Refer to page 173 for description of notes.

17.7 Temperature measurement characteristics

$2.3 \le V_{DD} \le 3.3$, $T_L \le T_A \le T_H$, unless otherwise specified.

Table 168. Temperature measurement characteristics

#	Characteristic	Symbol	Min	Typ	Max	Units	
	Temperature measurement (monotonic response)(11)						
121	T_{CODE} = 0	T	—	FAULT	—	°C	(1)
122	T_{CODE} = 15	T	-45	-40	-35	°C	(1)
123	T_{CODE} = 35	T	-23	-20	-17	°C	(3)
124	T_{CODE} = 55	T	-3	0	3	°C	(1)
125	T_{CODE} = 84	T	26	29	32	°C	(3)
126	T_{CODE} = 125	T	67	70	73	°C	(1)
127	T_{CODE} = 140	T	81	85	89	°C	(3)
128	T_{CODE} = 180	T	120	125	130	°C	(1)
129	T_{CODE} = 255	T	—	FAULT	—	°C	(1)
130	Temperature sensitivity, 3 V, 25°C	ΔT	—	1.0	—	°C/count	(1)
131	Temperature measurement stability range (10)	T_{STAB}	—	—	2	count	(1)
	Thermal shutdown recovery (TRE = 1)						
132	High re-arming temperature(13) (TRH = 1)	T_{REARMH}	—	—	118	°C	(1)
133	High reset temperature (TRH = 1)	T_{RESETH}	90	—	—	°C	(1)
134	Low re-arming temperature(13) (TRH = 0)	T_{REARML}	-90	—	—	°C	(1)
135	Low reset temperature (TRH = 0)	T_{RESETL}	—	—	4	°C	(1)
	Temperature measurement						
136	Sensor measurement time (7)	t_{TM}	—	—	1.0	mSec	(1)
137	Peak current (8)	I_T	—	—	3.0	mA	(1)
	Power consumption(22)						
138	Raw measurement, 12-bit	Q_T	—	0.19	0.3	µA-sec	(1)
139	Compensation, 8-bit	Q_T	—	0.43	0.6	µA-sec	(1)
140	Basic compensated reading, 8-bit	Q_T	—	0.62	0.82	µA-sec	(1)
141	Full compensated reading, 8-bit	Q_T	—	0.62	0.82	µA-sec	(1)

Note: Refer to page 173 for description of notes.

17.8 Pressure measurement characteristic (100 to 450 kPa Range)

$2.3 \le V_{DD} \le 3.3$, $T_L \le T_A \le T_H$, unless otherwise specified.

Pressure accuracy specified for pressure drops slower than 1 kPa/sec. For faster pressure drops refer to Appendix C, “Rapid Decompression Events”.

Table 169.

#	Characteristic	Symbol	Min	Typ	Max	Units	
	Pressure measurement (6)						
	0°C ≤ T_A ≤ +70°C						
142	P_{CODE} = 0	P	—	FAULT	—	kPa	(1)
143	P_{CODE} = 1	P	93	100	107	kPa	(4)
144	P_{CODE} = 16	P	103	110	117	kPa	(3)
145	P_{CODE} = 365	P	343	350	357	kPa	(3)
146	P_{CODE} = 495	P	433	440	447	kPa	(1)
147	P_{CODE} = 510	P	443	450	457	kPa	(4)
148	P_{CODE} = 511	P	—	FAULT	—	kPa	(1)
	-20°C ≤ T_A < 0°C, 70°C < T_A ≤ 85°C						
149	P_{CODE} = 0	P	—	FAULT	—	kPa	(1)
150	P_{CODE} = 1	P	89.5	100	110.5	kPa	(4)
151	P_{CODE} = 16	P	99.5	110	120.5	kPa	(3)
152	P_{CODE} = 365	P	339.5	350	360.5	kPa	(3)
153	P_{CODE} = 495	P	429.5	440	450.5	kPa	(1)
154	P_{CODE} = 510	P	439.5	450	460.5	kPa	(4)
155	P_{CODE} = 511	P	—	FAULT	—	kPa	(1)
	-40°C ≤ T_A < -20°C, 85°C < T_A ≤ 125°C						
156	P_{CODE} = 0	P	—	FAULT	—	kPa	(1)
157	P_{CODE} = 1	P	83.2	100	116.8	kPa	(4)
158	P_{CODE} = 16	P	93.2	110	126.8	kPa	(3)
159	P_{CODE} = 365	P	333.2	350	366.8	kPa	(3)
160	P_{CODE} = 495	P	423.2	440	456.8	kPa	(1)
161	P_{CODE} = 510	P	433.2	450	466.8	kPa	(4)
162	P_{CODE} = 511	P	—	FAULT	—	kPa	(1)
163	Pressure sensitivity (100-450 kPa)	ΔP_{450}	—	0.688	—	kPa/count	(1)
164	Pressure measurement stability range(10)	P_{STAB}	—	—	6	count	(1)
	Pressure sensitivity to Z-axis acceleration(12)						
165	0-500 g	P_{ACC}	0	—	0	Pa/g	(1)
166	>500 g	P_{ACC}	-6.5	-4.5	-2	Pa/g	(1)
	Pressure measurement						
167	Sensor measurement time(7)	t_{PM}	—	—	3	mSec	(1)
168	Peak current (8)	I_P	—	—	3	mA	(1)
	Power consumption(22)						
169	Raw measurement, 10-bit	Q_P	—	1.87	3.0	µA-sec	(1)
170	Compensation, 9-bit	Q_P	—	1.70	2.3	µA-sec	(1)
171	Basic compensated reading, 9-bit	Q_P	—	3.56	4.9	µA-sec	(1)
172	Full compensated reading, 9-bit	Q_P	—	3.89	5.3	µA-sec	(1)

Note: Refer to page 173 for description of notes.

17.9 Pressure measurement characteristic (100 to 900 kPa Range)

$2.3 \le V_{DD} \le 3.3$, $T_L \le T_A \le T_H$, unless otherwise specified.

Pressure accuracy specified for pressure drops slower than 1 kPa/sec. For faster pressure drops refer to Appendix C, “Rapid Decompression Events”.

Table 170.

#	Characteristic	Symbol	Min	Typ	Max	Units	
	Pressure measurement [6]						
	0°C ≤ T_A ≤ +70°C						
173	P_{CODE} = 0	P	—	FAULT	—	kPa	(1)
174	P_{CODE} = 1	P	90	100	110	kPa	(4)
175	P_{CODE} = 14	P	110	120	130	kPa	(3)
176	P_{CODE} = 351	P	640	650	660	kPa	(3)
177	P_{CODE}= 497	P	870	880	890	kPa	(1)
178	P_{CODE} = 510	P	890	900	910	kPa	(4)
179	P_{CODE} = 511	P	—	FAULT	—	kPa	(1)
	-20°C ≤ T_A < 0°C, 70°C < T_A ≤ 85°C						
180	P_{CODE} = 0	P	—	FAULT	—	kPa	(1)
181	P_{CODE} = 1	P	85	100	115	kPa	(4)
182	P_{CODE} = 14	P	105	120	135	kPa	(3)
183	P_{CODE} = 351	P	635	650	665	kPa	(3)
184	P_{CODE} = 497	P	865	880	895	kPa	(1)
185	P_{CODE} = 510	P	885	900	915	kPa	(4)
186	P_{CODE} = 511	P	—	FAULT	—	kPa	(1)
	-40°C ≤ T_A < -20°C, 85°C < T_A ≤ 12°C						
187	P_{CODE} = 0	P	—	FAULT	—	kPa	(1)
188	P_{CODE} = 1	P	76	100	124	kPa	(4)
189	P_{CODE} = 14	P	96	120	144	kPa	(3)
190	P_{CODE} = 351	P	626	650	674	kPa	(3)
191	P_{CODE} = 497	P	856	880	904	kPa	(1)
192	P_{CODE} = 510	P	876	900	924	kPa	(4)
193	P_{CODE} = 511	P	—	FAULT	—	kPa	(1)
194	Pressure sensitivity (100-900 kPa)	ΔP_{900}	—	1.572	—	kPa/count	(1)
195	Pressure measurement stability range[10]	P_{STAB}	—	—	4	count	(1)
	Pressure sensitivity to Z-axis acceleration[12]						
196	0-500g	P_{ACC}	0	—	0	Pa/g	(1)
197	>500g	P_{ACC}	-6.5	-4.5	-2	Pa/g	(1)
	Pressure measurement						
198	Sensor measurement time [7]	t_{PM}	—	—	3	mSec	(1)
199	Peak current [8]	I_P	—	—	3	mA	(1)
	Power consumption[22]						
200	Raw measurement, 10-bit	Q_P	—	1.87	3.0	µA-sec	(1)
201	Compensation, 9-bit	Q_P	—	1.70	2.3	µA-sec	(1)
202	Basic compensated reading, 9-bit	Q_P	—	3.56	4.9	µA-sec	(1)
203	Full compensated reading	Q_P	—	3.89	5.3	µA-sec	(1)

Note: Refer to page 173 for description of notes.

17.10 Optional -210 to +240 g Z-axis acceleration sensor characteristics

$2.3 \le V_{DD} \le 3.3$, $T_L \le T_A \le T_H$, unless otherwise specified.

