



Welcome to [E-XFL.COM](#)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Not For New Designs
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	I <sup>2</sup> C, IRSCI, SCI, SPI
Peripherals	LED, LVD, POR, PWM
Number of I/O	32
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-QFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908ap16acfber">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908ap16acfber</a>

## Table of Contents

4.3.2	Index Register . . . . .	56
4.3.3	Stack Pointer . . . . .	57
4.3.4	Program Counter . . . . .	57
4.3.5	Condition Code Register . . . . .	58
4.4	Arithmetic/Logic Unit (ALU) . . . . .	59
4.5	Low-Power Modes . . . . .	59
4.5.1	Wait Mode . . . . .	59
4.5.2	Stop Mode . . . . .	59
4.6	CPU During Break Interrupts . . . . .	60
4.7	Instruction Set Summary . . . . .	60
4.8	Opcode Map . . . . .	60

## Chapter 5 Oscillator (OSC)

5.1	Introduction . . . . .	71
5.2	Clock Selection . . . . .	71
5.2.1	CGM Reference Clock Selection . . . . .	72
5.2.2	TBM Reference Clock Selection . . . . .	73
5.3	Internal Oscillator . . . . .	74
5.4	RC Oscillator . . . . .	75
5.5	X-tal Oscillator . . . . .	75
5.6	I/O Signals . . . . .	76
5.6.1	Crystal Amplifier Input Pin (OSC1) . . . . .	76
5.6.2	Crystal Amplifier Output Pin (OSC2) . . . . .	76
5.6.3	Oscillator Enable Signal (SIMOSCEN) . . . . .	76
5.6.4	CGM Oscillator Clock (CGMXCLK) . . . . .	77
5.6.5	CGM Reference Clock (CGMRCLK) . . . . .	77
5.6.6	Oscillator Clock to Time Base Module (OSCCLK) . . . . .	77
5.7	Low Power Modes . . . . .	77
5.7.1	Wait Mode . . . . .	77
5.7.2	Stop Mode . . . . .	77
5.8	Oscillator During Break Mode . . . . .	77

## Chapter 6 Clock Generator Module (CGM)

6.1	Introduction . . . . .	79
6.2	Features . . . . .	79
6.3	Functional Description . . . . .	79
6.3.1	Oscillator Module . . . . .	81
6.3.2	Phase-Locked Loop Circuit (PLL) . . . . .	81
6.3.3	PLL Circuits . . . . .	81
6.3.4	Acquisition and Tracking Modes . . . . .	82
6.3.5	Manual and Automatic PLL Bandwidth Modes . . . . .	83
6.3.6	Programming the PLL . . . . .	83
6.3.7	Special Programming Exceptions . . . . .	86
6.3.8	Base Clock Selector Circuit . . . . .	86

**Chapter 23**  
**Mechanical Specifications**

23.1 Introduction . . . . . 309

23.2 48-Pin Low-Profile Quad Flat Pack (LQFP) . . . . . 310

23.3 44-Pin Quad Flat Pack (QFP) . . . . . 311

23.4 42-Pin Shrink Dual In-Line Package (SDIP) . . . . . 312

**Chapter 24**  
**Ordering Information**

24.1 Introduction . . . . . 313

24.2 MC Order Numbers . . . . . 313

12. After time,  $t_{rcv}$  (1  $\mu$ s), the memory can be accessed in read mode again.

This program sequence is repeated throughout the memory until all data is programmed.

**NOTE**

*The time between each FLASH address change (step 6 to step 6), or the time between the last FLASH addressed programmed to clearing the PGM bit (step 6 to step 9), must not exceed the maximum programming time,  $t_{prog\ max}$ .*

**NOTE**

*Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order shown, other unrelated operations may occur between the steps.*

### 2.5.6 FLASH Protection

Due to the ability of the on-board charge pump to erase and program the FLASH memory in the target application, provision is made to protect pages of memory from unintentional erase or program operations due to system malfunction. This protection is done by use of a FLASH block protect register (FLBPR). The FLBPR determines the range of the FLASH memory which is to be protected. The range of the protected area starts from a location defined by FLBPR and ends to the bottom of the FLASH memory (\$FFFF). When the memory is protected, the HVEN bit cannot be set in either erase or program operations.

**NOTE**

*The mask option register (\$FFCF) and the 48 bytes of user interrupt vectors (\$FFD0–\$FFFF) are always protected, regardless of the value in the FLASH block protect register. A mass erase is required to erase these locations.*

## Chapter 4

# Central Processor Unit (CPU)

### 4.1 Introduction

The M68HC08 CPU (central processor unit) is an enhanced and fully object-code-compatible version of the M68HC05 CPU. The *CPU08 Reference Manual* (Freescale document order number CPU08RM/AD) contains a description of the CPU instruction set, addressing modes, and architecture.

### 4.2 Features

- Object code fully upward-compatible with M68HC05 Family
- 16-bit stack pointer with stack manipulation instructions
- 16-Bit index register with x-register manipulation instructions
- 8-MHz CPU internal bus frequency
- 64-Kbyte program/data memory space
- 16 addressing modes
- Memory-to-memory data moves without using accumulator
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- Enhanced binary-coded decimal (BCD) data handling
- Modular architecture with expandable internal bus definition for extension of addressing range beyond 64 Kbytes
- Low-power stop and wait modes

### 6.3.5 Manual and Automatic PLL Bandwidth Modes

The PLL can change the bandwidth or operational mode of the loop filter manually or automatically. Automatic mode is recommended for most users.

