

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	Active
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LVD, POR, PWM
Number of I/O	5
Program Memory Size	1.5KB (1.5K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.209", 5.30mm Width)
Supplier Device Package	8-SO
Purchase URL	<a href="https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mchc908qt1vdwe">https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mchc908qt1vdwe</a>

## Revision History (Sheet 3 of 3)

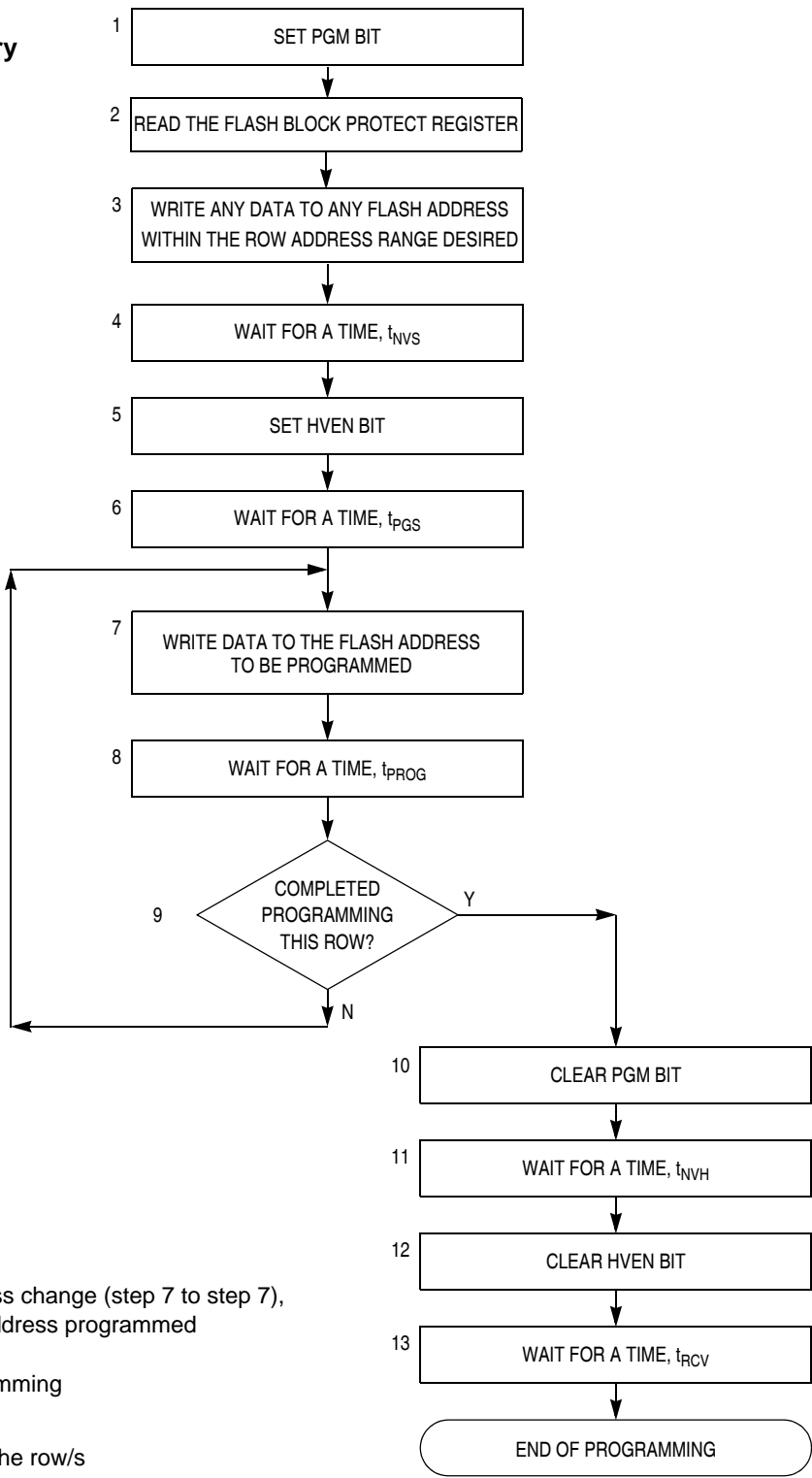
Date	Revision Level	Description	Page Number(s)
November, 2004	4.0	Reformatted to meet current documentation standards	Throughout
		<a href="#">6.3.1 BUSCLKX4</a> — Clarified description of BUSCLKX4	58
		<a href="#">Chapter 7 Central Processor Unit (CPU)</a> — In <a href="#">7.7 Instruction Set Summary</a> : Reworked definitions for STOP instruction Added WAIT instruction	70 71
		<a href="#">13.8.1 SIM Reset Status Register</a> — Clarified SRSR flag setting	117
		<a href="#">14.9.1 TIM Status and Control Register</a> — Added information to TSTOP note	127
		<a href="#">16.8 5-V Oscillator Characteristics</a> — Added values for deviation from trimmed internal oscillator	155
		<a href="#">16.12 3-V Oscillator Characteristics</a> — Added values for deviation from trimmed internal oscillator	158
July, 2005	5.0	<a href="#">Figure 5-2. Configuration Register 1 (CONFIG1)</a> — Clarified bit definitions for COPRS.	54
		<a href="#">Chapter 8 External Interrupt (IRQ)</a> — Reworked for clarification.	73
		<a href="#">11.3.4 RC Oscillator</a> — Improved RC oscillator wording.	93
		<a href="#">12.1 Introduction</a> — Added note pertaining to non-bonded port pins.	97
		<a href="#">17.3 Package Dimensions</a> — Updated package information.	165
March, 2010	6.0	Clarify internal oscillator trim register information.	26, 27, 31, 34, 35, 38, 91, 96