Table 171. Z-axis acceleration sensor characteristics

#	Characteristic	Symbol	Min	Typ	Max	Units	
	Acceleration measurement, Offset STEP = 7[9]						
204	A_{ZCODE} = 0	$A_{Z\text{-}7}$	—	FAULT	—	g	(1)
205	A_{ZCODE} = 1	$A_{Z\text{-}7}$	-5	0	+5	g	(3)
206	A_{ZCODE} = 255	$A_{Z\text{-}7}$	+22	+30	+38	g	(1)
207	A_{ZCODE} = 510	$A_{Z\text{-}7}$	+46	+60	+74	g	(1)
208	A_{ZCODE} = 511	$A_{Z\text{-}7}$	—	FAULT	—	g	(1)
209	Accel Sensitivity (0 to +60 g)	$\Delta A_{Z\text{-}7}$	—	0.118	—	g/count	(1)
210	Accel Sensitivity Variation	$\Delta\Delta A_{Z\text{-}7}$	-15	—	+15	%	(4)
	Acceleration measurement offset[9]						
211	Offset STEP = 0	$A_{Z\text{-}0}$	—	-210	—	g	(1)
212	Offset STEP = 1	$A_{Z\text{-}1}$	—	-180	—	g	(1)
213	Offset STEP = 2	$A_{Z\text{-}2}$	—	-150	—	g	(1)
214	Offset STEP = 3	$A_{Z\text{-}3}$	—	-120	—	g	(1)
215	Offset STEP = 4	$A_{Z\text{-}4}$	—	-90	—	g	(1)
216	Offset STEP = 5	$A_{Z\text{-}5}$	—	-60	—	g	(1)
217	Offset STEP = 6	$A_{Z\text{-}6}$	—	-30	—	g	(3)
218	**Offset STEP = 7, See lines 204 - 208 above.**	$A_{Z\text{-}7}$					
219	Offset STEP = 8	$A_{Z\text{-}8}$	—	30	—	g	(1)
220	Offset STEP = 9	$A_{Z\text{-}9}$	—	60	—	g	(1)
221	Offset STEP = 10	$A_{Z\text{-}10}$	—	90	—	g	(1)
222	Offset STEP = 11	$A_{Z\text{-}11}$	—	120	—	g	(1)
223	Offset STEP = 12	$A_{Z\text{-}12}$	—	150	—	g	(1)
224	Offset STEP = 13	$A_{Z\text{-}13}$	—	180	—	g	(1)
225	Offset STEP = 14	$A_{Z\text{-}14}$	—	210	—	g	(1)
226	Offset STEP = 15	$A_{Z\text{-}15}$	—	240	—	g	(1)
227	Acceleration measurement stability range[10]	A_{STAB}	—	—	5	count	(1)
228	Acceleration cross-axis sensitivity	A_{CROSS}	-5	—	+5	%	(4)
	Z-axis acceleration measurement						
229	Sensor measurement time [7]	t_{PM}	—	—	3	mSec	(1)
230	Peak current [8]	I_P	—	—	3	mA	(1)
	Power consumption						
231	Raw measurement, 10-bit	Q_P	—	1.90	3.0	µA-sec	(1)
232	Compensation, 9-bit	Q_P	—	1.77	2.3	µA-sec	(1)
233	Total compensated reading, 9-bit	Q_P	—	3.67	5.0	µA-sec	(1)
234	Total compensated reading, 9-bit, 3 V, 25°C	Q_P	—	4.00	5.4	µA-sec	(1)

Note: Refer to page 173 for description of notes.

17.11 Optional -270 to +350 g Z-axis acceleration sensor characteristics

$2.3 \le V_{DD} \le 3.3$, $T_L \le T_A \le T_H$, unless otherwise specified.

Table 172. Z-axis acceleration sensor characteristics

#	Characteristic	Symbol	Min	Typ	Max	Units	
	Acceleration measurement, Offset STEP = 6[9]						
235	A_{ZCODE} = 0	A_{Z-6}	—	FAULT	—	g	(1)
236	A_{ZCODE} = 1	A_{Z-6}	-51.9	-42.9	-33.2	g	(3)
237	A_{ZCODE} = 256	A_{Z-6}	-6	0	+6	g	(1)
238	A_{ZCODE} = 510	A_{Z-6}	+33.0	+42.1	+51.8	g	(1)
239	Accel Sensitivity (1 to 510 counts)	ΔA_{Z-6}	—	0.167	—	g	(1)
240	Accel Sensitivity Variation where x = STEP number 0 through 15	$\Delta\Delta A_{Z-x}$	-15	0	15	%t	(1)
	Acceleration measurement offset[9]						
241	Offset STEP = 0	A_{Z-0}	—	-231	—	g	(1)
242	Offset STEP = 1	A_{Z-1}	—	-193	—	g	(1)
243	Offset STEP = 2	A_{Z-2}	—	-156	—	g	(1)
244	Offset STEP = 3	A_{Z-3}	—	-117	—	g	(1)
245	Offset STEP = 4	A_{Z-4}	—	-78	—	g	(1)
246	Offset STEP = 5	A_{Z-5}	—	-39	—	g	(1)
247	**Offset STEP = 6, See lines 235 - 238 above.**	A_{Z-6}					
248	Offset STEP = 7	A_{Z-7}	—	38	—	g	(1)
249	Offset STEP = 8	A_{Z-8}	—	78	—	g	(1)
250	Offset STEP = 9	A_{Z-9}	—	118	—	g	(1)
251	Offset STEP = 10	A_{Z-10}	—	158	—	g	(1)
252	Offset STEP = 11	A_{Z-11}	—	198	—	g	(1)
253	Offset STEP = 12	A_{Z-12}	—	238	—	g	(1)
254	Offset STEP = 13	A_{Z-13}	—	279	—	g	(1)
255	Offset STEP = 14	A_{Z-14}	—	319	—	g	(1)
256	Offset STEP = 15	A_{Z-15}	—	359	—	g	(1)
257	Acceleration measurement stability range[10]	A_{STAB}	—	—	4	count	(1)
258	Acceleration cross-axis sensitivity	A_{CROSS}	-5	—	5	%	(4)
	Z-axis acceleration measurement						
259	Sensor measurement time [7]	t_{AM}	—	—	3	mSec	(1)
260	Peak current[8]	I_A	—	—	3	mA	(1)
	Power consumption						
261	Raw measurement, 10-bit	Q_A	—	1.90	2.9	µA-sec	(1)
262	Compensation, 9-bit	Q_A	—	1.77	2.1	µA-sec	(1)
263	Total compensated reading, 9-bit	Q_A	—	3.67	5.0	µA-sec	(1)
264	Total compensated reading, 9-bit, 3 V, 25°C	Q_A	—	4.00	5.4	µA-sec	(1)

Note: Refer to page 173 for description of notes.

17.12 X-axis acceleration sensor characteristics

$2.3 \le V_{DD} \le 3.3$, $T_L \le T_A \le T_H$, unless otherwise specified.

Table 173. X-axis acceleration sensor characteristics

#	Characteristic	Symbol	Min	Typ	Max	Units	
	Acceleration Measurement, Offset = 7[9]						
265	A_{XCODE} = 0	A_{X-7}	—	FAULT	—	g	(1)
266	A_{XCODE} = 1	A_{X-7}	-14	-10	-6	g	(1)
267	A_{XCODE} = 256	A_{X-7}	-4	0	+4	g	(3)
268	A_{XCODE} = 510	A_{X-7}	+6	+10	+14	g	(1)
269	A_{XCODE} = 511	A_{X-7}	—	FAULT	—	g	(1)
270	Accel Sensitivity (-10g to +10g)	ΔA_{X-7}	—	0.039	—	g/count	(1)
271	Accel Sensitivity Variation	$\Delta\Delta A_{X-7}$	-15	—	+15	%	(4)
	Acceleration Measurement Offset [9]						
272	Offset STEP = 0	A_{X-0}	—	-70	—	g	(1)
273	Offset STEP = 1	A_{X-1}	—	-60	—	g	(1)
274	Offset STEP = 2	A_{X-2}	—	-50	—	g	(1)
275	Offset STEP = 3	A_{X-3}	—	-40	—	g	(1)
276	Offset STEP = 4	A_{X-4}	—	-30	—	g	(1)
277	Offset STEP = 5	A_{X-5}	—	-20	—	g	(1)
278	Offset STEP = 6	A_{X-6}	—	-10	—	g	(1)
279	**Offset STEP = 7, See lines 265 - 269 above.**	A_{X-7}					
280	Offset STEP = 8	A_{X-8}	—	10	—	g	(1)
281	Offset STEP = 9	A_{X-9}	—	20	—	g	(1)
282	Offset STEP = 10	A_{X-10}	—	30	—	g	(1)
283	Offset STEP = 11	A_{X-11}	—	40	—	g	(1)
284	Offset STEP = 12	A_{X-12}	—	50	—	g	(1)
285	Offset STEP = 13	A_{X-13}	—	60	—	g	(1)
286	Offset STEP = 14	A_{X-14}	—	70	—	g	(1)
287	Offset STEP = 15	A_{X-15}	—	80	—	g	(1)
288	Acceleration measurement stability range[10]	A_{STAB}	—	—	4	count	(1)
289	Acceleration cross-axis sensitivity	A_{CROSS}	-5	—	+5	%	(1)
	X-axis Acceleration Measurement						
290	Sensor measurement time [7]	t_{PM}	—	—	3	mSec	(1)
291	Peak current [8]	I_P	—	—	3	mA	(1)
	Power consumption						
292	Raw measurement, 10-bit	Q_P	—	1.90	3.0	µA-sec	(1)
293	Compensation, 9-bit	Q_P	—	1.77	2.3	µA-sec	(1)
294	Total compensated reading, 9-bit	Q_P	—	3.67	5.0	µA-sec	(1)
295	Total compensated reading, 9-bit, 3V, 25°C	Q_P	—	4.00	5.4	µA-sec	(1)

Note: Refer to page 173 for description of notes.

