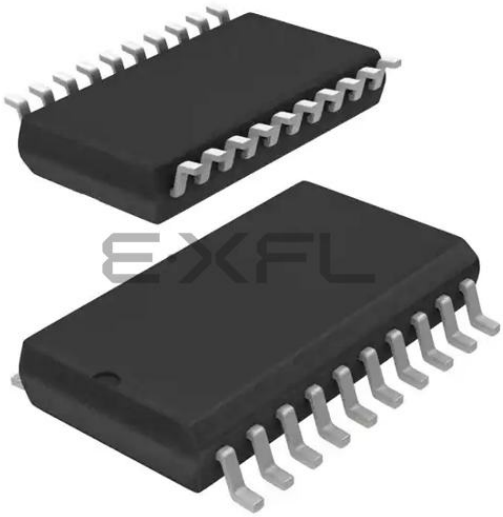


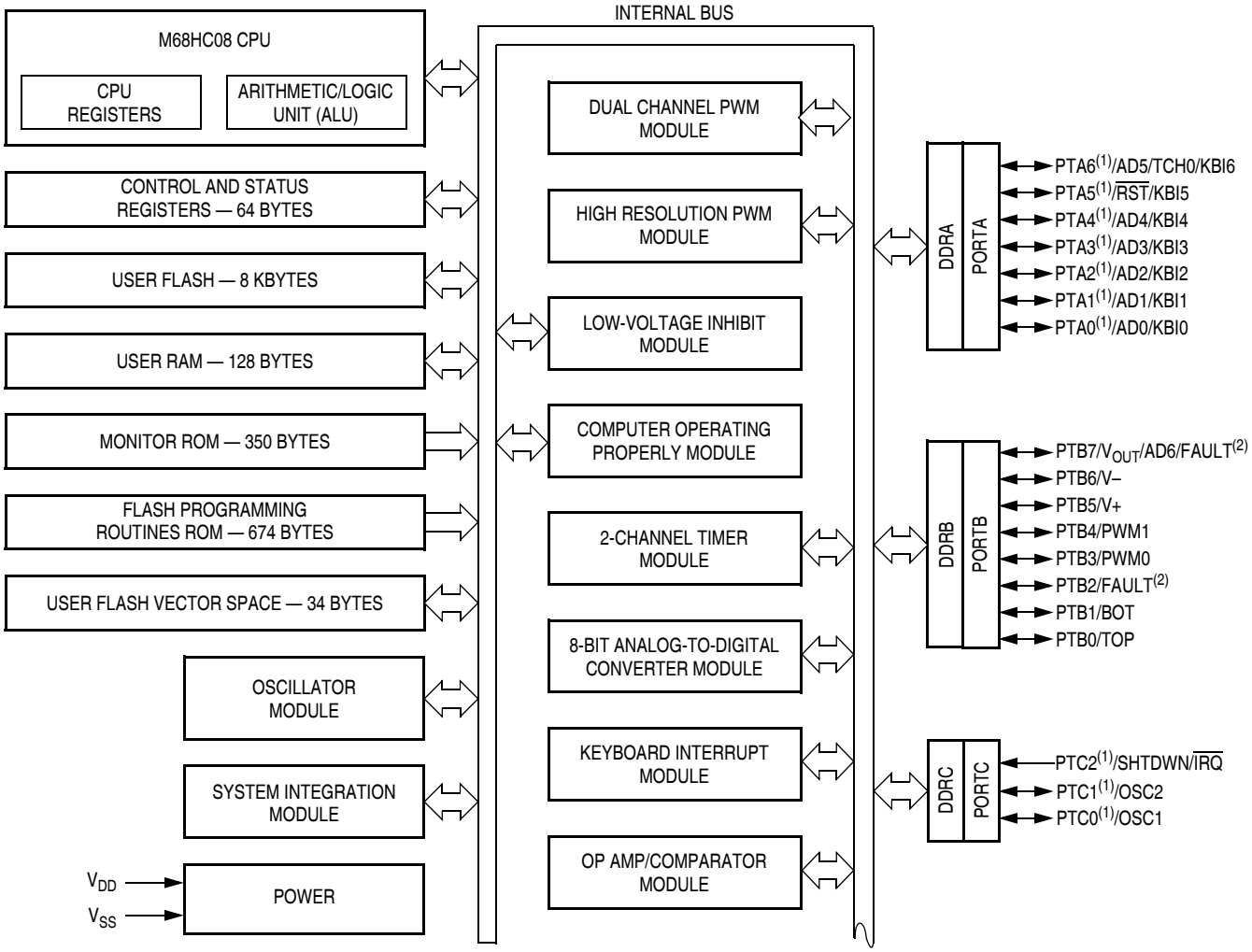
Welcome to E-XFL.COM

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	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LVR, POR, PWM
Number of I/O	18
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 7x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908lb8mdwer



Notes:
1. Pin contains integrated pullup device.
2. Fault function switchable between pins PTB2 and PTB7.