## General Description

- On-chip in-application programmable FLASH memory (with internal program/erase voltage generation)
  - MC68HC908QY4 and MC68HC908QT4 — 4096 bytes
  - MC68HC908QY2, MC68HC908QY1, MC68HC908QT2, and MC68HC908QT1 — 1536 bytes
- 128 bytes of on-chip random-access memory (RAM)
- 2-channel, 16-bit timer interface module (TIM)
- 4-channel, 8-bit analog-to-digital converter (ADC) on MC68HC908QY2, MC68HC908QY4, MC68HC908QT2, and MC68HC908QT4
- 5 or 13 bidirectional input/output (I/O) lines and one input only:
  - Six shared with keyboard interrupt function and ADC
  - Two shared with timer channels
  - One shared with external interrupt (IRQ)
  - Eight extra I/O lines on 16-pin package only
  - High current sink/source capability on all port pins
  - Selectable pullups on all ports, selectable on an individual bit basis
  - Three-state ability on all port pins
- 6-bit keyboard interrupt with wakeup feature (KBI)
- Low-voltage inhibit (LVI) module features:
  - Software selectable trip point in CONFIG register
- System protection features:
  - Computer operating properly (COP) watchdog
  - Low-voltage detection with reset
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- External asynchronous interrupt pin with internal pullup ( $\overline{\text{IRQ}}$ ) shared with general-purpose input pin
- Master asynchronous reset pin ( $\overline{\text{RST}}$ ) shared with general-purpose input/output (I/O) pin
- Power-on reset
- Internal pullups on  $\overline{\text{IRQ}}$  and  $\overline{\text{RST}}$  to reduce external components
- Memory mapped I/O registers
- Power saving stop and wait modes
- MC68HC908QY4, MC68HC908QY2, and MC68HC908QY1 are available in these packages:
  - 16-pin plastic dual in-line package (PDIP)
  - 16-pin small outline integrated circuit (SOIC) package
  - 16-pin thin shrink small outline package (TSSOP)
- MC68HC908QT4, MC68HC908QT2, and MC68HC908QT1 are available in these packages:
  - 8-pin PDIP
  - 8-pin SOIC
  - 8-pin dual flat no lead (DFN) package

**Algorithm for Programming  
a Row (32 Bytes) of FLASH Memory**



**NOTES:**

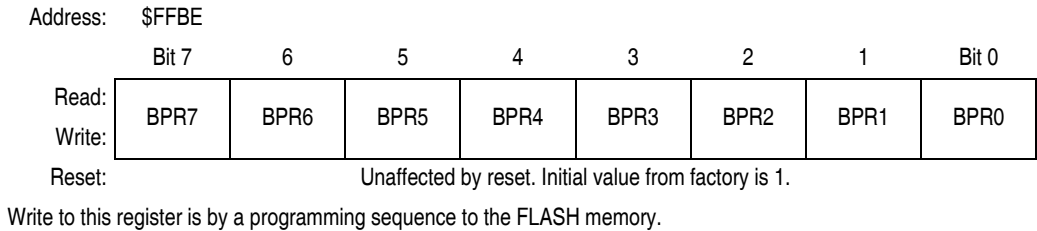
The time between each FLASH address change (step 7 to step 7), or the time between the last FLASH address programmed to clearing PGM bit (step 7 to step 10) must not exceed the maximum programming time,  $t_{\text{PROG max}}$ .

This row program algorithm assumes the row/s to be programmed are initially erased.