In automatic bandwidth control mode ( $AUTO = 1$ ), the lock detector automatically switches between acquisition and tracking modes. Automatic bandwidth control mode also is used to determine when the VCO clock, CGMVCLK, is safe to use as the source for the base clock, CGMOUT. (See [6.5.2 PLL Bandwidth Control Register](#).) If PLL interrupts are enabled, the software can wait for a PLL interrupt request and then check the LOCK bit. If interrupts are disabled, software can poll the LOCK bit continuously (during PLL start-up, usually) or at periodic intervals. In either case, when the LOCK bit is set, the VCO clock is safe to use as the source for the base clock. (See [6.3.8 Base Clock Selector Circuit](#).) If the VCO is selected as the source for the base clock and the LOCK bit is clear, the PLL has suffered a severe noise hit and the software must take appropriate action, depending on the application. (See [6.6 Interrupts](#) for information and precautions on using interrupts.)

The following conditions apply when the PLL is in automatic bandwidth control mode:

- The  $\overline{ACQ}$  bit (See [6.5.2 PLL Bandwidth Control Register](#).) is a read-only indicator of the mode of the filter. (See [6.3.4 Acquisition and Tracking Modes](#).)
- The  $\overline{ACQ}$  bit is set when the VCO frequency is within a certain tolerance and is cleared when the VCO frequency is out of a certain tolerance. (See [6.8 Acquisition/Lock Time Specifications](#) for more information.)
- The LOCK bit is a read-only indicator of the locked state of the PLL.
- The LOCK bit is set when the VCO frequency is within a certain tolerance and is cleared when the VCO frequency is out of a certain tolerance. (See [6.8 Acquisition/Lock Time Specifications](#) for more information.)
- CPU interrupts can occur if enabled ( $PLLIE = 1$ ) when the PLL's lock condition changes, toggling the LOCK bit. (See [6.5.1 PLL Control Register](#).)

The PLL also may operate in manual mode ( $AUTO = 0$ ). Manual mode is used by systems that do not require an indicator of the lock condition for proper operation. Such systems typically operate well below  $f_{BUSMAX}$ .

The following conditions apply when in manual mode:

- $\overline{ACQ}$  is a writable control bit that controls the mode of the filter. Before turning on the PLL in manual mode, the  $\overline{ACQ}$  bit must be clear.
- Before entering tracking mode ( $\overline{ACQ} = 1$ ), software must wait a given time,  $t_{ACQ}$  (See [6.8 Acquisition/Lock Time Specifications](#).), after turning on the PLL by setting PLLON in the PLL control register (PCTL).
- Software must wait a given time,  $t_{AL}$ , after entering tracking mode before selecting the PLL as the clock source to CGMOUT ( $BCS = 1$ ).
- The LOCK bit is disabled.
- CPU interrupts from the CGM are disabled.

### 6.3.6 Programming the PLL

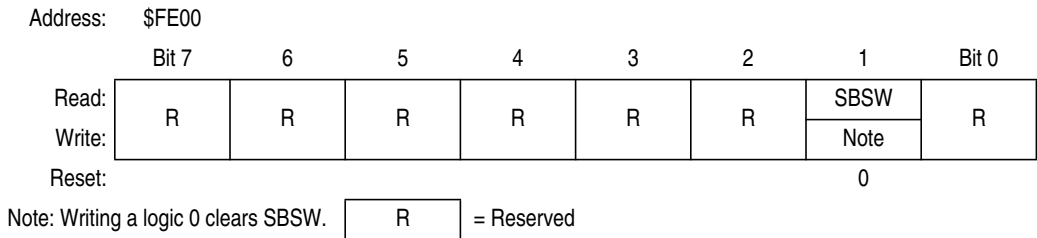
The following procedure shows how to program the PLL.

#### NOTE

*The round function in the following equations means that the real number should be rounded to the nearest integer number.*

### 7.7.1 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from stop mode or wait mode.



**Figure 7-20. SIM Break Status Register (SBSR)**

#### SBSW — SIM Break Stop/Wait

SBSW can be read within the break state SWI routine. The user can modify the return address on the stack by subtracting one from it.

- 1 = Wait mode was exited by break interrupt.
- 0 = Wait mode was not exited by break interrupt.

### TBR[2:0] — Timebase Rate Selection

These read/write bits are used to select the rate of timebase interrupts as shown in [Table 10-1](#).

#### NOTE

*Do not change TBR[2:0] bits while the timebase is enabled (TBON = 1).*

**Table 10-1. Timebase Rate Selection for OSCCLK = 4 MHz**

TBR2	TBR1	TBR0	Divider	Timebase Interrupt Rate <sup>(1)</sup>	
				Hz	ms
0	0	0	262144	15.259	65.536
0	0	1	131072	30.518	32.768
0	1	0	65536	61.035	16.384
0	1	1	32768	122.07	8.192
1	0	0	64	62,500	0.016
1	0	1	32	125,000	0.008
1	1	0	16	250,000	0.004
1	1	1	8	500,000	0.002

1. If user selects one of the shaded timebase settings, ensure that the operating bus frequency is fast enough to service the periodic interrupts.