Figure 1-1. MCU Block Diagram

1.4 Pin Assignments

Figure 1-2 illustrates the pin assignments for the 20-pin SOIC package.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0006	Data Direction Register C (DDRC) See page 139.	Read:	0	0	0	0	0	0	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007 ↓ $000C	Unimplemented									
$000D	Port A Input Pullup Enable Register (PTAPUE) See page 136.	Read:		PTA6PUE	PTA5PUE	PTA4PUE	PTA3PUE	PTA2PUE	PTA1PUE	PTA0PUE
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000E	Port C Input Pullup Enable Register (PTCPUE) See page 140.	Read:	OSC2EN	0	0	0	0	PTCPUE2	PTCPUE1	PTCPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000F ↓ $0019	Unimplemented									
$001A	Keyboard Status and Control Register (INTKBSCR) See page 89.	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$001B	Keyboard Interrupt Enable Register (INTKBIER) See page 90.	Read:		KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001D	IRQ Status and Control Register (INTSCR) See page 84.	Read:	0	0	0	0	IRQF	0	IMASK	MODE
		Write:						ACK		
		Reset:	0	0	0	0	0	0	0	0
$001E	Configuration Register 2 (CONFIG2)(1) See page 60.	Read:	IRQPUD	IRQEN	R	OSCOPT1	OSCOPT0	0	0	RSTEN
		Write:								
		Reset:	0	0	0	0	0	0	0	0(2)

1. One-time writable register after each reset.
2. RSTEN reset to 0 by a power-on reset (POR) only.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001F	Configuration Register 1 (CONFIG1)(1) See page 61.	Read:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	0	SSREC	STOP	COPD
		Write:								
		Reset:	0	0	0	0	0	0	0	0

1. One-time writable register after reach reset.

= Unimplemented R = Reserved

Bold = Buffered U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 2 of 8)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0030 ↓ $0033	Reserved		Reserved							
$0034 ↓ $0035	Unimplemented									
$0036	Oscillator Status Register (OSCSTAT) See page 130.	Read:	R	R	R	R	R	R	ECGON	EGGST
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0037	Unimplemented									
$0038	Oscillator Trim Register (OSCTRIM) See page 131.	Read:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
		Write:								
		Reset:	1	0	0	0	0	0	0	0
$0039	Op Amp/Comparator Control Register (OACCR) See page 55.	Read:	OACM							OACE
		Write:								
		Reset:	0	U	U	U	U	U	U	0
$003A ↓ $003B	Unimplemented									
$003C	ADC Status and Control Register (ADSCR) See page 48.	Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
		Write:								
		Reset:	0	0	0	1	1	1	1	1
$003D	Unimplemented									
$003E	ADC Data Register (ADR) See page 50.	Read:	AD7	AD6	AD5	AD4	A3	AD2	AD1	AD0
		Write:								
		Reset:	Unaffected by reset							
$003F	ADC Clock Register (ADCLK) See page 50.	Read:	ADIV2	ADIV1	ADIV0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved

Bold = Buffered U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 8)

6.3.6 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the configuration register 1 (CONFIG1). See Chapter 5 Configuration Register (CONFIG).

6.3.7 COPRS (COP Rate Select)

The COPRS signal reflects the state of the COP rate select bit (COPRS) in the configuration register 1 (CONFIG1). See Chapter 5 Configuration Register (CONFIG).

6.4 COP Control Register

The COP control register (COPCTL) is located at address $FFFF and overlaps the reset vector. Writing any value to $FFFF clears the COP counter and starts a new timeout period. Reading location $FFFF returns the low byte of the reset vector.

Address: $FFFF

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	LOW BYTE OF RESET VECTOR							
Write:	CLEAR COP COUNTER							
Reset:	Unaffected by reset							

Figure 6-2. COP Control Register (COPCTL)

6.5 Interrupts

The COP does not generate CPU interrupt requests.

6.6 Monitor Mode

The COP is disabled in monitor mode when V_{TST} is present on the $\overline{IRQ}$ pin.

6.7 Low-Power Modes

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

6.7.1 Wait Mode

The COP remains active during wait mode. If COP is enabled, a reset will occur at COP timeout.

6.7.2 Stop Mode

Stop mode turns off the BUSCLKX4 input to the COP and clears the SIM counter. Service the COP immediately before entering or after exiting stop mode to ensure a full COP timeout period after entering or exiting stop mode.

To prevent inadvertently turning off the COP with a STOP instruction, a configuration option is available that disables the STOP instruction. When the STOP bit in the configuration register has the STOP instruction disabled, execution of a STOP instruction results in an illegal opcode reset.

Table 7-1. Instruction Set Summary (Sheet 7 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
WAIT	Enable Interrupts; Wait for Interrupt	I bit ← 0; Inhibit CPU clocking until interrupted	–	–	0	–	–	–	INH	8F		1

A	Accumulator
C	Carry/borrow bit
CCR	Condition code register
dd	Direct address of operand
dd rr	Direct address of operand and relative offset of branch instruction
DD	Direct to direct addressing mode
DIR	Direct addressing mode
DIX+	Direct to indexed with post increment addressing mode
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing
EXT	Extended addressing mode
ff	Offset byte in indexed, 8-bit offset addressing
H	Half-carry bit
H	Index register high byte
hh ll	High and low bytes of operand address in extended addressing
I	Interrupt mask
ii	Immediate operand byte
IMD	Immediate source to direct destination addressing mode
IMM	Immediate addressing mode
INH	Inherent addressing mode
IX	Indexed, no offset addressing mode
IX+	Indexed, no offset, post increment addressing mode
IX+D	Indexed with post increment to direct addressing mode
IX1	Indexed, 8-bit offset addressing mode
IX1+	Indexed, 8-bit offset, post increment addressing mode
IX2	Indexed, 16-bit offset addressing mode
M	Memory location
N	Negative bit
n	Any bit
opr	Operand (one or two bytes)
PC	Program counter
PCH	Program counter high byte
PCL	Program counter low byte
REL	Relative addressing mode
rel	Relative program counter offset byte
rr	Relative program counter offset byte
SP1	Stack pointer, 8-bit offset addressing mode
SP2	Stack pointer 16-bit offset addressing mode
SP	Stack pointer
U	Undefined
V	Overflow bit
X	Index register low byte
Z	Zero bit
&	Logical AND
|	Logical OR
⊕	Logical EXCLUSIVE OR
( )	Contents of
–( )	Negation (two's complement)
#	Immediate value
«	Sign extend
←	Loaded with
?	If
:	Concatenated with
↕	Set or cleared
—	Not affected

7.8 Opcode Map

See Table 7-2.