**Figure 2-4. FLASH Programming Flowchart**

## 2.6.6 FLASH Block Protect Register

The FLASH block protect register is implemented as a byte within the FLASH memory, and therefore can only be written during a programming sequence of the FLASH memory. The value in this register determines the starting address of the protected range within the FLASH memory.

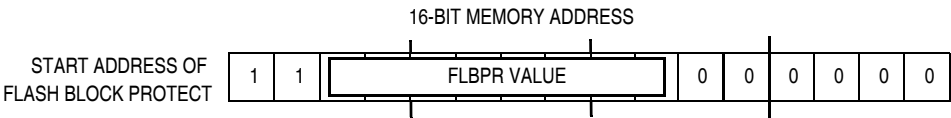


**Figure 2-5. FLASH Block Protect Register (FLBPR)**

### BPR[7:0] — FLASH Protection Register Bits [7:0]

These eight bits in FLBPR represent bits [13:6] of a 16-bit memory address. Bits [15:14] are 1s and bits [5:0] are 0s.

The resultant 16-bit address is used for specifying the start address of the FLASH memory for block protection. The FLASH is protected from this start address to the end of FLASH memory, at \$FFFF. With this mechanism, the protect start address can be XX00, XX40, XX80, or XXC0 within the FLASH memory. See [Figure 2-6](#) and [Table 2-2](#).



**Figure 2-6. FLASH Block Protect Start Address**

**Table 2-2. Examples of Protect Start Address**

BPR[7:0]	Start of Address of Protect Range
\$00–\$B8	The entire FLASH memory is protected.
\$B9 (1011 1001)	\$EE40 (1110 1110 0100 0000)
\$BA (1011 1010)	\$EE80 (1110 1110 1000 0000)
\$BB (1011 1011)	\$EEC0 (1110 1110 1100 0000)
\$BC (1011 1100)	\$EF00 (1110 1111 0000 0000)
and so on...	
\$DE (1101 1110)	\$F780 (1111 0111 1000 0000)
\$DF (1101 1111)	\$F7C0 (1111 0111 1100 0000)
\$FE (1111 1110)	\$FF80 (1111 1111 1000 0000) FLBPR, internal oscillator trim values, and vectors are protected
\$FF	The entire FLASH memory is not protected.

## Auto Wakeup Module (AWU)

### Bits 7–4 — Not used

These read-only bits always read as 0s.

### KEYF — Keyboard Flag Bit

This read-only bit is set when a keyboard interrupt is pending on port A or auto wakeup. Reset clears the KEYF bit.

- 1 = Keyboard/auto wakeup interrupt pending
- 0 = No keyboard/auto wakeup interrupt pending

### ACKK — Keyboard Acknowledge Bit

Writing a 1 to this write-only bit clears the keyboard/auto wakeup interrupt request on port A and auto wakeup logic. ACKK always reads as 0. Reset clears ACKK.

### IMASKK — Keyboard Interrupt Mask Bit

Writing a 1 to this read/write bit prevents the output of the keyboard interrupt mask from generating interrupt requests on port A or auto wakeup. Reset clears the IMASKK bit.

- 1 = Keyboard/auto wakeup interrupt requests masked
- 0 = Keyboard/auto wakeup interrupt requests not masked

#### NOTE


*MODEK is not used in conjunction with the auto wakeup feature. To see a description of this bit, see [9.7.1 Keyboard Status and Control Register](#).*

## 4.6.3 Keyboard Interrupt Enable Register

The keyboard interrupt enable register (KBIER) enables or disables the auto wakeup to operate as a keyboard/auto wakeup interrupt input.

Address: \$001B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	AWUIE	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Write:								
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

**Figure 4-4. Keyboard Interrupt Enable Register (KBIER)**

### AWUIE — Auto Wakeup Interrupt Enable Bit

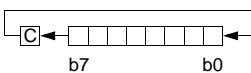
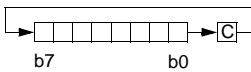
This read/write bit enables the auto wakeup interrupt input to latch interrupt requests. Reset clears AWUIE.