17.13 LFR sensitivity - low temp

2.3 ≤ V_{DD} ≤ 3.6, -20 ≤ T_A ≤ 85°C, unless otherwise specified. Detection and no detection criteria defined by note 19.

Table 174.

#	Characteristic	Symbol	Min	Typ	Max	Units	
	LFR input sensitivity in carrier mode 125 kHz carrier, LFCDTM = 256 µsec UOS Very-low sensitivity, SENS[1:0] = 00(20)						
296	Detect level, LFA:B	S_{DET_VL}	—	—	60	mV p-p	(1)
297	No detect level, LFA:B	S_{NODET_VL}	12	—	—	mV p-p	(1)
	Low sensitivity, SENS[1:0] = 01(20)						
298	Detect level, LFA:B	S_{DET_L}	—	—	14	mV p-p	(1)
299	No detect level, LFA:B	S_{NODET_L}	2	—	—	mV p-p	(1)
	High sensitivity, SENS[1:0] = 10, CHK125 = 1						
300	Detect level, LFA:B	S_{DET_H}	—	—	3.5	mV p-p	(1)
301	No detect level, LFA:B	S_{NODET_H}	0.35	—	—	mV p-p	(1)
	Sensitivity shift in high sensitivity, SENS[1:0] = 10, CHK125 = 0						
302	Detect level, LFA:B	S_{DET_H}	—	-0.3	—	mV p-p	(1)
303	No detect level, LFA:B	S_{NODET_H}	—	-0.3	—	mV p-p	(1)
	LFR input sensitivity in carrier mode 125 kHz carrier, LFCDTM = 8 msec Very-low sensitivity, SENS[1:0] = 00(20)						
304	Long carrier detect level	S_{DET_LG}	—	—	60	mV p-p	(1)
305	Long carrier no detect level	S_{NODET_LG}	12	—	—	mV p-p	(1)
	LFR input sensitivity in data mode 125 kHz carrier, LFCDTM = 64 µsec Valid reception of 10 consecutive frames of 16-bit Manchester bit preamble + 9T SYNC +16-bit ID + 4bytes of data. Continuously ON High sensitivity, SENS[1:0] = 10, CHK125 = 1						
306	Data detect level	S_{PER}	—	—	3.5	mV p-p	(2)
307	Data no detect level	S_{NOPER}	0.1	—	—	mV p-p	(2)

Note: Refer to page 173 for description of notes.

17.14 LFR characteristics

2.3 ≤ V_{DD} ≤ 3.6, -20 ≤ T_A ≤ 85°C, unless otherwise specified.

Table 175.

#	Characteristic	Symbol	Min	Typ	Max	Units	
308	Contents deleted						
309	Contents deleted						
310	Contents deleted						
	LF input signal characteristics						
311	Dynamic range, DEQEN = 0	V_{IN}	66	—	—	dB	(1)
312	Dynamic field variation (signals < 10 mV_{P-P})	V_{IN}	—	—	1.5	V_{P-P}/sec	(1)
	LF input signal characteristics (Manchester data mode)						
313	Modulation depth (Data 1 - Data 0)/Data 1	M_R	70	—	100	%	(1)
314	Data bit time	t_{DATA}	248	256	264	µsec	(1)
315	Bit duty cycle	M_{DC}	45	—	55	%	(1)
	LFR differential input (LFA to LFB, Figure 53)						
316	Differential resistance	R_{LFDF}	1	—	4	MΩ	(1)
317	Differential capacitance (C_3)	C_{LFDF}	2	3.8	6	pF	(1)
	LFR carrier frequency range VALEN = 1, LFCDTM = 256 µsec Always accepted carrier						
318	CHK125 = 0[20]	f_{LFC}	112.5	—	137.5	kHz	(1)
319	CHK125 = 1	f_{LFC}	121.25	—	128.75	kHz	(1)
320	Always rejected carrier	f_{LFC}	—	—	80	kHz	(1)
321	Always rejected carrier, CHK125 = 0	f_{LFC}	210	—	—	kHz	(1)
	VALEN = 0, LFCDTM = 256 msec						
322	Low cutoff freq, 40 mVp-p input, SENS[1:0] = 00	f_{LFC}	—	10	—	kHz	(1)
323	High cutoff freq, 40 mVp-p input, SENS[1:0] = 00	f_{LFC}	—	1000	—	kHz	(1)
324	LFR detector power up settling time (two LFO cycles)	t_{PU}	1.4	2.0	2.6	msec	(4)
	LFR preamble decoder settling time Data mode only, LFCDTM plus t_{DEC}						
325	LPSM = 0	t_{DEC}	—	—	200	µsec	(4)
326	LPSM = 1	t_{DEC}	—	—	400	µsec	(4)

Note: Refer to page 173 for description of notes.

17.15 LFR power consumption

$2.3 \le V_{DD} \le 3.6$, $-20 \le T_A \le 85°C$, unless otherwise specified.

Table 176.

#	Characteristic	Symbol	Min	Typ	Max	Units	
327	LFR supply current, carrier detect mode Monitor for carrier with VALEN = 0, LPSM = 1	I_{LFR}	—	4.0	5.5	µA	(1)
328	Frequency validation with VALEN = 1, LPSM = 1 and CHK125 = 0	I_{LFR}	—	5.8	7.5	µA	(1)
329	Frequency validation with VALEN = 1, LPSM = 1 and CHK125 = 1	I_{LFR}	—	6.0	8.0	µA	(1)
330	LFR Supply Current, Manchester Data Mode Decoding of data stream after carrier detected	I_{LFR}	—	12.0	15.5	µA	(1)

Note: Refer to page 173 for description of notes.

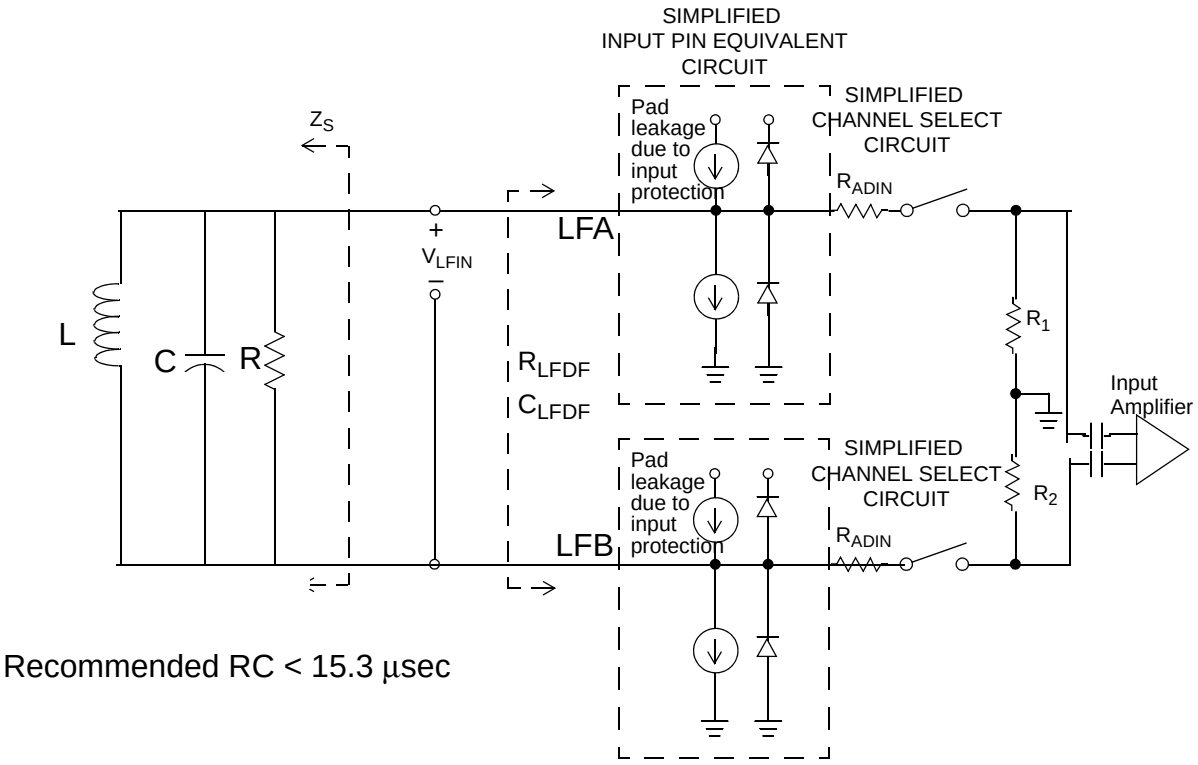


Figure 53. LFR detector input equivalent circuit

17.16 RF output stage

$1.8 \le V_{DD} \le 3.6$, $T_L \le T_A \le T_H$, unless otherwise specified.

Power output based on using Dynamic RF Power Correction firmware routine.

Output load of 50 Ω resistance as shown in Figure 56 unless otherwise specified.

MCU in Stop1 mode during all RF tests.