### TACK — Timebase ACKnowledge

The TACK bit is a write-only bit and always reads as 0. Writing a logic 1 to this bit clears TBIF, the timebase interrupt flag bit. Writing a logic 0 to this bit has no effect.

1 = Clear timebase interrupt flag

0 = No effect

### TBIE — Timebase Interrupt Enabled

This read/write bit enables the timebase interrupt when the TBIF bit becomes set. Reset clears the TBIE bit.

1 = Timebase interrupt enabled

0 = Timebase interrupt disabled

### TBON — Timebase Enabled

This read/write bit enables the timebase. Timebase may be turned off to reduce power consumption when its function is not necessary. The counter can be initialized by clearing and then setting this bit. Reset clears the TBON bit.

1 = Timebase enabled

0 = Timebase disabled and the counter initialized to 0's



## 11.3 Pin Name Conventions

The generic names of the SCI I/O pins are:

- RxD (receive data)
- TxD (transmit data)

SCI I/O (input/output) lines are implemented by sharing parallel I/O port pins. The full name of an SCI input or output reflects the name of the shared port pin. Table 11-1 shows the full names and the generic names of the SCI I/O pins. The generic pin names appear in the text of this section.


**Table 11-1. Pin Name Conventions**

<b>Generic Pin Names:</b>	RxD	TxD
<b>Full Pin Names:</b>	PTB3/RxD	PTB2/TxD

### NOTE

*When the SCI is enabled, the TxD pin is an open-drain output and requires a pullup resistor to be connected for proper SCI operation.*

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$0013	SCI Control Register 1 (SCC1)	Read: LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Write:							
		Reset:	0	0	0	0	0	0	0
\$0014	SCI Control Register 2 (SCC2)	Read: SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Write:							
		Reset:	0	0	0	0	0	0	0
\$0015	SCI Control Register 3 (SCC3)	Read: R8	T8	DMARE	DMATE	ORIE	NEIE	FEIE	PEIE
		Write:							
		Reset:	U	U	0	0	0	0	0
\$0016	SCI Status Register 1 (SCS1)	Read: SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:							
		Reset:	1	1	0	0	0	0	0
\$0017	SCI Status Register 2 (SCS2)	Read:						BKF	RPF
		Write:							
		Reset:	0	0	0	0	0	0	0
\$0018	SCI Data Register (SCDR)	Read: R7	R6	R5	R4	R3	R2	R1	R0
		Write: T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset						
\$0019	SCI Baud Rate Register (SCBR)	Read: 0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:							
		Reset:	0	0	0	0	0	0	0

 = Unimplemented      R = Reserved      U = Unaffected

**Figure 11-1. SCI I/O Register Summary**

### 13.5.2 Transmission Format When CPHA = 0

Figure 13-4 shows an SPI transmission in which CPHA is logic 0. The figure should not be used as a replacement for data sheet parametric information.

Two waveforms are shown for SPSCCK: one for CPOL = 0 and another for CPOL = 1. The diagram may be interpreted as a master or slave timing diagram since the serial clock (SPSCCK), master in/slave out (MISO), and master out/slave in (MOSI) pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The  $\overline{SS}$  line is the slave select input to the slave. The slave SPI drives its MISO output only when its slave select input ( $\overline{SS}$ ) is at logic 0, so that only the selected slave drives to the master. The  $\overline{SS}$  pin of the master is not shown but is assumed to be inactive. The  $\overline{SS}$  pin of the master must be high or must be reconfigured as general-purpose I/O not affecting the SPI. (See 13.7.2 Mode Fault Error.) When CPHA = 0, the first SPSCCK edge is the MSB capture strobe. Therefore, the slave must begin driving its data before the first SPSCCK edge, and a falling edge on the  $\overline{SS}$  pin is used to start the slave data transmission. The slave's  $\overline{SS}$  pin must be toggled back to high and then low again between each byte transmitted as shown in Figure 13-5.

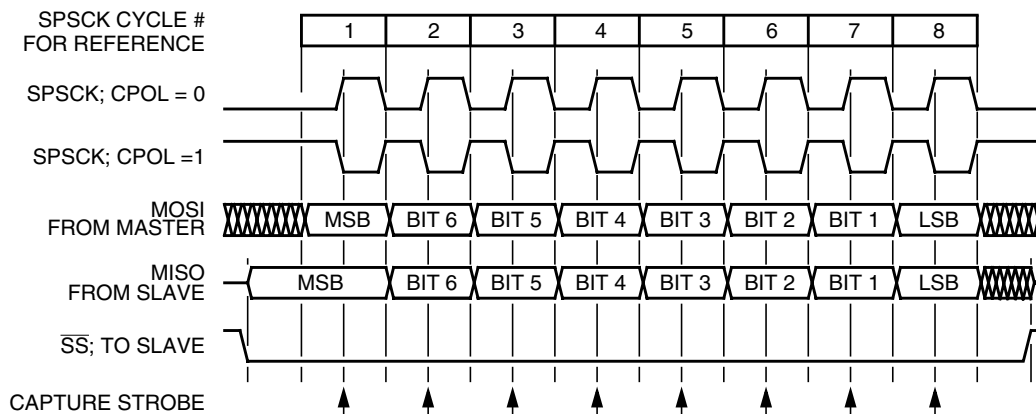


Figure 13-4. Transmission Format (CPHA = 0)