- 1 = Auto wakeup enabled as interrupt input
- 0 = Auto wakeup not enabled as interrupt input

#### NOTE

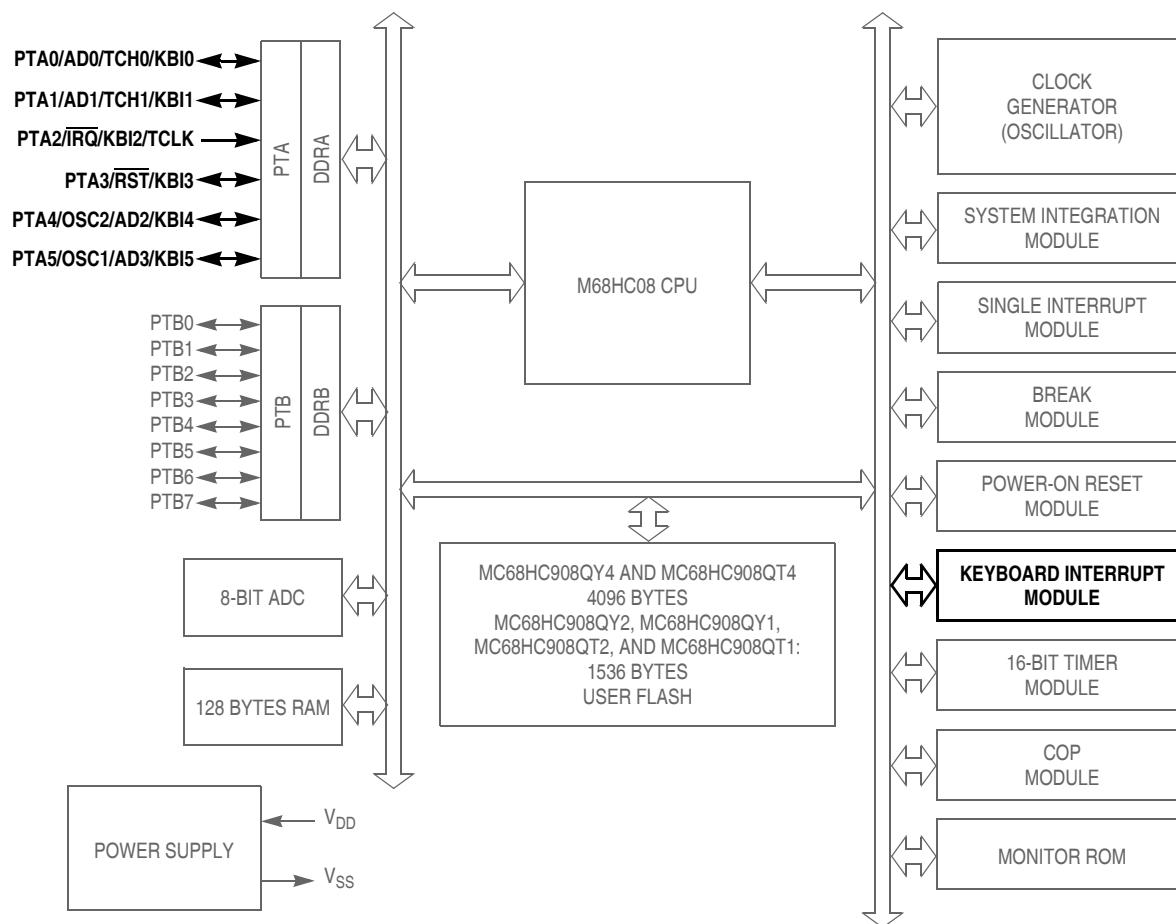
*KBIE5–KBIE0 bits are not used in conjunction with the auto wakeup feature. To see a description of these bits, see [9.7.2 Keyboard Interrupt Enable Register](#).*