9.2 Features

Features include:

- Seven keyboard interrupt pins with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Hysteresis buffers
- Programmable edge-only or edge- and level- interrupt sensitivity
- Exit from low-power modes
- I/O (input/output) port bit(s) software configurable with pullup device(s) if configured as input port bit(s)

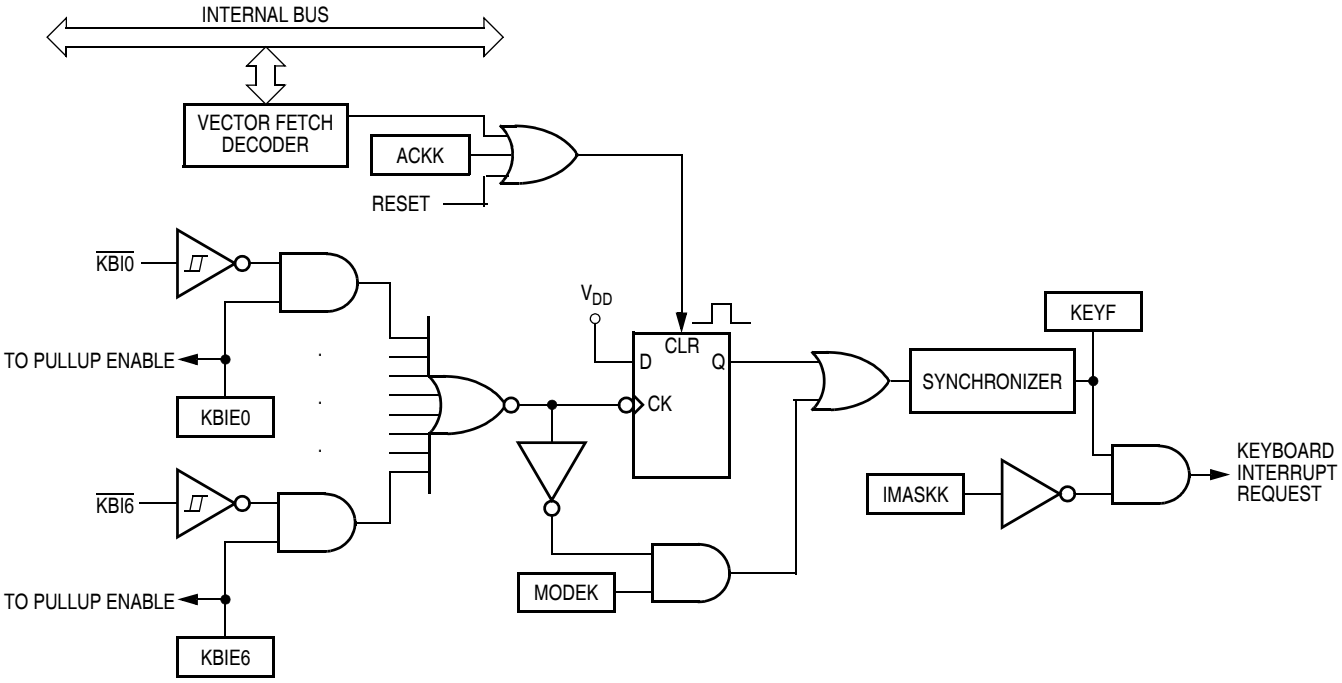


Figure 9-2. Keyboard Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001A	Keyboard Status and Control Register (INTKBSCR) See page 89.	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$001B	Keyboard Interrupt Enable Register (INTKBIER) See page 90.	Read:		KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 9-3. I/O Register Summary

9.4 Keyboard Initialization

When a keyboard interrupt pin is enabled, it takes time for the internal pullup to reach a logic 1. Therefore, a false interrupt can occur as soon as the pin is enabled.

To prevent a false interrupt on keyboard initialization:

1. Mask keyboard interrupts by setting the IMASKK bit in the keyboard status and control register.
2. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.
3. Write to the ACKK bit in the keyboard status and control register to clear any false interrupts.
4. Clear the IMASKK bit.

An interrupt signal on an edge-triggered pin can be acknowledged immediately after enabling the pin. An interrupt signal on an edge- and level-triggered interrupt pin must be acknowledged after a delay that depends on the external load.

Another way to avoid a false interrupt:

1. Configure the keyboard pins as outputs by setting the appropriate DDRA bits in data direction register A.
2. Write 1s to the appropriate port A data register bits.
3. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.

9.5 Low-Power Modes

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

9.5.1 Wait Mode

The keyboard module remains active in wait mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of wait mode.

9.5.2 Stop Mode

The keyboard module remains active in stop mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of stop mode.

9.6 Keyboard Module During Break Interrupts

The system integration module (SIM) controls whether the keyboard interrupt latch can be cleared during the break state. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state.

To allow software to clear the keyboard interrupt latch during a break interrupt, write a 1 to the BCFE bit. If a latch is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect the latch during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), writing to the keyboard acknowledge bit (ACKK) in the keyboard status and control register during the break state has no effect. See 9.7.1 Keyboard Status and Control Register.