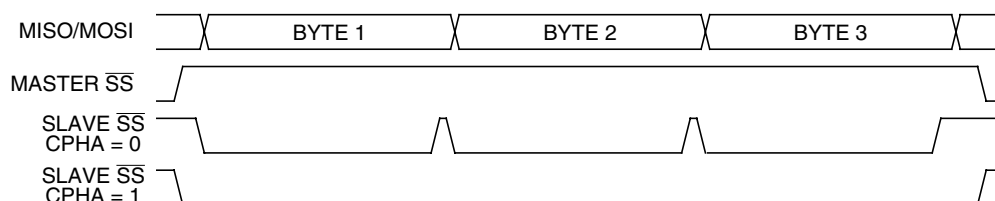


Figure 13-5. CPHA/ $\overline{SS}$  Timing

When CPHA = 0 for a slave, the falling edge of  $\overline{SS}$  indicates the beginning of the transmission. This causes the SPI to leave its idle state and begin driving the MISO pin with the MSB of its data. Once the transmission begins, no new data is allowed into the shift register from the transmit data register. Therefore, the SPI data register of the slave must be loaded with transmit data before the falling edge of  $\overline{SS}$ . Any data written after the falling edge is stored in the transmit data register and transferred to the shift register after the current transmission.

Since the SPTE bit cannot be cleared during a break with the BCFE bit cleared, a write to the transmit data register in break mode does not initiate a transmission nor is this data transferred into the shift register. Therefore, a write to the SPDR in break mode with the BCFE bit cleared has no effect.

### 13.12 I/O Signals

The SPI module has five I/O pins and shares four of them with a parallel I/O port. They are:

- MISO — Data received
- MOSI — Data transmitted
- SPSCCK — Serial clock
- $\overline{SS}$  — Slave select
- CGND — Clock ground (internally connected to  $V_{SS}$ )

The SPI has limited inter-integrated circuit ( $I^2C$ ) capability (requiring software support) as a master in a single-master environment. To communicate with  $I^2C$  peripherals, MOSI becomes an open-drain output when the SPWOM bit in the SPI control register is set. In  $I^2C$  communication, the MOSI and MISO pins are connected to a bidirectional pin from the  $I^2C$  peripheral and through a pullup resistor to  $V_{DD}$ .

#### 13.12.1 MISO (Master In/Slave Out)

MISO is one of the two SPI module pins that transmits serial data. In full duplex operation, the MISO pin of the master SPI module is connected to the MISO pin of the slave SPI module. The master SPI simultaneously receives data on its MISO pin and transmits data from its MOSI pin.

Slave output data on the MISO pin is enabled only when the SPI is configured as a slave. The SPI is configured as a slave when its SPMSTR bit is logic 0 and its  $\overline{SS}$  pin is at logic 0. To support a multiple-slave system, a logic 1 on the  $\overline{SS}$  pin puts the MISO pin in a high-impedance state.

When enabled, the SPI controls data direction of the MISO pin regardless of the state of the data direction register of the shared I/O port.

#### 13.12.2 MOSI (Master Out/Slave In)

MOSI is one of the two SPI module pins that transmits serial data. In full-duplex operation, the MOSI pin of the master SPI module is connected to the MOSI pin of the slave SPI module. The master SPI simultaneously transmits data from its MOSI pin and receives data on its MISO pin.

When enabled, the SPI controls data direction of the MOSI pin regardless of the state of the data direction register of the shared I/O port.

#### 13.12.3 SPSCCK (Serial Clock)

The serial clock synchronizes data transmission between master and slave devices. In a master MCU, the SPSCCK pin is the clock output. In a slave MCU, the SPSCCK pin is the clock input. In full-duplex operation, the master and slave MCUs exchange a byte of data in eight serial clock cycles.

When enabled, the SPI controls data direction of the SPSCCK pin regardless of the state of the data direction register of the shared I/O port.

## Serial Peripheral Interface Module (SPI)

### SPWOM — SPI Wired-OR Mode Bit

This read/write bit disables the pullup devices on pins SPSCCK, MOSI, and MISO so that those pins become open-drain outputs.

- 1 = Wired-OR SPSCCK, MOSI, and MISO pins
- 0 = Normal push-pull SPSCCK, MOSI, and MISO pins

### SPE — SPI Enable

This read/write bit enables the SPI module. Clearing SPE causes a partial reset of the SPI. (See [13.9 Resetting the SPI](#).) Reset clears the SPE bit.

- 1 = SPI module enabled
- 0 = SPI module disabled

### SPTIE— SPI Transmit Interrupt Enable

This read/write bit enables CPU interrupt requests generated by the SPTE bit. SPTE is set when a byte transfers from the transmit data register to the shift register. Reset clears the SPTIE bit.