RF output will shutdown when the total RF I_{DD} causes V_{DD} to fall below 1.8 V ($V_{DD} = V_{BATT} - I_{BATT} \times R_{BATT}$). See Figure 57.

Table 177. RF output stage

#	Characteristic	Symbol	Min	Typ	Max	Units	
	Nominal output power with 50 Ω matching network (21)						
331	315 MHz, T_A =25°C, V_{DD} = 3V, PWR[4:0] = 01110	P_{RF}	4.0	5.2	6.0	dBm	(3)
332	434 MHz, T_A =25°C, V_{DD} = 3V, PWR[4:0] = 01111	P_{RF}	3.5	4.9	6.0	dBm	(3)
333	Nominal output power with 50 Ω matching network at maximum power step, PWR[4:0]=10100. Temperature and voltage range defined by Figure 54 and Figure 55.	P_{RF}	3	—	—	dBm	(1)
	Programmable power adjustment PWR[4:0] = 00000 through 11111						
334	Low-power mode (PWR[4:0] = 00000)	P_{LPM}	—	-10	—	dBm	(1)
335	Range (nominal, (PWR[4:0] > 00000)	P_{RF}	-1.5	—	8.0	dBm	(1)
336	Adjustment step (-1.5 TO +8 dBm))	P_{ADJ}	—	0.5	—	dBm	(1)
337	Programmable frequency steps Carrier and FSK deviation (AFREQ[12:0] and BFREQ[12:0])	f_{STEP}	—	3.174	—	kHz	(1)
338	External crystal frequency(15)	f_{XTAL}	—	26.000	—	MHz	(1)
339	PLL lock time	t_{LOCK}	—	45	—	µsec	(1)
	Ready to send delay						
340	From write to SEND bit rise of RCTS bit	t_{RCTS}	200	500	600	µsec	(1)
341	OOK modulation depth	M_{OOK}	50	—	—	dBc	(1)
	Manchester encoding data rate						
342	Bit rate(18)	DR	2	—	19.2	kbps	(1)
343	Accuracy (MCU direct with MFOCAL used)	DR	-5	—	+5	%	(1)
344	Accuracy (MCU direct without MFOCAL)	DR	-14	—	+4	%	(1)
345	Modulation duty-cycle (OOK and FSK)	DC	45	50	55	%	(1)
346	XTAL oscillator margin (over 25 MHz)(17)	ML	800	—	—	ohm	(4)
	Harmonic 2 level (315 and 434 MHz bands, with matching reference network)						
347	V_{DD} = 3 V, T_A = 25°C, PWR[4:0] = 01110	H2	—	-35	-22	dBc	(1)
348	$1.8 \le V_{DD} \le 3.6$, $T_L \le T_A \le T_H$, power step adjusted to reach the targeted power in each domain	H2	—	-25	-20	dBc	(1)
	Harmonic 4 level and above (315 and 434 MHz bands, with matching reference network)						
349	V_{DD} = 3 V, T_A = 25°C, PWR[4:0] = 01110	H4	—	—	-30	dBc	(1)
350	$1.8 \le V_{DD} \le 3.6$, $T_L \le T_A \le T_H$, power step adjusted to reach the targeted power in each domain	H4	—	—	-30	dBc	(1)
	Harmonic 3 level (315 and 434 MHz bands, with matching reference network)						
351	V_{DD} = 3 V, T_A = 25°C, PWR[4:0] = 01110	H3	—	-31	-28	dBc	(1)
352	$1.8 \le V_{DD} \le 3.6$, $T_L \le T_A \le T_H$, power step adjusted to reach the targeted power in each domain	H3	—	-27	-25	dBc	(1)

Table 177. RF output stage (continued)

#	Characteristic	Symbol	Min	Typ	Max	Units	
	Noise for BOOST = 0						
	Phase Noise (315 MHz)						
353	$f_{RF} \pm 10$ kHz	N_{PH}	—	-86	-78	dBc/Hz	(1)
354	$f_{RF} \pm 100$ kHz	N_{PH}	—	-92	-86	dBc/Hz	(1)
355	$f_{RF} \pm 1$ MHz	N_{PH}	—	-84	-82	dBc/Hz	(1)
	Phase Noise (434 MHz)						
356	$f_{RF} \pm 10$ kHz	N_{PH}	—	-84	-76	dBc/Hz	(1)
357	$f_{RF} \pm 100$ kHz	N_{PH}	—	-89	-83	dBc/Hz	(1)
358	$f_{RF} \pm 1$ MHz	N_{PH}	—	-82	-80	dBc/Hz	(1)
	Spurious Noise (315 and 434 MHz bands)						
359	$f_{RF} \pm f_{REF}$	N_{SPUR}	—	-45	-40	dBc	(1)
	Occupied Bandwidth (Korea, MIC 2007-63)						
360	For FSK up to ± 45 kHz and 9600 bit/sec and for OOK up to 9600 bit/sec Analyzed setup: RBW = VBW up to 10 kHz, Span up to 1.25 MHz, and MaxHold	OBW_K	—	—	180	kHz	(1)
	Noise for BOOST = 1						
	Phase Noise (315 MHz)						
361	$f_{RF} \pm 10$ kHz	N_{PH}	—	-75	-67	dBc/Hz	(1)
362	$f_{RF} \pm 100$ kHz	N_{PH}	—	-80	-76	dBc/Hz	(1)
363	$f_{RF} \pm 1$ MHz	N_{PH}	—	-95	-93	dBc/Hz	(1)
	Spurious Noise (315 MHz bands)						
364	$f_{RF} \pm f_{REF}$	N_{SPUR}	—	-45	-40	dBc	(1)
	Occupied Bandwidth (Japan, ARIB STD-T93)						
365	For FSK up to ±45 kHz and 9600 bit/sec	OBW_J	—	—	400	kHz	(1)
366	For OOK up to 9600 bit/sec Analyzed setup: RBW = VBW up to 30 kHz, Span up to 3.5 MHz, and MaxHold	OBW_J	—	—	600	kHz	(1)

Note: Refer to page 173 for description of notes.

17.17 Power consumption RF transmissions

$1.8 \le V_{DD} \le 3.6$, $T_A = 25°C$ unless otherwise specified.

Table 178.

#	Characteristic	Symbol	Min	Typ	Max	Units	
	RF Supply Transmission Current						
	V_{DD} = 3.0 V, PWR[4:0] set for nominal 5 dBm						
	315 MHZ Carrier Frequency, BOOST = 0						
367	Data 1, FSK or OOK	I_{DD}	—	6.0	7.0	mA	(3)
368	Interframe period, IFPD = 0	I_{DD}	—	0.65	0.8	mA	(1)
369	Interframe period, IFPD = 1	I_{DD}	—	15	26	µA	(1)
370	Power delta for BOOST = 1	ΔI_{DD}	—	—	0.55	mA	(3)
	434 MHZ Carrier Frequency, BOOST = 0						
371	Data 1, FSK or OOK	I_{DD}	—	6.6	7.6	mA	(3)
372	Interframe period, IFPD = 0	I_{DD}	—	0.65	0.8	mA	(1)
373	Interframe period, IFPD = 1	I_{DD}	—	15	26	µA	(1)
	RF Oscillator Frequency Accuracy, XCO						
374	excluding external crystal and component variations	f_{XCO}	-30	—	+30	ppm	(1)

Note: Refer to page 173 for description of notes.

Using the TPMS_RF_DYNAMIC_POWER firmware routine (see Section 14) allows adjusting power step in order to compensate variations of output power versus temperature and voltage. This routine will be associated to a part to part trimming that initially adjusts the power step to compensate for process variations.

Both these trim and look-up table allow to guarantee by characterization the typical values of power consumption as presented below (average values among 100 parts plus improvements prediction from design).