Read: Write:	P2	P1	P0	STEP4	STEP3	STEP2	STEP1	STEP0
Reset:	0	0	0	0	0	0	0	0

Figure 10-12. HRP Period Registers (HRPPERH:HRPPERL)

P[10:0] — 11-Bit Period Value

STEP[4:0] — 5-Bit Dithering Step Value

10.8.4 HRP Deadtime Register

This read/write register contains an 8-bit value corresponding to the number of HRPCLK cycles that will be subtracted from the logic 1 level of the TOP and BOT output signals to provide deadtime between the two signals.

$$\text{Dead Time} = \frac{\text{HRPDT}}{\text{HRPCLK}} \qquad \textbf{(EQ 10-14)}$$

Address: $0056

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
Reset:	0	0	0	0	1	0	0	0

Figure 10-13. HRP Deadtime Register (HRPDT)

10.8.5 Frequency Dithering HRP Timebase Registers

The two read/write frequency dithering timebase registers HRPTBH:HRPTBL contain a 16-bit value used to determine the time base for switching between the two dithering frequencies. The timebase is calculated from the following formula:

$$\text{Frequency Dithering Timebase (seconds)} = \frac{\text{HRPTBH:HRPTBL}}{\text{HRPCLK}} \qquad \textbf{(EQ 10-15)}$$

Writes to the high byte (HRPTBH) are stored in a latch until the low byte (HRPTBL) is written. Both registers are then updated simultaneously. This prevents glitches occurring on the output signal.

Address: HRPTBH — $0057 HRPTBL — $0058

	Bit 15	14	13	12	11	10	9	Bit 8
Read: Write:	TB15	TB14	TB13	TB12	TB11	TB10	TB9	TB8
Reset:	0	0	0	0	0	0	0	0
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
Reset:	0	0	0	0	0	0	0	0

Figure 10-14. HRP Timebase Registers (HRPTBH:HRPTBL)

11.3.2 Stop Mode

The break module is inactive in stop mode. The STOP instruction does not affect break module register states.

11.4 Central Processor Unit (CPU)

11.4.1 Wait Mode

The WAIT instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling interrupts. After exit from wait mode by interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

11.4.2 Stop Mode

The STOP instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling external interrupts. After exit from stop mode by external interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

After exiting stop mode, the CPU clock begins running after the oscillator stabilization delay.

11.5 Computer Operating Properly Module (COP)

11.5.1 Wait Mode

The COP remains active during wait mode. If COP is enabled, a reset will occur at COP timeout.

11.5.2 Stop Mode

Stop mode turns off the COPCLK input to the COP and clears the COP prescaler. Service the COP immediately before entering or after exiting stop mode to ensure a full COP timeout period after entering or exiting stop mode.

The STOP bit in the CONFIG1 register enables the STOP instruction. To prevent inadvertently turning off the COP with a STOP instruction, disable the STOP instruction by clearing the STOP bit.

11.6 External Interrupt Module (IRQ)

11.6.1 Wait Mode

The external interrupt (IRQ) module remains active in wait mode. Clearing the IMASK bit in the IRQ status and control register enables $\overline{\text{IRQ}}$ CPU interrupt requests to bring the MCU out of wait mode if IRQ function is enabled.

11.6.2 Stop Mode

The IRQ module remains active in stop mode. Clearing the IMASK bit in the IRQ status and control register enables $\overline{\text{IRQ}}$ CPU interrupt requests to bring the MCU out of stop mode.

13.4.6 Internal Oscillator Clock (INTCLK)

INTCLK is the internal oscillator output signal. Its nominal frequency is fixed to 16 MHz, but it can be also trimmed using the oscillator trimming feature of the OSCTRIM register (see 13.3.1.1 Internal Oscillator Trimming).

13.4.7 Oscillator Out 2 (BUSCLKX4)

BUSCLKX4 is the same as the input clock (XTALCLK, RCCLK, or INTCLK). This signal is driven to the SIM module and is used to determine the COP cycles.

13.4.8 Oscillator Out (BUSCLKX2)

The frequency of this signal is equal to half of the BUSCLKX4, this signal is driven to the SIM for generation of the bus clocks used by the CPU and other modules on the MCU. BUSCLKX2 will be divided again in the SIM and results in the internal bus frequency being one fourth of either the XTALCLK, RCCLK, or INTCLK frequency.

13.5 Low Power Modes

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

13.5.1 Wait Mode

The WAIT instruction has no effect on the oscillator logic. BUSCLKX2 and BUSCLKX4 continue to drive to the SIM module.

13.5.2 Stop Mode

The STOP instruction disables either the XTALCLK, the RCCLK, or INTCLK output, hence BUSCLKX2 and BUSCLKX4.

13.6 Oscillator During Break Mode

The oscillator continues to drive BUSCLKX2 and BUSCLKX4 when the device enters the break state.

13.7 CONFIG2 Options

Two CONFIG2 register options affect the operation of the oscillator module: OSCOPT1 and OSCOPT0. All CONFIG2 register bits will have a default configuration. Refer to Chapter 5 Configuration Register (CONFIG) for more information on how the CONFIG2 register is used.

Table 13-2 shows how the OSCOPT bits are used to select the oscillator clock source.

NOTES:
1. X = Don't care
2. I/O pin pulled up to V_{DD} by internal pullup device
3. Writing affects data register, but does not affect input.
4. Hi-Z = High impedance

14.2.3 Port A Input Pullup Enable Register

The port A input pullup enable register (PTAPUE) contains a software configurable pullup device for each of the seven port A pins. Each bit is individually configurable and requires that the data direction register, DDRA, bit be configured as an input. Each pullup is automatically and dynamically disabled when a port bit's DDRA is configured for output mode.

Address: $000D

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:		PTA6PUE	PTA5PUE	PTA4PUE	PTA3PUE	PTA2PUE	PTA1PUE	PTA0PUE
Reset:	-	0	0	0	0	0	0	0

Figure 14-5. Port A Input Pullup Enable Register (PTAPUE)

PTA6PUE–PTA0PUE — Port A Input Pullup Enable Bits

These writable bits are software programmable to enable pullup devices on an input port bit.

1 = Corresponding port A pin configured to have internal pullup
0 = Corresponding port A pin has internal pullup disconnected

14.3 Port B

Port B is an 8-bit special-function port that shares all eight of its pins with the high resolution PWM (HRP), pulse-width modulator (PWM) module, and op amp/comparator module. See Table 1-1 . Pin Functions for a description of the priority of these functions.