Table 7-1. Instruction Set Summary (Sheet 5 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z				
PULA	Pull A from Stack	$SP \leftarrow (SP + 1); \text{Pull (A)}$	–	–	–	–	–	INH	86		2
PULH	Pull H from Stack	$SP \leftarrow (SP + 1); \text{Pull (H)}$	–	–	–	–	–	INH	8A		2
PULX	Pull X from Stack	$SP \leftarrow (SP + 1); \text{Pull (X)}$	–	–	–	–	–	INH	88		2
ROL <i>opr</i> ROLA ROLX ROL <i>opr</i> ,X ROL ,X ROL <i>opr</i> ,SP	Rotate Left through Carry		↑	–	–	–	↑	DIR INH INH IX1 IX SP1	39 49 59 69 79 9E69	dd ff ff	4 1 1 4 3 5
ROR <i>opr</i> RORA RORX ROR <i>opr</i> ,X ROR ,X ROR <i>opr</i> ,SP	Rotate Right through Carry		↑	–	–	–	↑	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	4 1 1 4 3 5
RSP	Reset Stack Pointer	$SP \leftarrow \$FF$	–	–	–	–	–	INH	9C		1
RTI	Return from Interrupt	$SP \leftarrow (SP + 1); \text{Pull (CCR)}$ $SP \leftarrow (SP + 1); \text{Pull (A)}$ $SP \leftarrow (SP + 1); \text{Pull (X)}$ $SP \leftarrow (SP + 1); \text{Pull (PCH)}$ $SP \leftarrow (SP + 1); \text{Pull (PCL)}$	↑	↑	↑	↑	↑	INH	80		7
RTS	Return from Subroutine	$SP \leftarrow SP + 1; \text{Pull (PCH)}$ $SP \leftarrow SP + 1; \text{Pull (PCL)}$	–	–	–	–	–	INH	81		4
SBC # <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> ,X SBC <i>opr</i> ,X SBC ,X SBC <i>opr</i> ,SP SBC <i>opr</i> ,SP	Subtract with Carry	$A \leftarrow (A) - (M) - (C)$	↑	–	–	–	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A2 B2 C2 D2 E2 F2 9EE2 9ED2	ii dd hh ll ee ff ff ff ff ff ee ff	2 3 4 4 3 2 4 5
SEC	Set Carry Bit	$C \leftarrow 1$	–	–	–	–	1	INH	99		1
SEI	Set Interrupt Mask	$I \leftarrow 1$	–	–	1	–	–	INH	9B		2
STA <i>opr</i> STA <i>opr</i> STA <i>opr</i> ,X STA <i>opr</i> ,X STA ,X STA <i>opr</i> ,SP STA <i>opr</i> ,SP	Store A in M	$M \leftarrow (A)$	0	–	–	–	↑	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh ll ee ff ff ff ff ff ee ff	3 4 4 3 2 4 5
STHX <i>opr</i>	Store H:X in M	$(M:M + 1) \leftarrow (H:X)$	0	–	–	–	↑	DIR	35	dd	4
STOP	Enable Interrupts, Stop Processing, Refer to MCU Documentation	$I \leftarrow 0$ ; Stop Processing	–	–	0	–	–	INH	8E		1
STX <i>opr</i> STX <i>opr</i> STX <i>opr</i> ,X STX <i>opr</i> ,X STX ,X STX <i>opr</i> ,SP STX <i>opr</i> ,SP	Store X in M	$M \leftarrow (X)$	0	–	–	–	↑	DIR EXT IX2 IX1 IX SP1 SP2	BF CF DF EF FF 9EEF 9EDF	dd hh ll ee ff ff ff ff ff ee ff	3 4 4 3 2 4 5
SUB # <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> ,X SUB <i>opr</i> ,X SUB ,X SUB <i>opr</i> ,SP SUB <i>opr</i> ,SP	Subtract	$A \leftarrow (A) - (M)$	↑	–	–	–	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A0 B0 C0 D0 E0 F0 9EE0 9ED0	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5



## Keyboard Interrupt Module (KBI)



$\overline{RST}$ ,  $\overline{IRQ}$ : Pins have internal (about 30K Ohms) pull up

PTA[0:5]: High current sink and source capability

PTA[0:5]: Pins have programmable keyboard interrupt and pull up

PTB[0:7]: Not available on 8-pin devices – MC68HC908QT1, MC68HC908QT2, and MC68HC908QT4 (see note in [12.1 Introduction](#))

ADC: Not available on the MC68HC908QY1 and MC68HC908QT1

**Figure 9-1. Block Diagram Highlighting KBI Block and Pins**

### 11.3.1.1 Internal Oscillator Trimming

The 8-bit trimming register, OSCTRIM, allows a clock period adjust of +127 and –128 steps. Increasing OSCTRIM value increases the clock period. Trimming allows the internal clock frequency to be set to 12.8 MHz  $\pm$  5%.

All devices are factory programmed with trim values in reserved FLASH memory locations \$FFC0 and \$FFC1. The trim value is not automatically loaded into the OSCTRIM register. User software must copy the trim value from \$FFC0 or \$FFC1 into OSCTRIM if needed. The factory trim value provides the accuracy required for communication using forced monitor mode. Some production programmers erase the factory trim values, so confirm with your programmer vendor that the trim values at \$FFC0 and \$FFC1 are preserved, or are re-trimmed. Trimming the device in the user application board will provide the most accurate trim value.