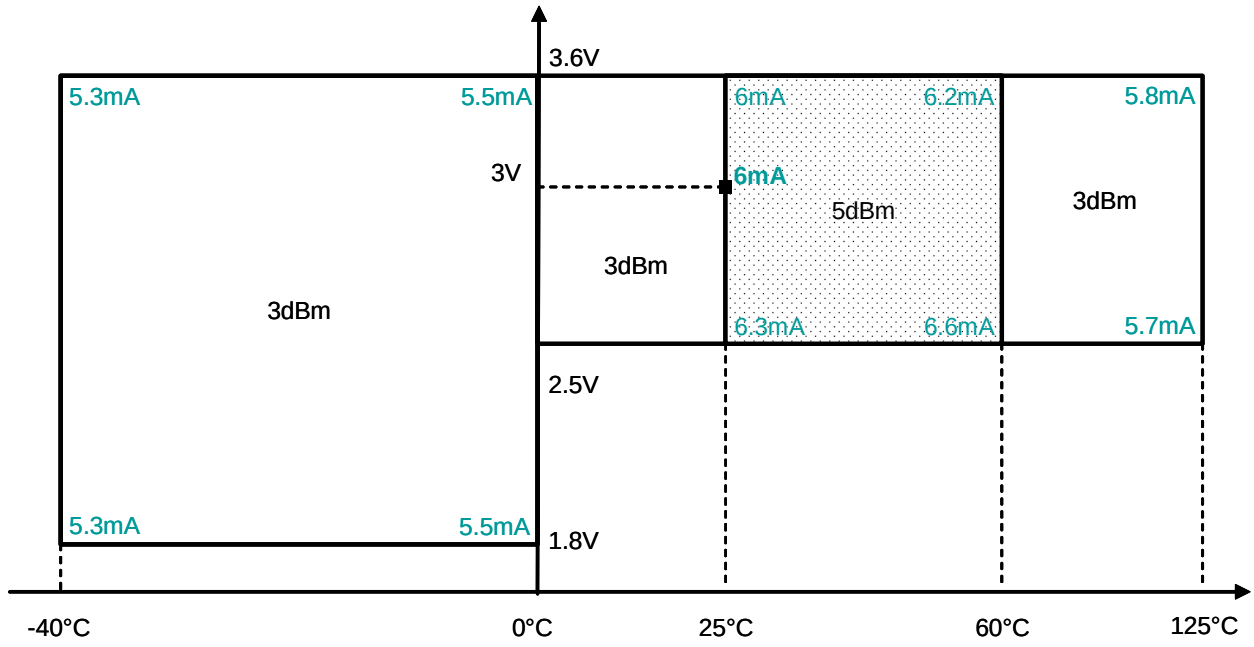


Figure 54. Dynamic power adjustment - 315 MHz

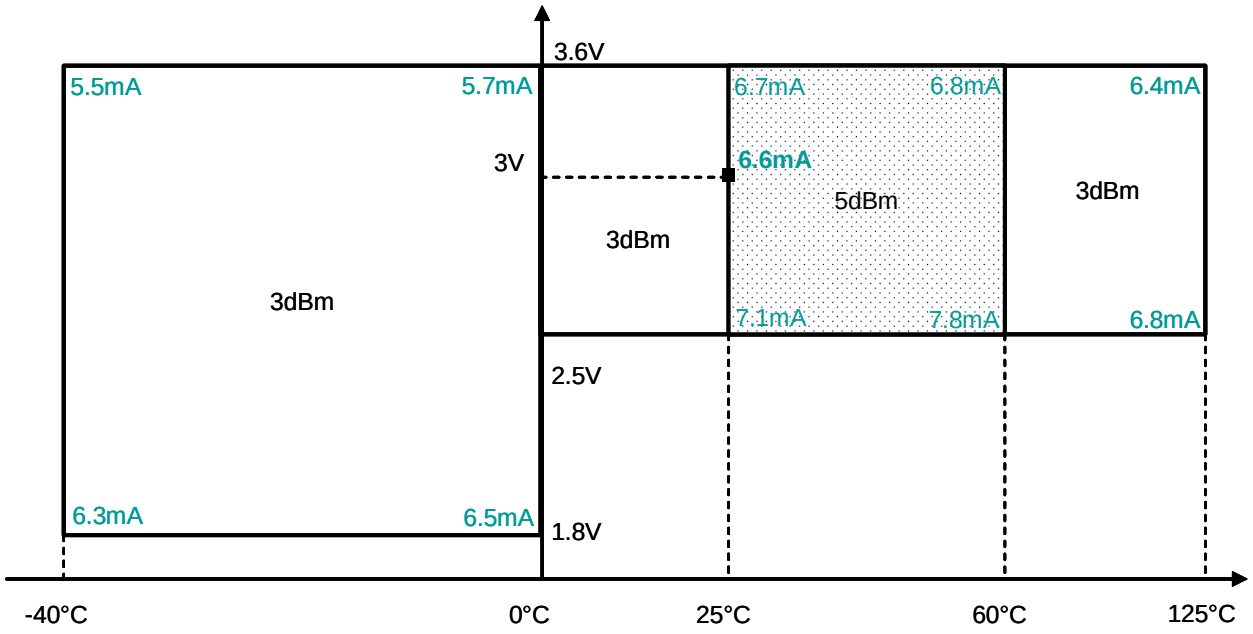


Figure 55. Dynamic power adjustment - 434 MHz

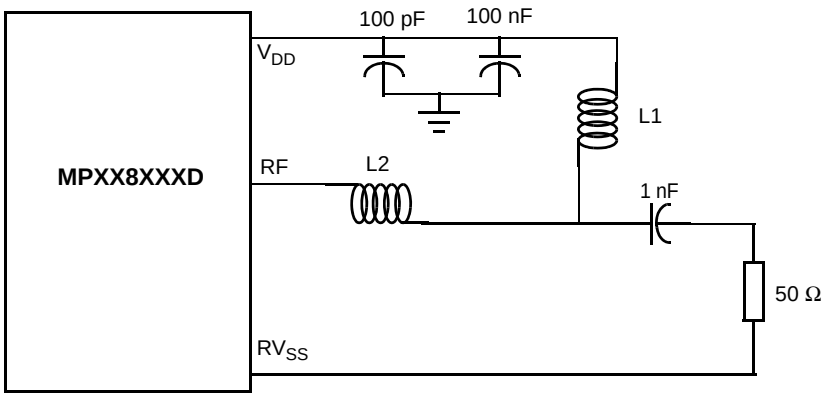


Figure 56. RF output power measurement configuration

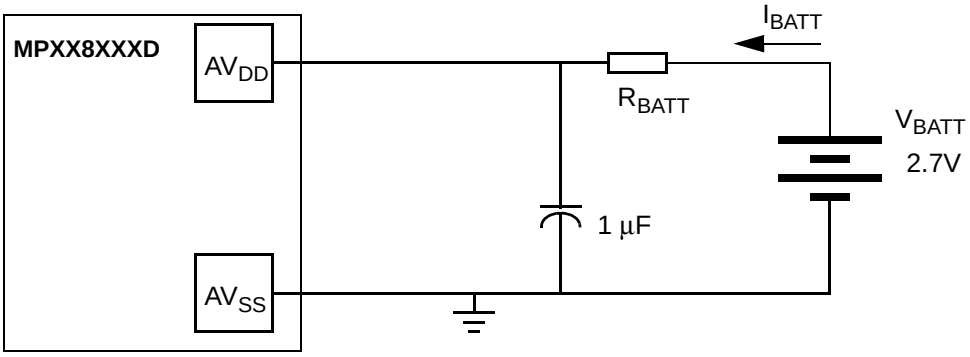


Figure 57. Battery performance test configuration

18 Mechanical Specifications

18.1 Maximum ratings

Maximum ratings are the extreme limits to which the device can be exposed without permanently damaging it. The device contains circuitry to protect the inputs against damage from high static voltages; however, do not apply voltages higher than those shown in the table below. Keep V_{IN} and V_{OUT} within the range $V_{SS} \le (V_{IN}$ or $V_{OUT}) \le V_{DD}$.

Table 179. Maximum ratings

#	Rating	Symbol	Value	Unit	
	Maximum pressure (absolute)				
375	Continuous	p_{max}	1400	kPa	(1)
376	Pulsed, 5 seconds, 25°C	p_{max}	2000	kPa	(1)
	Centrifugal force effects (Z-axis)				
377	Sustained acceleration (Z-axis)	g_{CENT}	2500	g	(1)
378	Powered shock (peak, 0.1 msec, half-sine, 6 -axis)	g_{shock}	6000	g	(1)
379	Drop test (onto concrete, unpowered)	h_{DROP}	1.2	m	(1)
	Pressure sensor resonance				
380	Resonant frequency	f_{P0}	5.5	MHz	(4)
381	Damping ratio	Q_P	1		(4)
	Optional Z-axis accelerometer sensor resonance				
382	Resonant frequency (no-peak, over-damped)	f_{Z0}	9	kHz	(4)
383	Damping ratio	Q_Z	5		(4)
384	Maximum acceleration before limit stops are reached	g_{MAX}	800	g	(4)
	Optional X-axis accelerometer sensor resonance				
385	Resonant frequency (under-damped)	f_{X0}	12.5	kHz	(4)
386	Damping ratio	Q_X	5		(4)
387	Maximum acceleration before limit stops are reached	g_{MAX}	TBD	g	(4)
388	Package weight	w	0.45	gm	(1)

Note: Refer to page 173 for description of notes.

18.2 Media compatibility

Media compatibility is based on media and test method described in Freescale specification 12MWS10081G, Media Test for TPMS MCM Automotive Pressure Sensors. Consult factory for more details and specific requirements.