- 1 = SPTE CPU interrupt requests enabled
- 0 = SPTE CPU interrupt requests disabled

## 13.13.2 SPI Status and Control Register

The SPI status and control register contains flags to signal these conditions:

- Receive data register full
- Failure to clear SPRF bit before next byte is received (overflow error)
- Inconsistent logic level on  $\overline{SS}$  pin (mode fault error)
- Transmit data register empty

The SPI status and control register also contains bits that perform these functions:

- Enable error interrupts
- Enable mode fault error detection
- Select master SPI baud rate

Address	\$0011							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SPRF	ERRIE	OVRF	MODF	SPTE	MODFEN	SPR1	SPR0
Write:								
Reset:	0	0	0	0	1	0	0	0
	<div style="display: inline-block; width: 20px; height: 10px; background-color: #cccccc; border: 1px solid black;"></div> = Unimplemented							

**Figure 13-14. SPI Status and Control Register (SPSCR)**


### SPRF — SPI Receiver Full Bit

This clearable, read-only flag is set each time a byte transfers from the shift register to the receive data register. SPRF generates a CPU interrupt request if the SPRIE bit in the SPI control register is set also. During an SPRF CPU interrupt, the CPU clears SPRF by reading the SPI status and control register with SPRF set and then reading the SPI data register. Reset clears the SPRF bit.

- 1 = Receive data register full
- 0 = Receive data register not full

## 14.6.2 MMIIC Control Register 1 (MMCR1)

Address:	\$0049							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MMEN	MMIEN	0	0	MMTXAK	REPSEN	MMCRCBYTE	0
Write:			MMCLRBB					
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

**Figure 14-5. MMIIC Control Register 1 (MMCR1)**

### MMEN — MMIIC Enable

This bit is set to enable the Multi-master IIC module. When MMEN = 0, module is disabled and all flags will restore to its power-on default states. Reset clears this bit.

1 = MMIIC module enabled

0 = MMIIC module disabled

### MMIEN — MMIIC Interrupt Enable

When this bit is set, the MMTXIF, MMRXIF, MMALIF, and MMNAKIF flags are enabled to generate an interrupt request to the CPU. When MMIEN is cleared, the these flags are prevented from generating an interrupt request. Reset clears this bit.

1 = MMTXIF, MMRXIF, MMALIF, and/or MMNAKIF bit set will generate interrupt request to CPU

0 = MMTXIF, MMRXIF, MMALIF, and/or MMNAKIF bit set will not generate interrupt request to CPU

### MMCLRBB — MMIIC Clear Busy Flag

Writing a logic 1 to this write-only bit clears the MMBB flag. MMCLRBB always reads as a logic 0.

Reset clears this bit.

1 = Clear MMBB flag

0 = No affect on MMBB flag

### MMTXAK — MMIIC Transmit Acknowledge Enable

This bit is set to disable the MMIIC from sending out an acknowledge signal to the bus at the 9th clock bit after receiving 8 data bits. When MMTXAK is cleared, an acknowledge signal will be sent at the 9th clock bit. Reset clears this bit.

1 = MMIIC does not send acknowledge signals at 9th clock bit

0 = MMIIC sends acknowledge signal at 9th clock bit

### REPSEN — Repeated Start Enable

This bit is set to enable repeated START signal to be generated when in master mode transfer (MMAST = 1). The REPSEN bit is cleared by hardware after the completion of repeated START signal or when the MMAST bit is cleared. Reset clears this bit.

1 = Repeated START signal will be generated if MMAST bit is set

0 = No repeated START signal will be generated

### MMCRCBYTE — MMIIC CRC Byte

In receive mode, this bit is set by software to indicate that the next receiving byte will be the packet error checking (PEC) data.

In master receive mode, after completion of CRC generation on the received PEC data, an acknowledge signal is sent if MMTXAK = 0; no acknowledge is sent If MMTXAK = 1.

In slave receive mode, no acknowledge signal is sent if a CRC error is detected on the received PEC data. If no CRC error is detected, an acknowledge signal is sent if MMTXAK = 0; no acknowledge is sent If MMTXAK = 1.

## 16.2.3 Port-A LED Control Register (LEDA)

The port-A LED control register (LEDA) controls the direct LED drive capability on PTA7–PTA0 pins. Each bit is individually configurable and requires that the data direction register, DDRA, bit be configured as an output.

Address:	\$000C							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	LEDA7	LEDA6	LEDA5	LEDA4	LEDA3	LEDA2	LEDA1	LEDA0
Write:								
Reset:	0	0	0	0	0	0	0	0

**Figure 16-5. Port A LED Control Register (LEDA)**

### LEDA[7:0] — Port A LED Drive Enable Bits

These read/write bits are software programmable to enable the direct LED drive on an output port pin.

- 1 = Corresponding port A pin is configured for direct LED drive, with 15mA current sinking capability
- 0 = Corresponding port A pin is configured for standard drive

## 16.3 Port B

Port B is an 8-bit special-function port that shares two of its pins with the multi-master IIC (MMIIC) module, two of its pins with SCI module, and four of its pins with two timer interface (TIM1 and TIM2) modules.