### 11.3.1.2 Internal to External Clock Switching

When external clock source (external OSC, RC, or XTAL) is desired, the user must perform the following steps:

1. For external crystal circuits only, OSCOPT[1:0] = 1:1: To help precharge an external crystal oscillator, set PTA4 (OSC2) as an output and drive high for several cycles. This may help the crystal circuit start more robustly.
2. Set CONFIG2 bits OSCOPT[1:0] according to . The oscillator module control logic will then set OSC1 as an external clock input and, if the external crystal option is selected, OSC2 will also be set as the clock output.
3. Create a software delay to wait the stabilization time needed for the selected clock source (crystal, resonator, RC) as recommended by the component manufacturer. A good rule of thumb for crystal oscillators is to wait 4096 cycles of the crystal frequency, i.e., for a 4-MHz crystal, wait approximately 1 msec.
4. After the manufacturer's recommended delay has elapsed, the ECGON bit in the OSC status register (OSCSTAT) needs to be set by the user software.
5. After ECGON set is detected, the OSC module checks for oscillator activity by waiting two external clock rising edges.
6. The OSC module then switches to the external clock. Logic provides a glitch free transition.
7. The OSC module first sets the ECGST bit in the OSCSTAT register and then stops the internal oscillator.

#### NOTE

*Once transition to the external clock is done, the internal oscillator will only be reactivated with reset. No post-switch clock monitor feature is implemented (clock does not switch back to internal if external clock dies).*

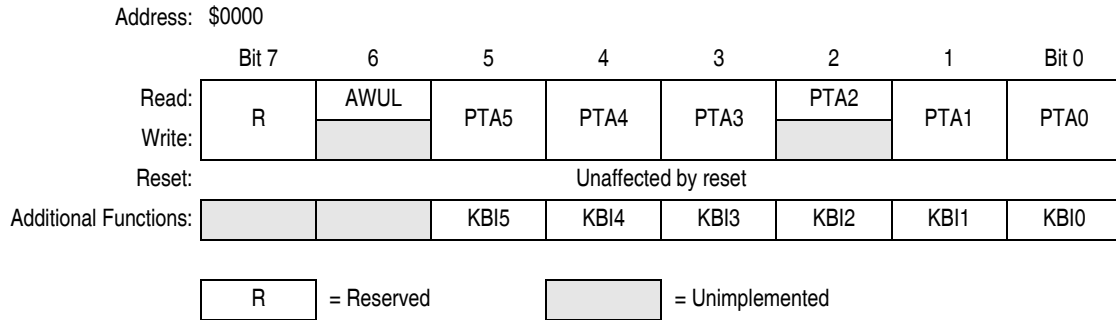
### 11.3.2 External Oscillator

The external clock option is designed for use when a clock signal is available in the application to provide a clock source to the microcontroller. The OSC1 pin is enabled as an input by the oscillator module. The clock signal is used directly to create BUSCLKX4 and also divided by two to create BUSCLKX2.

In this configuration, the OSC2 pin cannot output BUSCLKX4. So the OSC2EN bit in the port A pullup enable register will be clear to enable PTA4 I/O functions on the pin

## 12.2.1 Port A Data Register

The port A data register (PTA) contains a data latch for each of the six port A pins.



**Figure 12-1. Port A Data Register (PTA)**

### PTA[5:0] — Port A Data Bits

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

### AWUL — Auto Wakeup Latch Data Bit

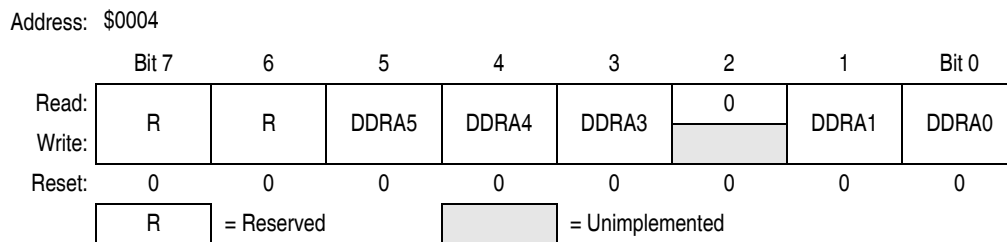
This is a read-only bit which has the value of the auto wakeup interrupt request latch. The wakeup request signal is generated internally (see [Chapter 4 Auto Wakeup Module \(AWU\)](#)). There is no PTA6 port nor any of the associated bits such as PTA6 data register, pullup enable or direction.

### KBI[5:0] — Port A Keyboard Interrupts

The keyboard interrupt enable bits, KBIE5–KBIE0, in the keyboard interrupt control enable register (KBIER) enable the port A pins as external interrupt pins (see [Chapter 9 Keyboard Interrupt Module \(KBI\)](#)).

## 12.2.2 Data Direction Register A

Data direction register A (DDRA) determines whether each port A pin is an input or an output. Writing a 1 to a DDRA bit enables the output buffer for the corresponding port A pin; a 0 disables the output buffer.