14.3.1 Port B Data Register

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

Address: $0001

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Reset:	Unaffected by reset							

Figure 14-6. Port B Data Register (PTB)

PTB7–PTB0 — 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.

15.8.9 Fault Control Register 2

The fault control register 2 (FCR2) is used to acknowledge and clear the FFLAG.

Address: $0044

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	0
Write:								FTACK
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 15-25. Fault Control Register (FCR2)

FTACK — Fault Acknowledge Bit

The FTACK bit is used to acknowledge and clear FFLAG. This bit will always read 0. Writing a 1 to this bit will clear FFLAG. Writing a 0 will have no effect.

15.9 PWM Glossary

CPU cycle

One internal bus cycle (1/BUSCLK)

PWM clock cycle (or period)

One tick of the PWM counter (1/BUSCLK with no prescaler). See Figure 15-26.

PWM cycle (or period)

Edge-aligned mode: The time it takes the PWM counter to count up (modulus/BUSCLK). See Figure 15-26.

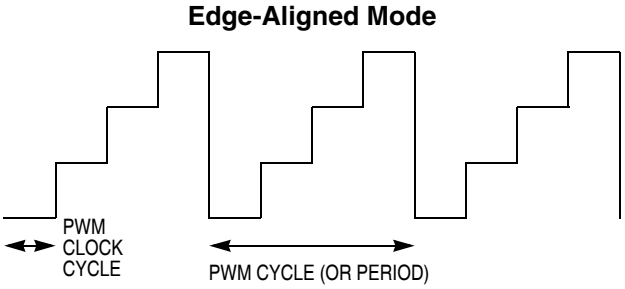


Figure 15-26. PWM Clock Cycle and PWM Cycle Definition

16.3 Interrupts

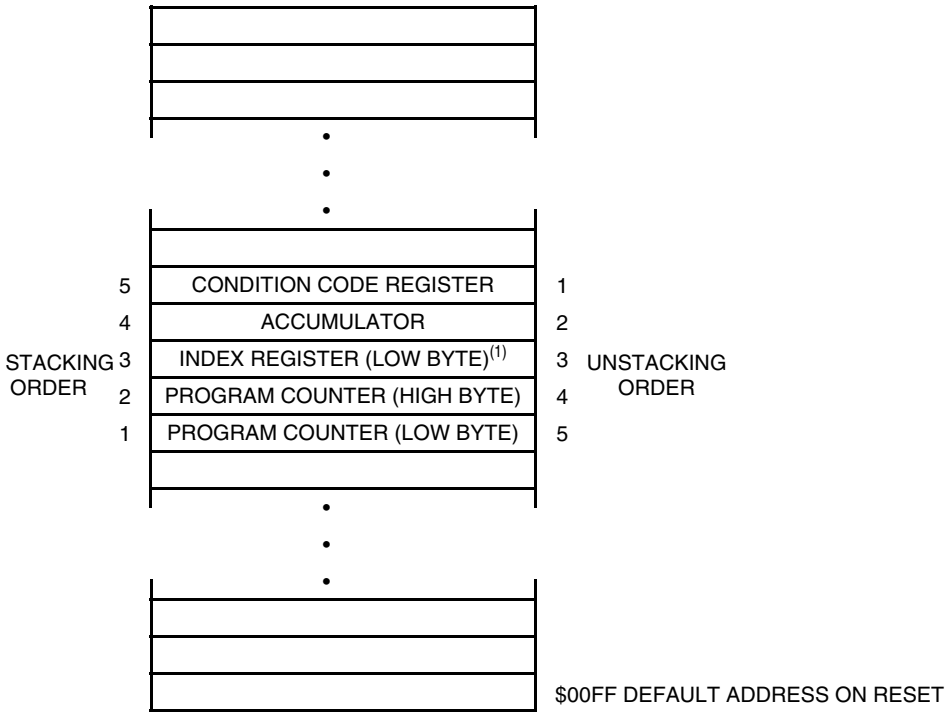
An interrupt temporarily changes the sequence of program execution to respond to a particular event. An interrupt does not stop the operation of the instruction being executed, but begins when the current instruction completes its operation.

16.3.1 Effects

An interrupt:

- Saves the CPU registers on the stack. At the end of the interrupt, the RTI instruction recovers the CPU registers from the stack so that normal processing can resume.
- Sets the interrupt mask (I bit) to prevent additional interrupts. Once an interrupt is latched, no other interrupt can take precedence, regardless of its priority.
- Loads the program counter with a user-defined vector address

After every instruction, the CPU checks all pending interrupts if the I bit is not set. If more than one interrupt is pending when an instruction is done, the highest priority interrupt is serviced first. In the example shown in Figure 16-4, if an interrupt is pending upon exit from the interrupt service routine, the pending interrupt is serviced before the LDA instruction is executed.