### NOTE

*PTB3–PTB0 are open-drain pins when configured as outputs regardless whether the pins are used as general purpose I/O pins, MMIIC pins, or SCI pins. Therefore, when configured as general purpose output pins, MMIIC pins, or SCI pins (the TxD pin), pullup resistors must be connected to these pins.*

### 16.3.1 Port B Data Register (PTB)

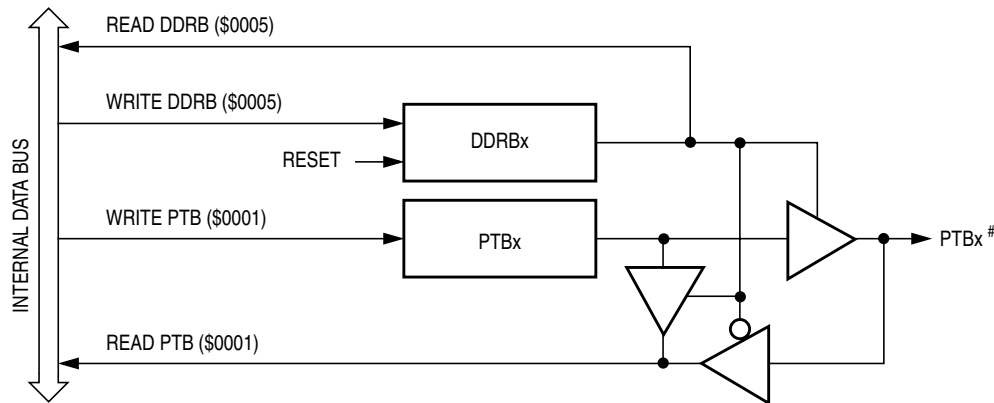
The port B data register contains a data latch for each of the eight port B pins.

Address:	\$0001							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Write:								
Reset:	Unaffected by reset							
Alternative Function:	T2CH1	T2CH0	T1CH1	T1CH0	RxD	TxD	SCL	SDA

**Figure 16-6. Port B Data Register (PTB)**

### PTB[7:0] — Port B Data Bits

These read/write bits are software-programmable. Data direction of each port B pin is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.



# PTB3–PTB0 are open-drain pins when configured as outputs.  
PTB7–PTB4 have schmitt trigger inputs.

**Figure 16-8. Port B I/O Circuit**

When DDRBx is a logic 1, reading address \$0001 reads the PTBx data latch. When DDRBx is a logic 0, reading address \$0001 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit.

Table 16-3 summarizes the operation of the port B pins.

**Table 16-3. Port B Pin Functions**

DDRB Bit	PTB Bit	I/O Pin Mode	Accesses to DDRB	Accesses to PTB	
			Read/Write	Read	Write
0	X <sup>(1)</sup>	Input, Hi-Z <sup>(2)</sup>	DDRB[7:0]	Pin	PTB[7:0] <sup>(3)</sup>
1	X	Output	DDRB[7:0]	PTB[7:0]	PTB[7:0]

1. X = don't care.
2. Hi-Z = high impedance.
3. Writing affects data register, but does not affect input.

## 16.4 Port C

Port C is an 8-bit special-function port that shares one of its pins with the  $\overline{\text{IRQ2}}$ , four of its pins with the SPI module, and two of its pins with the IRSCI module.

### 16.4.1 Port C Data Register (PTC)

The port C data register contains a data latch for each of the eight port C pins.

Address:	\$0002							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTC7	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
Write:	PTC7	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
Reset:	Unaffected by reset							
Alternative Function:	SCRxD	SCTxD	SPSCK	SS	MOSI	MISO		IRQ2

**Figure 16-9. Port C Data Register (PTC)**

### IMASK1 — IRQ1 Interrupt Mask Bit

Writing a logic 1 to this read/write bit disables IRQ1 interrupt requests. Reset clears IMASK1.

- 1 =  $\overline{\text{IRQ1}}$  interrupt requests disabled
- 0 =  $\overline{\text{IRQ1}}$  interrupt requests enabled

### MODE1 — IRQ1 Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the  $\overline{\text{IRQ1}}$  pin. Reset clears MODE1.

- 1 =  $\overline{\text{IRQ1}}$  interrupt requests on falling edges and low levels
- 0 =  $\overline{\text{IRQ1}}$  interrupt requests on falling edges only

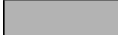
## 17.6.2 IRQ2 Status and Control Register

The IRQ2 status and control register (INTSCR2) controls and monitors operation of IRQ2. The INTSCR2 has the following functions:

- Enables/disables the internal pullup device on  $\overline{\text{IRQ2}}$  pin
- Shows the state of the IRQ2 flag
- Clears the IRQ2 latch
- Masks IRQ2 interrupt request
- Controls triggering sensitivity of the  $\overline{\text{IRQ2}}$  interrupt pin

Address: \$001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PUC0ENB	0	0	IRQ2F	0	IMASK2	MODE2
Write:						ACK2		
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

**Figure 17-5. IRQ2 Status and Control Register (INTSCR2)**

### PUC0ENB — $\overline{\text{IRQ2}}$ Pin Pullup Enable Bit.

Setting this bit to logic 1 disables the pullup on PTC0/ $\overline{\text{IRQ2}}$  pin.

Reset clears this bit.

- 1 =  $\overline{\text{IRQ2}}$  pin internal pullup is disabled
- 0 =  $\overline{\text{IRQ2}}$  pin internal pullup is enabled

### IRQ2F — IRQ2 Flag Bit

This read-only status bit is high when the IRQ2 interrupt is pending.

- 1 =  $\overline{\text{IRQ2}}$  interrupt pending
- 0 =  $\overline{\text{IRQ2}}$  interrupt not pending

### ACK2 — IRQ2 Interrupt Request Acknowledge Bit

Writing a logic 1 to this write-only bit clears the IRQ2 latch. ACK2 always reads as logic 0. Reset clears ACK2.