**Figure 12-2. Data Direction Register A (DDRA)**

### DDRA[5:0] — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA[5:0], configuring all port A pins as inputs.

- 1 = Corresponding port A pin configured as output
- 0 = Corresponding port A pin configured as input

### NOTE

*Avoid glitches on port A pins by writing to the port A data register before changing data direction register A bits from 0 to 1.*

### 12.3.3 Port B Input Pullup Enable Register

The port B input pullup enable register (PTBPUE) contains a software configurable pullup device for each of the eight port B pins. Each bit is individually configurable and requires the corresponding data direction register, DDRBx, be configured as input. Each pullup device is automatically and dynamically disabled when its corresponding DDRBx bit is configured as output.

Address: \$000C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTBPUE7	PTBPUE6	PTBPUE5	PTBPUE4	PTBPUE3	PTBPUE2	PTBPUE1	PTBPUE0
Write:	PTBPUE7	PTBPUE6	PTBPUE5	PTBPUE4	PTBPUE3	PTBPUE2	PTBPUE1	PTBPUE0
Reset:	0	0	0	0	0	0	0	0

**Figure 12-8. Port B Input Pullup Enable Register (PTBPUE)**

#### PTBPUE[7:0] — Port B Input Pullup Enable Bits

- These read/write bits are software programmable to enable pullup devices on port B pins
- 1 = Corresponding port B pin configured to have internal pull if its DDRB bit is set to 0
  - 0 = Pullup device is disconnected on the corresponding port B pin regardless of the state of its DDRB bit.

Table 12-3 summarizes the operation of the port B pins.

**Table 12-3. Port B Pin Functions**

PTBPUE Bit	DDRBB Bit	PTB Bit	I/O Pin Mode	Accesses to DDRB	Accesses to PTB	
				Read/Write	Read	Write
1	0	X <sup>(1)</sup>	Input, V <sub>DD</sub> <sup>(2)</sup>	DDRBB7–DDRBB0	Pin	PTB7–PTB0 <sup>(3)</sup>
0	0	X	Input, Hi-Z <sup>(4)</sup>	DDRBB7–DDRBB0	Pin	PTB7–PTB0 <sup>(3)</sup>
X	1	X	Output	DDRBB7–DDRBB0	PTB7–PTB0	PTB7–PTB0

1. X = don't care
2. I/O pin pulled to V<sub>DD</sub> by internal pullup.
3. Writing affects data register, but does not affect input.
4. Hi-Z = high impedance

### 13.8.1 SIM Reset Status Register

The SRSR register contains flags that show the source of the last reset. The status register will automatically clear after reading SRSR. A power-on reset sets the POR bit and clears all other bits in the register. All other reset sources set the individual flag bits but do not clear the register. More than one reset source can be flagged at any time depending on the conditions at the time of the internal or external reset. For example, the POR and LVI bit can both be set if the power supply has a slow rise time.

Address: \$FE01

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
Write:								
POR:	1	0	0	0	0	0	0	0


 = Unimplemented

Figure 13-19. SIM Reset Status Register (SRSR)

#### POR — Power-On Reset Bit

- 1 = Last reset caused by POR circuit
- 0 = Read of SRSR

#### PIN — External Reset Bit

- 1 = Last reset caused by external reset pin ( $\overline{\text{RST}}$ )
- 0 = POR or read of SRSR

#### COP — Computer Operating Properly Reset Bit

- 1 = Last reset caused by COP counter
- 0 = POR or read of SRSR

#### ILOP — Illegal Opcode Reset Bit

- 1 = Last reset caused by an illegal opcode
- 0 = POR or read of SRSR

#### ILAD — Illegal Address Reset Bit (illegal attempt to fetch an opcode from an unimplemented address)

- 1 = Last reset caused by an opcode fetch from an illegal address
- 0 = POR or read of SRSR

#### MODRST — Monitor Mode Entry Module Reset Bit

- 1 = Last reset caused by monitor mode entry when vector locations \$FFFE and \$FFFF are \$FF after POR while  $\overline{\text{IRQ}} \neq V_{\text{TST}}$
- 0 = POR or read of SRSR

#### LVI — Low Voltage Inhibit Reset Bit

- 1 = Last reset caused by LVI circuit
- 0 = POR or read of SRSR

### 14.4.1 TIM Counter Prescaler

The TIM clock source is one of the seven prescaler outputs or the TIM clock pin, TCLK. The prescaler generates seven clock rates from the internal bus clock. The prescaler select bits, PS[2:0], in the TIM status and control register (TSC) select the TIM clock source.