1. High byte of index register is not stacked.

Figure 16-3. Interrupt Stacking Order

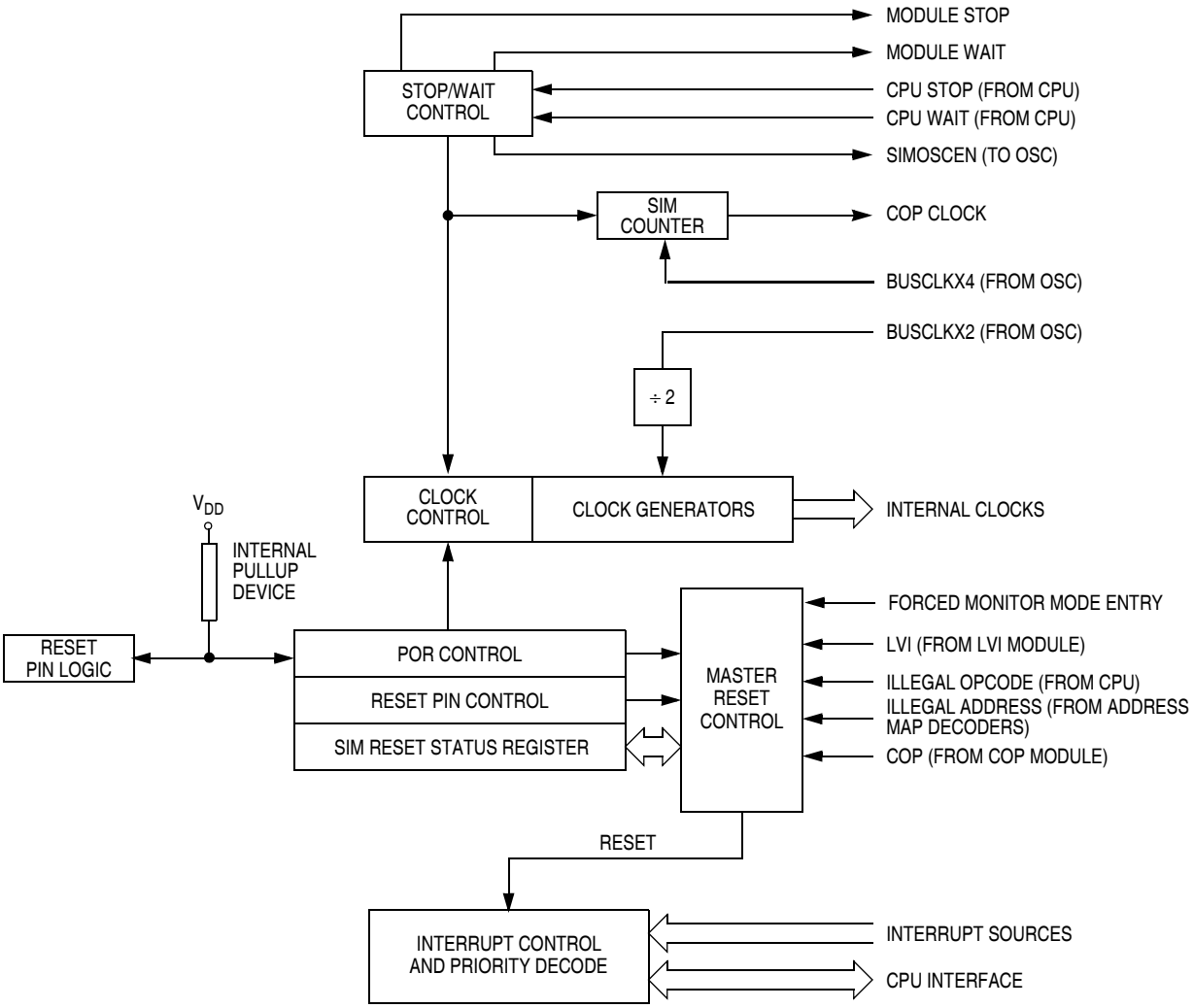


Figure 17-1. SIM Block Diagram

Table 17-1. Signal Name Conventions

Signal Name	Description
BUSCLKX4	Buffered clock from the internal, RC or XTAL oscillator circuit.
BUSCLKX2	The BUSCLKX4 frequency divided by two. This signal is again divided by two in the SIM to generate the internal bus clocks (bus clock = BUSCLKX4 ÷ 4).
IAB	Internal address bus
IDB	Internal data bus
PORRST	Signal from the power-on reset module to the SIM
IRST	Internal reset signal
R/$\overline{W}$	Read/write signal

cycles later, the CPU is released from reset to allow the reset vector sequence to occur. The SIM actively pulls down the $\overline{RST}$ pin for all internal reset sources.

17.3.2.6 Monitor Mode Entry Module Reset (MODRST)

The monitor mode entry module reset (MODRST) asserts its output to the SIM when monitor mode is entered in the condition where the reset vectors are erased ($FF). When MODRST gets asserted, an internal reset occurs. The SIM actively pulls down the $\overline{RST}$ pin for all internal reset sources.

17.4 SIM Counter

The SIM counter is used by the power-on reset module (POR) and in stop mode recovery to allow the oscillator time to stabilize before enabling the internal bus (IBUS) clocks. The SIM counter is 13 bits long.

17.4.1 SIM Counter During Power-On Reset

The power-on reset module (POR) detects power applied to the MCU. At power-on, the POR circuit asserts the signal PORRST. Once the SIM is initialized, it enables the clock generation module (CGM) to drive the bus clock state machine.

17.4.2 SIM Counter During Stop Mode Recovery

The SIM counter also is used for stop mode recovery. The STOP instruction clears the SIM counter. After an interrupt, break, or reset, the SIM senses the state of the short stop recovery bit, SSREC, in the mask option register. If the SSREC bit is a 1, then the stop recovery is reduced from the normal delay of 4096 BUSCLKX4 cycles down to 32 BUSCLKX4 cycles. This is ideal for applications using canned oscillators that do not require long startup times from stop mode. External crystal applications should use the full stop recovery time, that is, with SSREC cleared.

17.4.3 SIM Counter and Reset States

External reset has no effect on the SIM counter. See 17.6.2 Stop Mode for details. The SIM counter is free-running after all reset states. See 17.3.2 Active Resets from Internal Sources for counter control and internal reset recovery sequences.