18.3 Mounting recommendations

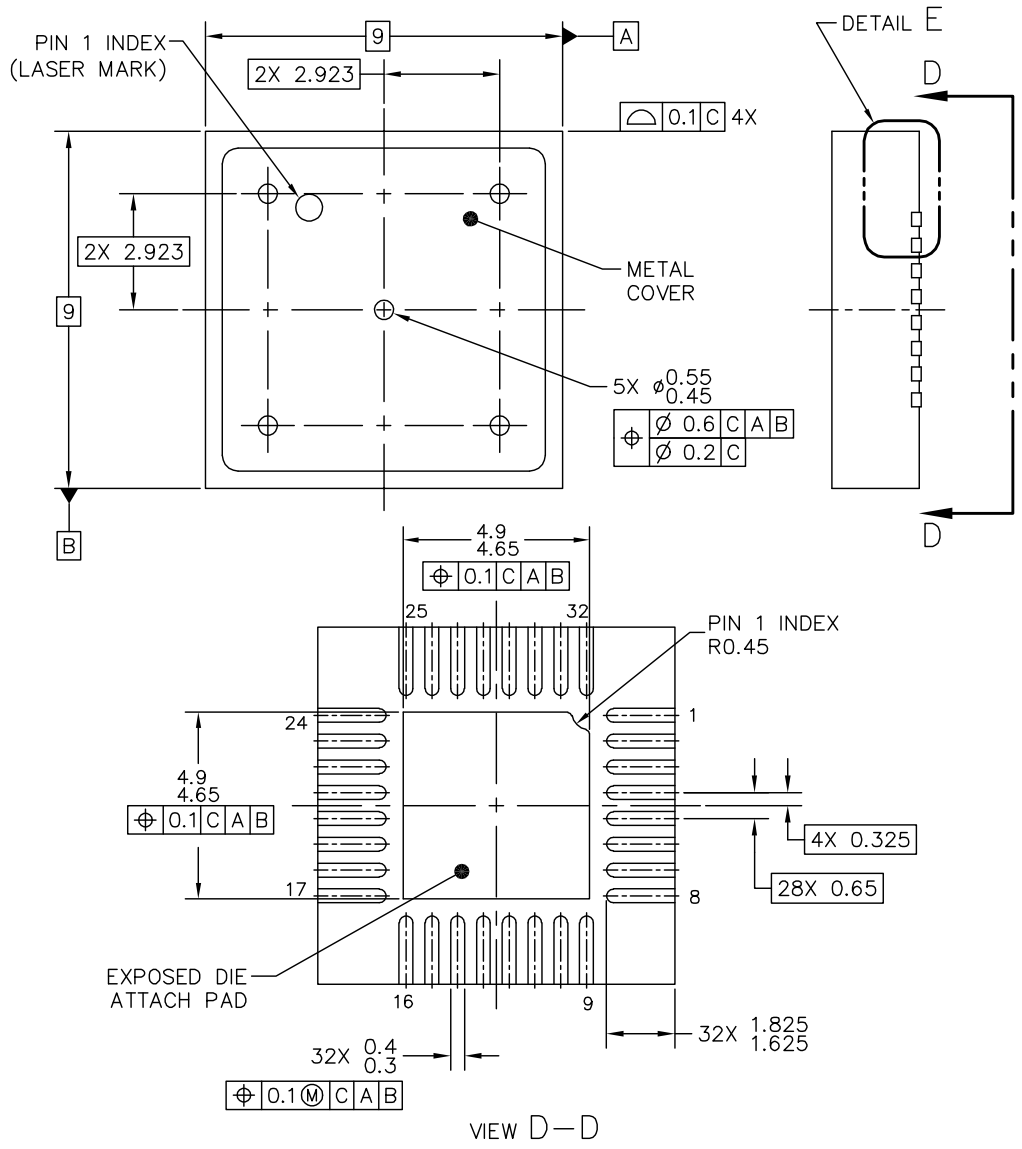
The package should be mounted with the pressure port pointing away from the axis of tire rotation so that centrifugal force will propel any contaminants out of the pressure port.

A plugged port will exhibit no change in pressure and can be cross checked in the user's software using the method described in Section 10.1.

Notes:

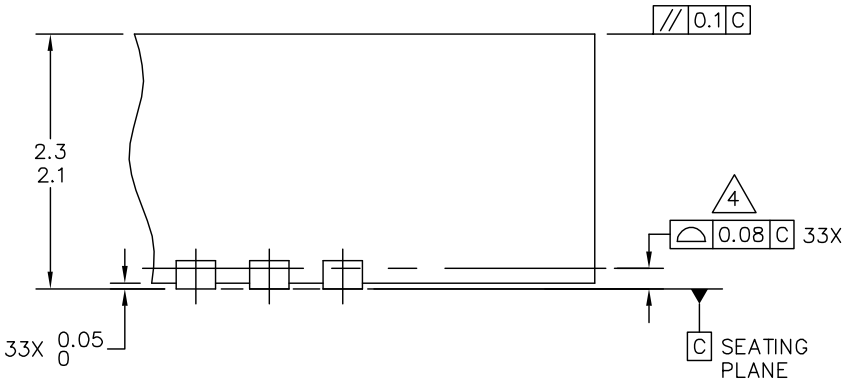
1. Verified by characterization, not tested in production.
2. Parameters tested 100% at final test.
3. Parameters tested 100% at unit probe.
4. For information only, may be determined by simulation.
5. Total of 3 hours over the life of the device.
6. Fully compensated pressure reading using TPMS_COMP_PRESSURE routine with single reading and 500 Hz low-pass filter ON.
7. Measurement times for one complete compensated reading; and times dependent on clock tolerances.
8. Peak currents measured as the current over 10 MCU bus cycles immediately after the ADC wakes the MCU using the external network shown in Figure 57 with R_{BATT} equal to zero ohm resistance.
9. Fully compensated acceleration reading using TPMS_COMP_ACCELERATION_X or TPMS_COMP_ACCELERATION_Z routine with single reading and 500 Hz low-pass filter ON.
10. Total range of variation over 30 consecutive measurements, using compensated output format.
11. Temperature error for MCU or RFM powered up at less the 10% duty-cycle
12. Package mounted with pressure port facing radially outward from axis of rotation.
13. Temperature shutdown points trimmed at final test.
14. Suggested crystal is NDK NX3225SA, 26.000 MHz.
15. Low voltage detection and warning thresholds defined for voltage change rates less than 20 mV/µs.
16. Response time to V_{DD} of more than 100 mV below the minimum V_{DD} falling threshold.
17. Crystal oscillator margin is the value of the total series resistance including the XTAL ESR, that can be applied before the XCO does not start up. This definition does not define any specific start up time.
18. Accuracy of the RF data rate when using the data buffer is dependent on the accuracy of the external crystal used.
19. LFR detection is defined as receiving 10 consecutive messages/pulses out of 10. LFR no detection is defined as not receiving any of 10 consecutive messages/pulses. In carrier detect mode the applied input pulse is at least 2X the LFCDTM selected. In all cases the envelop of the input waveform must have a RC time constant less than 15.3 µsec.
20. Using latest firmware available.
21. Actual final test value degraded by losses in the tester. Correlation study done as characterization to infer actual value.
22. Power consumption values refer to the firmware data flow in Figure 21. The BASIC Compensated value includes just the raw measurement and compensation routine for that parameter. Other raw readings needed for full compensation will be pulled from the UUMA. The FULL Compensated value includes taking all required raw readings and using the compensation routine for that parameter.

19 Package Dimensions



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: QFN, THERMALLY ENHANCED, 9 X 9 X 2.2 PKG, 0.65 PITCH, 32 TERMINAL	DOCUMENT NO: 98ASA10842D	REV: O
	CASE NUMBER: 2038–06	25 OCT 2011
	STANDARD: NON–JEDEC	