### IMASK2 — IRQ2 Interrupt Mask Bit

Writing a logic 1 to this read/write bit disables IRQ2 interrupt requests. Reset clears IMASK2.

- 1 = IRQ2 interrupt requests disabled
- 0 = IRQ2 interrupt requests enabled

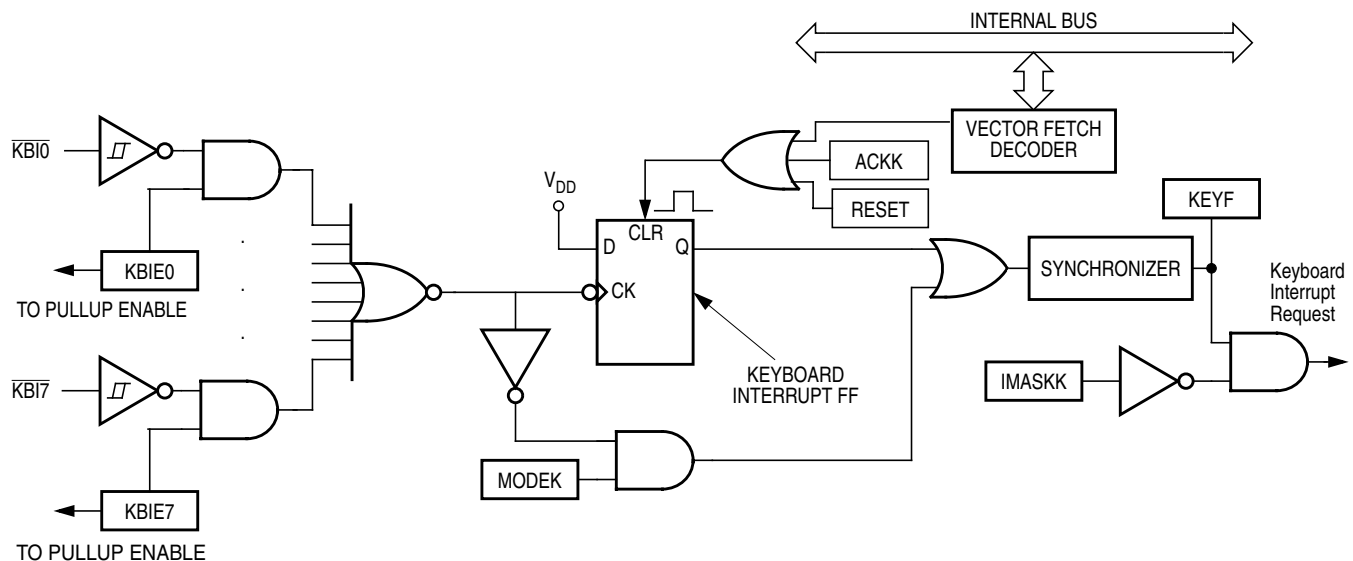
### MODE2 — IRQ2 Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the  $\overline{\text{IRQ2}}$  pin. Reset clears MODE2.

- 1 = IRQ2 interrupt requests on falling edges and low levels
- 0 =  $\overline{\text{IRQ2}}$  interrupt requests on falling edges only



## 18.4 Functional Description



**Figure 18-2. Keyboard Interrupt Block Diagram**

Writing to the KBIE7–KBIE0 bits in the keyboard interrupt enable register independently enables or disables each port D pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin in port D also enables its internal pull-up device. A logic 0 applied to an enabled keyboard interrupt pin latches a keyboard interrupt request.

A keyboard interrupt is latched when one or more keyboard pins goes low after all were high. The MODEK bit in the keyboard status and control register controls the triggering mode of the keyboard interrupt.

- If the keyboard interrupt is edge-sensitive only, a falling edge on a keyboard pin does not latch an interrupt request if another keyboard pin is already low. To prevent losing an interrupt request on one pin because another pin is still low, software can disable the latter pin while it is low.
- If the keyboard interrupt is falling edge- and low level-sensitive, an interrupt request is present as long as any keyboard pin is low.

If the MODEK bit is set, the keyboard interrupt pins are both falling edge- and low level-sensitive, and both of the following actions must occur to clear a keyboard interrupt request:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the interrupt request. Software may generate the interrupt acknowledge signal by writing a logic 1 to the ACKK bit in the keyboard status and control register KBSCR. The ACKK bit is useful in applications that poll the keyboard interrupt pins and require software to clear the keyboard interrupt request. Writing to the ACKK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACKK does not affect subsequent transitions on the keyboard interrupt pins. A falling edge that occurs after writing to the ACKK bit latches another interrupt request. If the keyboard interrupt mask bit, IMASKK, is clear, the CPU loads the program counter with the vector address at locations \$FFE0 and \$FFE1.
- Return of all enabled keyboard interrupt pins to logic 1 — As long as any enabled keyboard interrupt pin is at logic 0, the keyboard interrupt remains set.

The vector fetch or software clear and the return of all enabled keyboard interrupt pins to logic 1 may occur in any order.

### CAUTION

*A break address should be placed at the address of the instruction opcode. When software does not change the break address and clears the BRKA bit in the first break interrupt routine, the next break interrupt will not be generated after exiting the interrupt routine even when the internal address bus matches the value written in the break address registers.*

## 21.3.3 TIM1 and TIM2 During Break Interrupts

A break interrupt stops the timer counters.