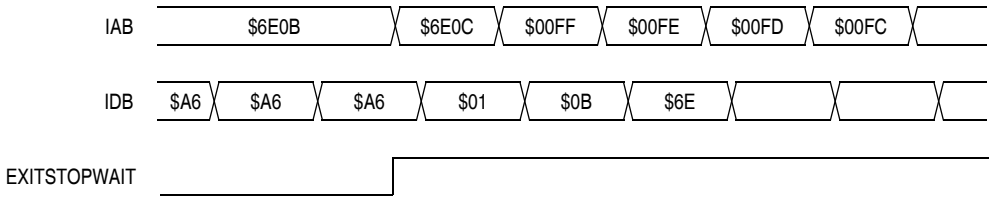
17.5 Exception Control

Normal, sequential program execution can be changed in three different ways:

- Interrupts:
  - Maskable hardware CPU interrupts
  - Non-maskable software interrupt instruction (SWI)
- Reset
- Break interrupts

17.5.1 Interrupts

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. Figure 17-8 shows interrupt entry timing. Figure 17-9 shows interrupt recovery timing.



Note: EXITSTOPWAIT = $\overline{\text{RST}}$ pin or CPU interrupt

Figure 17-13. Wait Recovery from Interrupt

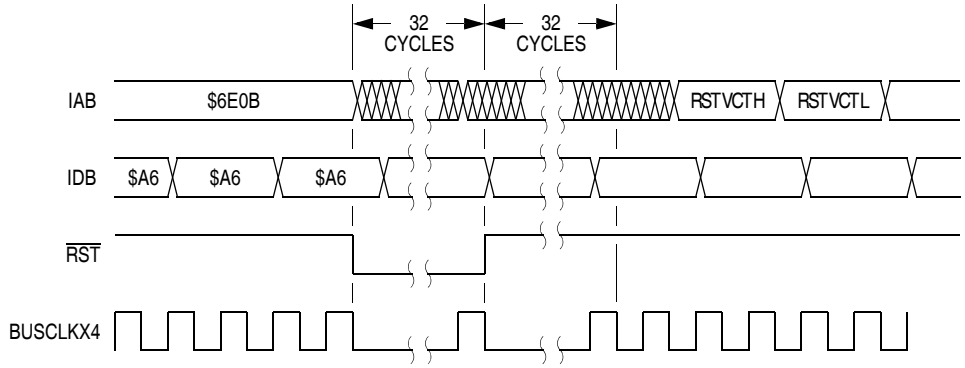


Figure 17-14. Wait Recovery from Internal Reset

17.6.2 Stop Mode

In stop mode, the SIM counter is reset and the system clocks are disabled. An interrupt request from a module can cause an exit from stop mode. Stacking for interrupts begins after the selected stop recovery time has elapsed. Reset also causes an exit from stop mode.

The SIM disables the clock generator module outputs (BUSCLKX2 and BUSCLKX4) in stop mode, stopping the CPU and peripherals. Stop recovery time is selectable using the SSREC bit in the mask option register (MOR). If SSREC is set, stop recovery is reduced from the normal delay of 4096 BUSCLKX4 cycles down to 32. This is ideal for applications using canned oscillators that do not require long startup times from stop mode.

NOTE

External crystal applications should use the full stop recovery time by clearing the SSREC bit.

The SIM counter is held in reset from the execution of the STOP instruction until the beginning of stop recovery. It is then used to time the recovery period. Figure 17-15 shows stop mode entry timing. Figure 17-16 shows stop mode recovery time from interrupt or break.

NOTE

To minimize stop current, all pins configured as inputs should be driven to a logic 1 or logic 0.

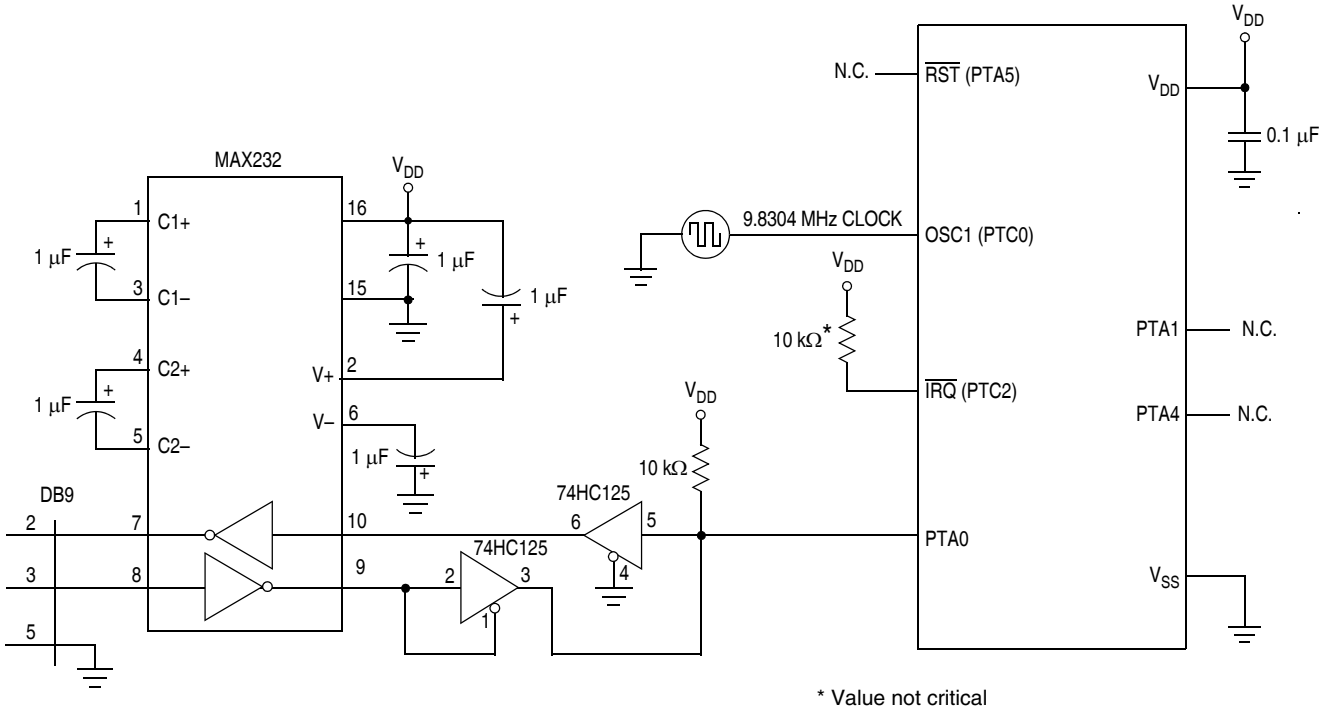


Figure 19-11. Forced Monitor Mode Circuit (External Clock, No High Voltage)