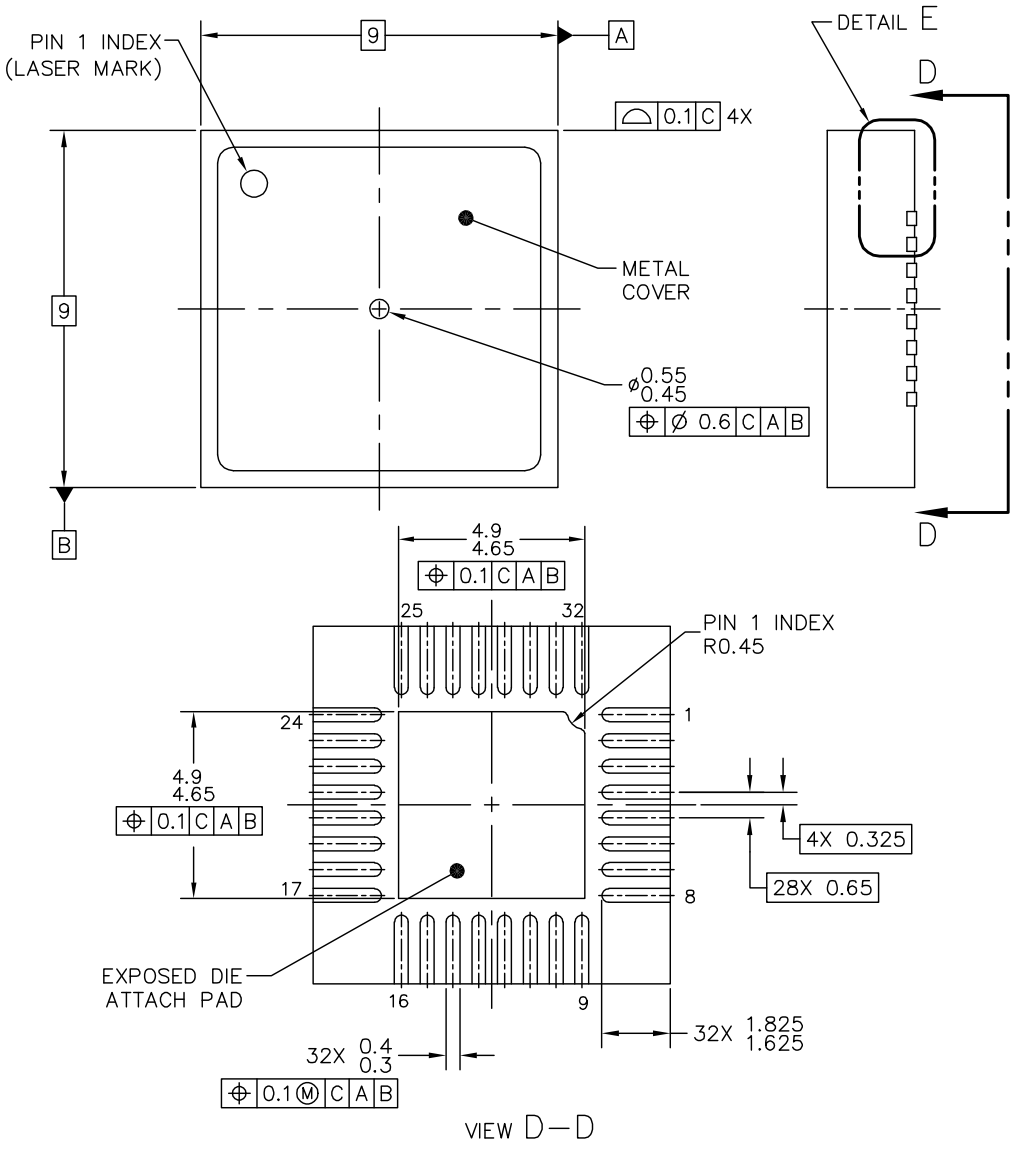
CASE 2038-06
ISSUE 0
32-TERMINAL QFN



DETAIL E
VIEW ROTATED 90° CW

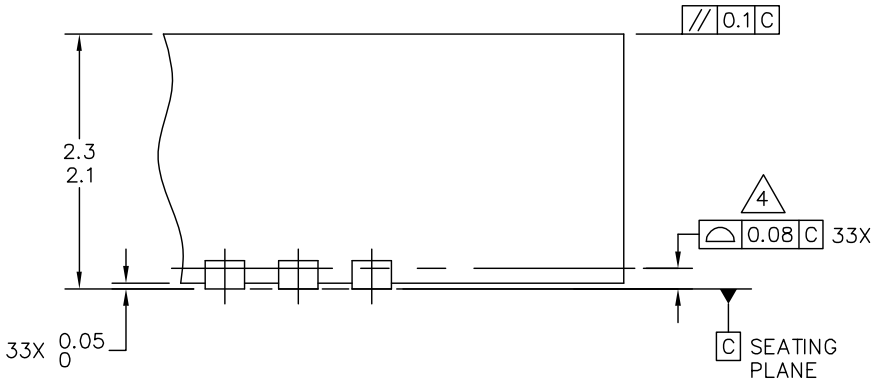
© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: QFN, THERMALLY ENHANCED, 9 X 9 X 2.2 PKG, 0.65 PITCH, 32 TERMINAL	DOCUMENT NO: 98ASA10842D	REV: O
	CASE NUMBER: 2038–06	25 OCT 2011
	STANDARD: NON–JEDEC	

CASE 2038-06
ISSUE 0
32-TERMINAL QFN



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: QFN, THERMALLY ENHANCED, 9 X 9 X 2.2 PKG, 0.65 PITCH, 32 TERMINAL	DOCUMENT NO: 98ASA00434D	REV: A
	CASE NUMBER: 2265–02	13 JAN 2012
	STANDARD: NON–JEDEC	

CASE 2265-02
ISSUE A
32-TERMINAL QFN



DETAIL E
VIEW ROTATED 90° CW

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: QFN, THERMALLY ENHANCED, 9 X 9 X 2.2 PKG, 0.65 PITCH, 32 TERMINAL		DOCUMENT NO: 98ASA00434D	REV: A
		CASE NUMBER: 2265–02	13 JAN 2012
		STANDARD: NON–JEDEC	

CASE 2265-02
ISSUE A
32-TERMINAL QFN

NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M–1994.
3. THIS IS NON JEDEC REGISTERED PACKAGE.
4. COPLANARITY APPLIES TO LEADS AND DIE ATTACH PAD.
5. MIN METAL GAP SHOULD BE 0.2MM.

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: QFN, THERMALLY ENHANCED, 9 X 9 X 2.2 PKG, 0.65 PITCH, 32 TERMINAL	DOCUMENT NO: 98ASA00434D	REV: A
	CASE NUMBER: 2265–02	13 JAN 2012
	STANDARD: NON–JEDEC	

CASE 2265-02
ISSUE A
32-TERMINAL QFN

20 Revision history

Table 180. Revision history

Version Number	Revision Date	Description of Changes
1.0	04/2013	• Initial release.
1.1	05/2013	• Updated ordering information table. • Section 14.2.3: Table 149 and Table 150: Changed CODE1, Bit 7 and 6, from ES2 and ES1 to Reserved.

Appendix C Rapid Decompression Events

This appendix describes the resulting effects of a rapid decompression event (RDE) when applied to a TPMS device with existing gel protection over the device cavity.

C.1 RDE characteristic

A low modulus gel is used to fill the cavity in the MPXx85/86xxD TPMS products to provide media protection against corrosive elements that may otherwise cause damage to metal bond wires and IC conductors. By nature, these gels are permeable to gases. Under pressure, gases may migrate through the gel and fill tiny lead-frame “microcavities”, formed during construction between the metal and plastic interfaces. These microcavities are too small to allow the gel to fill, and the pressure in these microcavities will eventually equalize to the applied pressure on the gel surface.

If the applied pressure should quickly decompress (referred to as a rapid decompression event - RDE), the gas contained in the microcavities will try to escape to the surface of the gel and therefore may form bubbles within the gel. If the formation or migration of these bubbles pass over the semiconductor transducer surfaces or bond wires, they may present an error until they dissipate. The effect of bubbles is variable and will not cause transducer shifts in all devices. Due to the time required for bubbles to form and/or migrate, a possible shift may not appear during the initial drop in pressure. During the initial decompression, the transducers will remain within their specified accuracy range until the applied pressure begins to approach the atmospheric level of 100 kPa.

Bubble formation does not occur unless the applied pressure has had a long enough time (dwell) to allow complete permeation of air to the microcavities.

C.2 Factors affecting RDE

From the previous discussion an RDE can cause residual pressure and/or acceleration shift after the following sequence:

1. An applied pressure above ambient pressure.
2. A long enough dwell time at a high applied pressure.
3. A rapid loss of the applied pressure, then
4. A period of time for any bubbles to form, and
5. No recovery to the original applied pressure.

In some cases, the bubbles that may have formed from microcavities have been demonstrated to remain over time. In most cases, the bubbles will dissipate with either time or a return to high pressures.

C.3 Characterized RDE response

A large sample of 180 devices have been tested and any resulting sensor shifts recorded and the amount of time necessary for the sensor shift to dissipate for a rapid decompression from 500 to 100 kPa. These devices are first subjected to a solder reflow cycle and then groups of 30 each are tested at the temperatures of -40°C, 25°C, 45°C, 85°C, 100°C and 125°C. The devices are then subjected to 10 cycles of rapid decompression from 900 to 100 kPa and then retested at their respective temperatures again.

Another sample of parts was tested for various rates of decompression to determine the fastest rate that might trigger a sensor shift.

The resulting data show the operation region where the specified accuracy of the devices is unaffected by loss of pressure; and defines the worst-case magnitude of error when such an RDE does occur.

C.4 Specification deviation

The following sections describe the expected deviation from the standard accuracy specifications for pressure for two levels of RDE, 500-100 kPa and 900-100 kPa. These are assuming no use of the software work-around which forces 100 KPa output readings in some cases. The range of pressure drop accuracy is defined as:

$$\Delta P_{RANGE} = 100 \times \frac{P_0 - P_1}{P_0 - P_{AMB}} \qquad \textit{Eqn. 20}$$

where:

ΔP_{RANGE} is the range of accuracy from the starting pressure, in percent

P_0 is the starting pressure, in kPa absolute

P_1 is the last accurate pressure within the specification, in kPa absolute

P_{AMB} is the atmospheric pressure, in kPa absolute (typically 100 kPa)

Examples of a 50% and 75% ΔP_{RANGE} are shown in Figure 58.

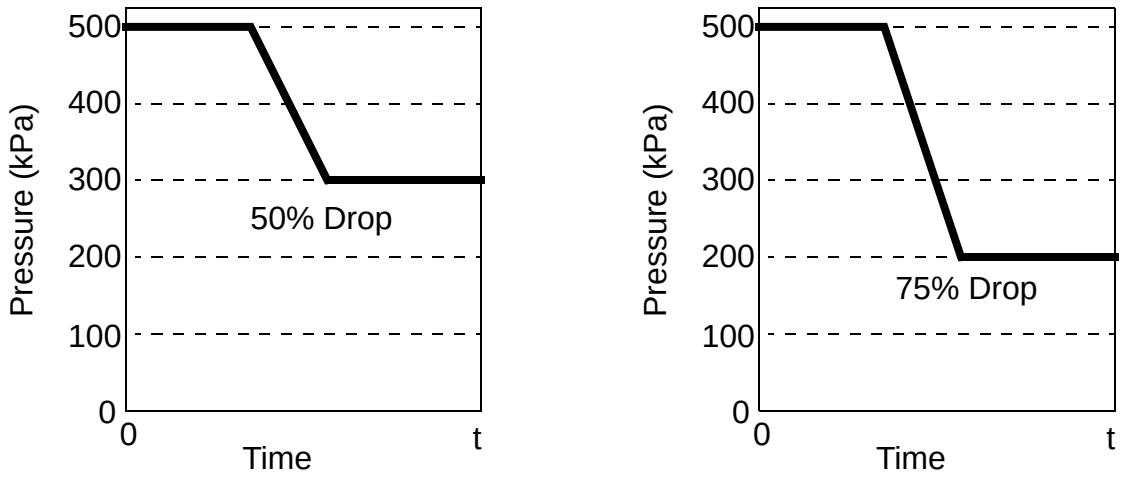


Figure 58. Examples of ΔP_{RANGE} percentage

For all cases of pressure change that is slower than 1 kPa/sec the pressure accuracy specifications meet those specified in the main product specification and Appendices A and B.