## 21.3.4 COP During Break Interrupts

The COP is disabled during a break interrupt when  $V_{TST}$  is present on the  $\overline{RST}$  pin.

# 21.4 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

## 21.4.1 Wait Mode

If enabled, the break module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if SBSW is set. (see [Chapter 7 System Integration Module \(SIM\)](#)) Clear the BW bit by writing logic 0 to it.

# 21.5 Break Module Registers


These registers control and monitor operation of the break module:

- Break status and control register (BRKSCR)
- Break address register high (BRKH)
- Break address register low (BRKL)
- SIM break status register (SBSR)
- SIM break flag control register (SBFCR)

## 21.5.1 Break Status and Control Register

The break status and control register (BRKSCR) contains break module enable and status bits.

Address:	\$FE0E							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BRKE	BRKA	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

**Figure 21-3. Break Status and Control Register (BRKSCR)**

## Chapter 23

# Mechanical Specifications

### 23.1 Introduction

This section gives the dimensions for:

- 48-pin plastic low-profile quad flat pack (case #932)
- 44-pin plastic quad flat pack (case #824A)
- 42-pin shrink dual in-line package (case #858)

# 23.2 48-Pin Low-Profile Quad Flat Pack (LQFP)

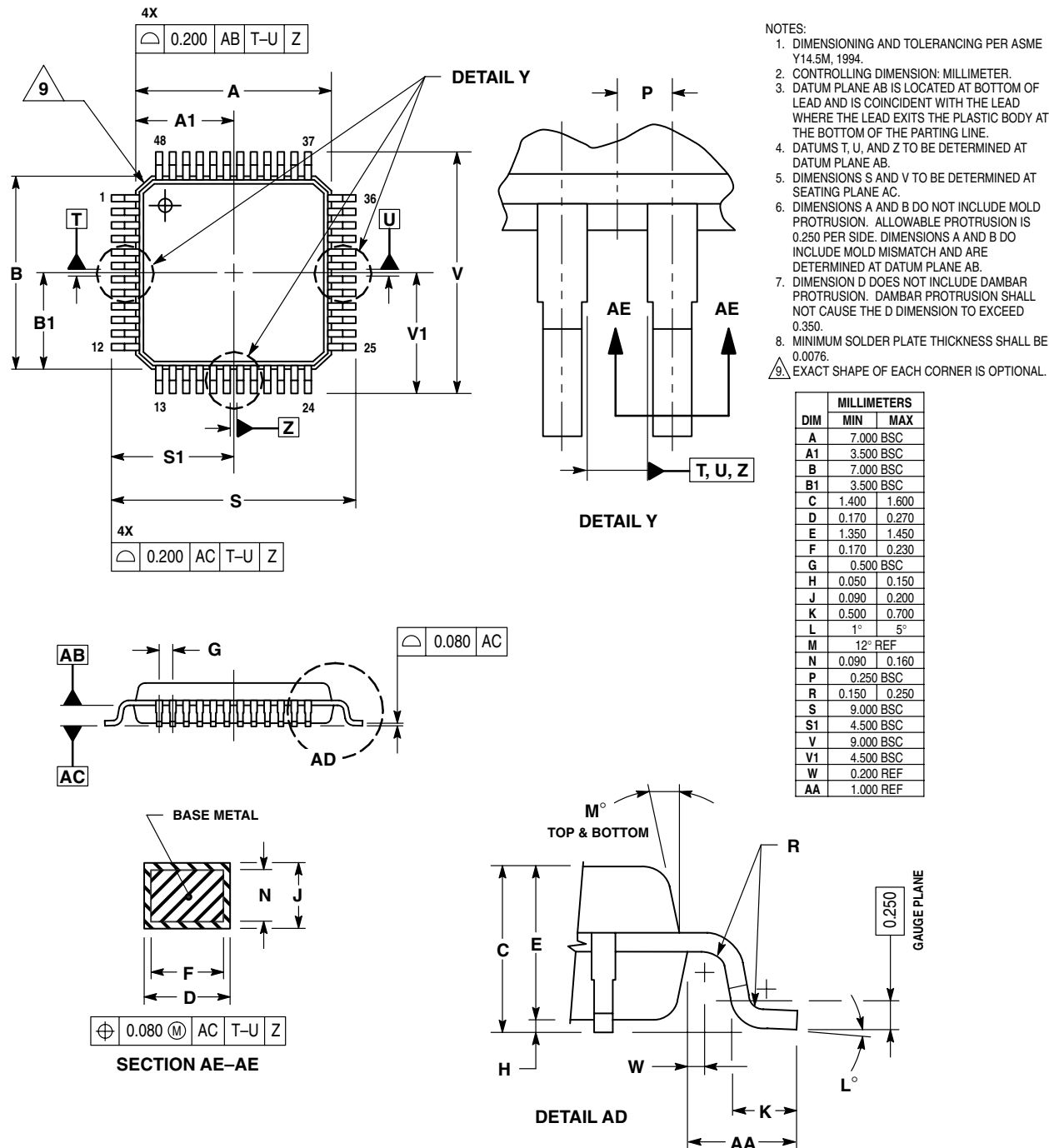


Figure 23-1. 48-Pin LQFP (Case #932)