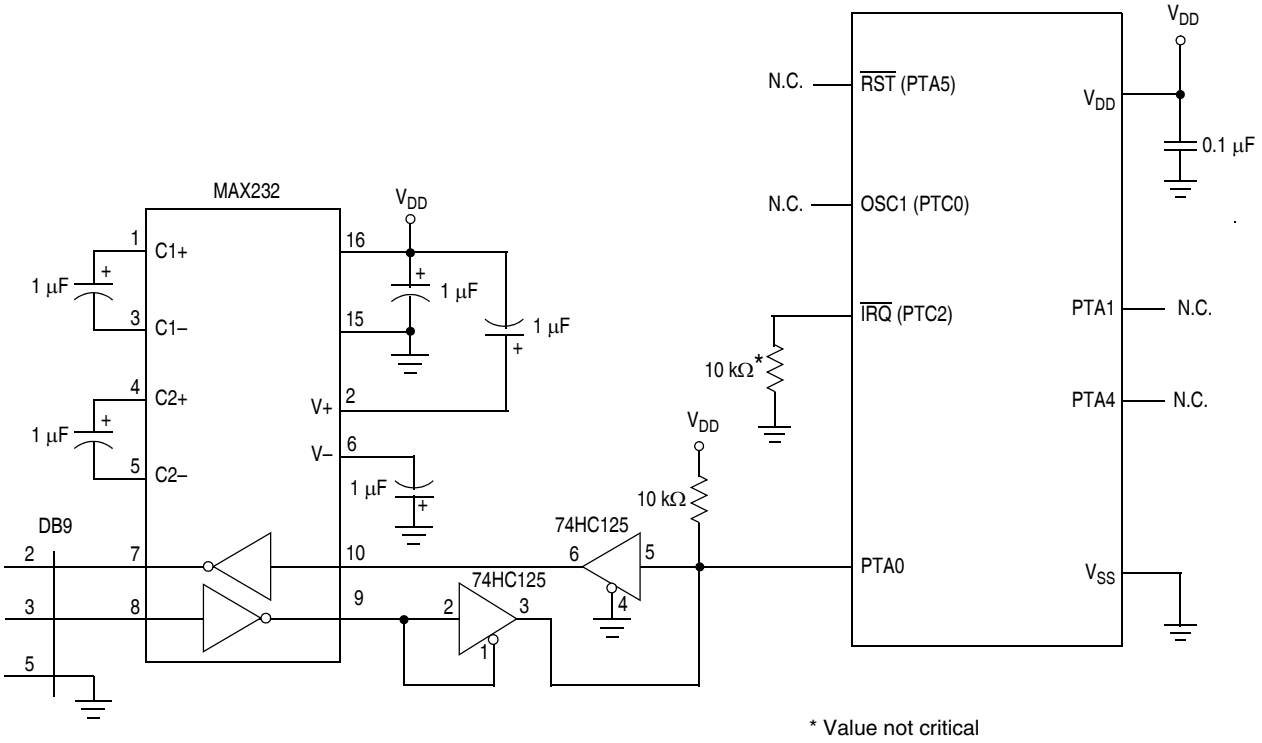


Figure 19-12. Forced Monitor Mode Circuit (Internal Clock, No High Voltage)

Table 19-1. Monitor Mode Signal Requirements and Options

Mode	$\overline{IRQ}$ (PTC2)	$\overline{RST}$ (PTA5)	Reset Vector	Serial Communication	Mode Selection		COP	Communication Speed			Comments
				PTA0	PTA1	PTA4		External Clock	Bus Frequency	Baud Rate	
—	X	GND	X	X	X	X	X	X	X	X	Reset condition
Normal Monitor	V_{TST}	V_{DD}	X	1	1	0	Disabled	9.8304 MHz	2.4576 MHz	9600	Provide external clock at OSC1
Forced Monitor	V_{DD}	V_{DD}	$FF (blank)	1	X	X	Disabled	9.8304 MHz	2.4576 MHz	9600	Provide external clock at OSC1
	GND	V_{DD}	$FF (blank)	1	X	X	Disabled	X	4 MHz	9600	Internal clock is active
User	V_{DD} or GND	V_{DD}	Not $FF	X	X	X	Enabled	X	X	X	
MON08 Function [Pin No.]	V_{TST} [6]	$\overline{RST}$ [4]	—	COM [8]	MOD0 [12]	MOD1 [10]	—	OSC1 [13]	—	—	

1. PTA0 must have a pullup resistor to V_{DD} in monitor mode.
2. Communication speed in the table is an example to obtain a baud rate of 9600. Baud rate using external oscillator is bus frequency / 256 and baud rate using internal oscillator is bus frequency / 417.
3. External clock is an 9.8304 MHz on OSC1.
4. X = don't care
5. MON08 pin refers to P&E Microcomputer Systems' MON08-Cyclone 2 by 8-pin connector.

NC	1	2	GND
NC	3	4	$\overline{RST}$
NC	5	6	IRQ
NC	7	8	PTA0
NC	9	10	PTA4
NC	11	12	PTA1
OSC1	13	14	NC
V_{DD}	15	16	NC