C.4.1 450-100 kPa RDE

The MPXx85/86xxD TPMS products in the 450 kPa pressure range will meet the standard accuracy limits within the white area of operation as shown in Figure 58. This accuracy is met as long as the maximum drop in pressure is less than 80% and the rate of pressure loss is slower than 1 kPa/sec as shown in Figure 59.

The RDE accuracy limits apply to the shaded area of operation as shown in Figure 58.

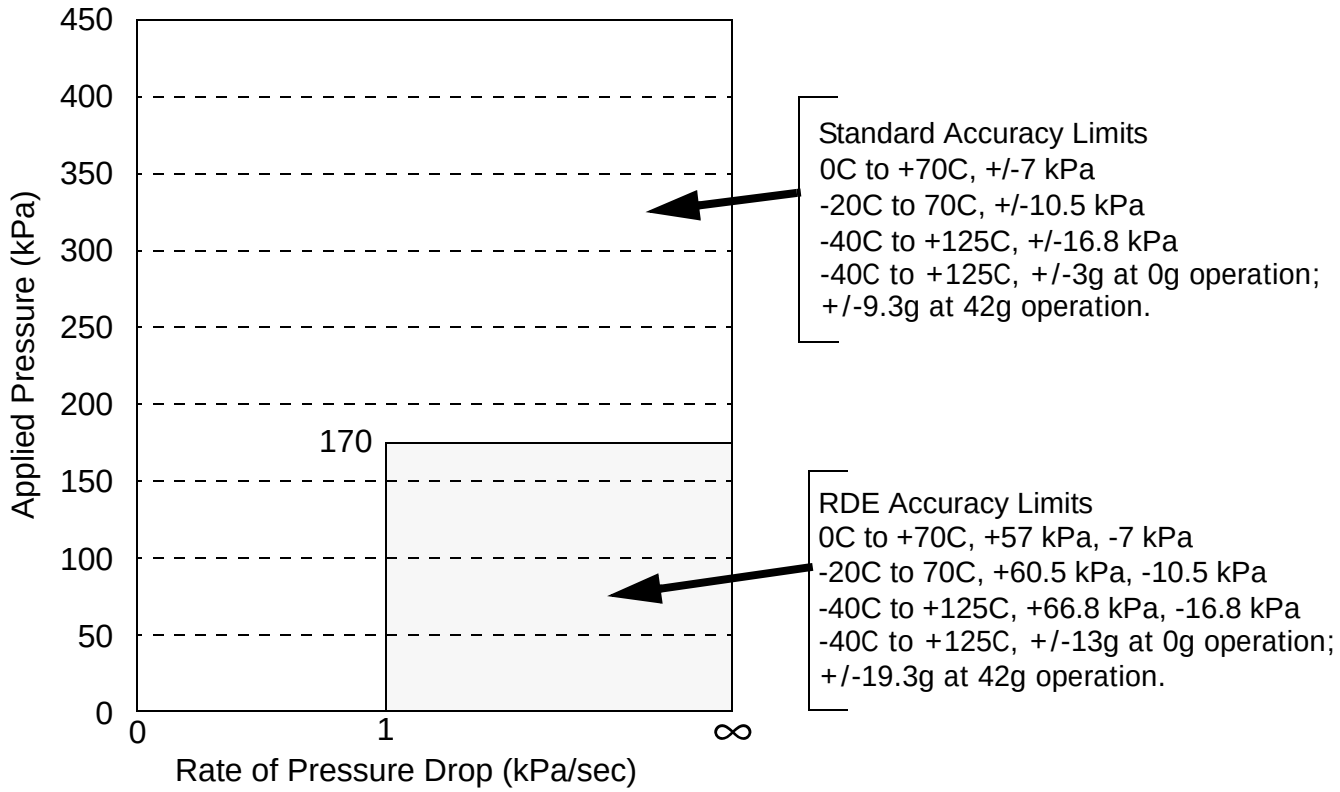


Figure 59. Accuracy area for 100-450 kPa range

These RDE accuracy limits will exist for up to 60 minutes after the RDE occurs. Restoration of the pressure above 50% of the original applied pressure will return the accuracy to standard specification. Acceleration accuracy may not return to standard specification, and may be assumed to operate to the RDE accuracy limits stated above.

An RDE that occurs following less than 5 minutes of dwell time at the highest pressure will not show any accuracy loss and will continue to meet the standard accuracy specification.

The above limits as specified over the 10-year lifetime of the device and reflect a process capability index (Cpk) of 2.00 or greater.

C.4.2 900-100 kPa RDE

The MPXx85/86xxD TPMS products in the 900 kPa pressure range will meet the standard accuracy limits within the white area of operation as shown in Figure 60. This accuracy is met as long as the maximum drop in pressure is less than 80% and the rate of pressure loss is slower than 1 kPa/sec as shown in Figure 60.

The RDE accuracy limits apply to the shaded area of operation as shown in Figure 60.

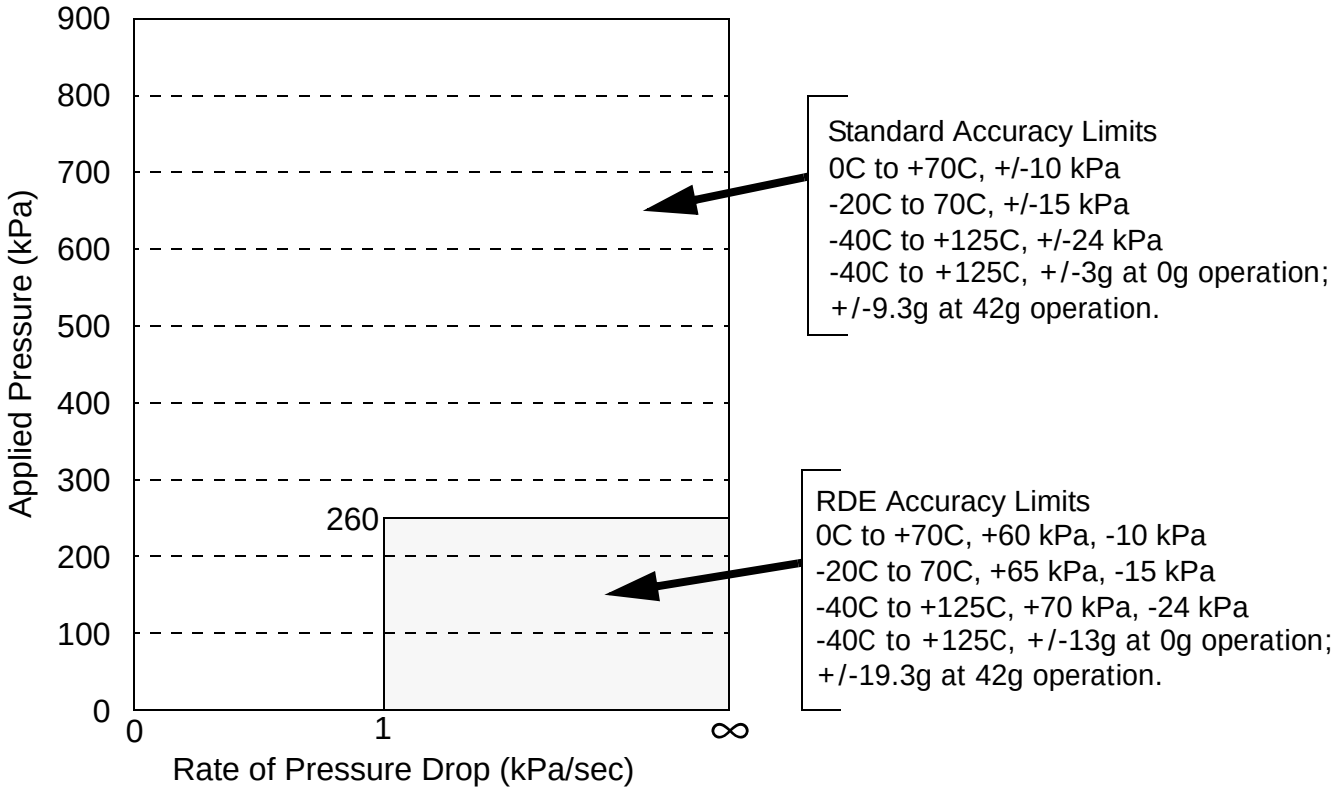


Figure 60. Accuracy area for 100-900 kPa range

These RDE accuracy limits will exist for up to 60 minutes after the RDE occurs. Restoration of the pressure above 50% of the original applied pressure will return the accuracy to standard specification. Acceleration accuracy may not return to standard specification, and may be assumed to operate to the RDE accuracy limits stated above.

An RDE that occurs following less than 5 minutes of dwell time at the highest pressure will not show any accuracy loss and will continue to meet the standard accuracy specification.

The above limits as specified over the 10-year lifetime of the device and reflect a process capability index (Cpk) of 2.00 or greater.

C.5 Software correction for RDE response

An optional software library routine is available from Freescale, Inc. that monitors the change in pressure based on the parameters specified in this appendix. This software routine will force the pressure reading to 100 kPa for those cases where it appears that the raw pressure reading is not following the expected pressure drop for an RDE. The user is advised to consider the acceleration RDE tolerances as potentially permanent after an RDE has been flagged by the software routine.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: MPXX8XXXD
Rev. 1.1
05/2013

freescale™