### 14.4.2 Input Capture

With the input capture function, the TIM can capture the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the TIM latches the contents of the TIM counter into the TIM channel registers, TCHxH:TCHxL. The polarity of the active edge is programmable. Input captures can generate TIM central processor unit (CPU) interrupt requests.

### 14.4.3 Output Compare

With the output compare function, the TIM can generate a periodic pulse with a programmable polarity, duration, and frequency. When the counter reaches the value in the registers of an output compare channel, the TIM can set, clear, or toggle the channel pin. Output compares can generate TIM CPU interrupt requests.

#### 14.4.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in [14.4.3 Output Compare](#). The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change an output compare value could cause incorrect operation for up to two counter overflow periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that counter overflow period. Also, using a TIM overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIM may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the output compare value on channel x:

- When changing to a smaller value, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current output compare pulse. The interrupt routine has until the end of the counter overflow period to write the new value.
- When changing to a larger output compare value, enable TIM overflow interrupts and write the new value in the TIM overflow interrupt routine. The TIM overflow interrupt occurs at the end of the current counter overflow period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same counter overflow period.

#### 14.4.3.2 Buffered Output Compare

Channels 0 and 1 can be linked to form a buffered output compare channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the output.

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The output compare value in the TIM channel 0 registers initially controls the output on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the output after the TIM overflows. At each subsequent overflow, the TIM channel registers (0 or 1) that

### 14.4.4.3 PWM Initialization

To ensure correct operation when generating unbuffered or buffered PWM signals, use the following initialization procedure:

1. In the TIM status and control register (TSC):
  - a. Stop the TIM counter by setting the TIM stop bit, TSTOP.
  - b. Reset the TIM counter and prescaler by setting the TIM reset bit, TRST.
2. In the TIM counter modulo registers (TMODH:TMODL), write the value for the required PWM period.
3. In the TIM channel x registers (TCHxH:TCHxL), write the value for the required pulse width.
4. In TIM channel x status and control register (TSCx):
  - a. Write 0:1 (for unbuffered output compare or PWM signals) or 1:0 (for buffered output compare or PWM signals) to the mode select bits, MSxB:MSxA. See [Table 14-3](#).
  - b. Write 1 to the toggle-on-overflow bit, TOVx.
  - c. Write 1:0 (polarity 1 — to clear output on compare) or 1:1 (polarity 0 — to set output on compare) to the edge/level select bits, ELSxB:ELSxA. The output action on compare must force the output to the complement of the pulse width level. See [Table 14-3](#).

#### NOTE

*In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare can also cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.*

5. In the TIM status control register (TSC), clear the TIM stop bit, TSTOP.

Setting MS0B links channels 0 and 1 and configures them for buffered PWM operation. The TIM channel 0 registers (TCH0H:TCH0L) initially control the buffered PWM output. TIM status control register 0 (TSCR0) controls and monitors the PWM signal from the linked channels. MS0B takes priority over MS0A.

Clearing the toggle-on-overflow bit, TOVx, inhibits output toggles on TIM overflows. Subsequent output compares try to force the output to a state it is already in and have no effect. The result is a 0% duty cycle output.

Setting the channel x maximum duty cycle bit (CHxMAX) and setting the TOVx bit generates a 100% duty cycle output. See [14.9.4 TIM Channel Status and Control Registers](#).

## 14.5 Interrupts

The following TIM sources can generate interrupt requests:

- TIM overflow flag (TOF) — The TOF bit is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. The TIM overflow interrupt enable bit, TOIE, enables TIM overflow CPU interrupt requests. TOF and TOIE are in the TIM status and control register.
- TIM channel flags (CH1F:CH0F) — The CHxF bit is set when an input capture or output compare occurs on channel x. Channel x TIM CPU interrupt requests are controlled by the channel x interrupt enable bit, CHxIE. Channel x TIM CPU interrupt requests are enabled when CHxIE = 1. CHxF and CHxIE are in the TIM channel x status and control register.

### 14.9.3 TIM Counter Modulo Registers

The read/write TIM modulo registers contain the modulo value for the TIM counter. When the TIM counter reaches the modulo value, the overflow flag (TOF) becomes set, and the TIM counter resumes counting from \$0000 at the next timer clock. Writing to the high byte (TMODH) inhibits the TOF bit and overflow interrupts until the low byte (TMODL) is written. Reset sets the TIM counter modulo registers.

Address: \$0023	TMODH							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:								
Reset:	1	1	1	1	1	1	1	1

Address: \$0024	TMODL							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:								
Reset:	1	1	1	1	1	1	1	1

**Figure 14-6. TIM Counter Modulo Registers (TMODH:TMODL)**

#### NOTE

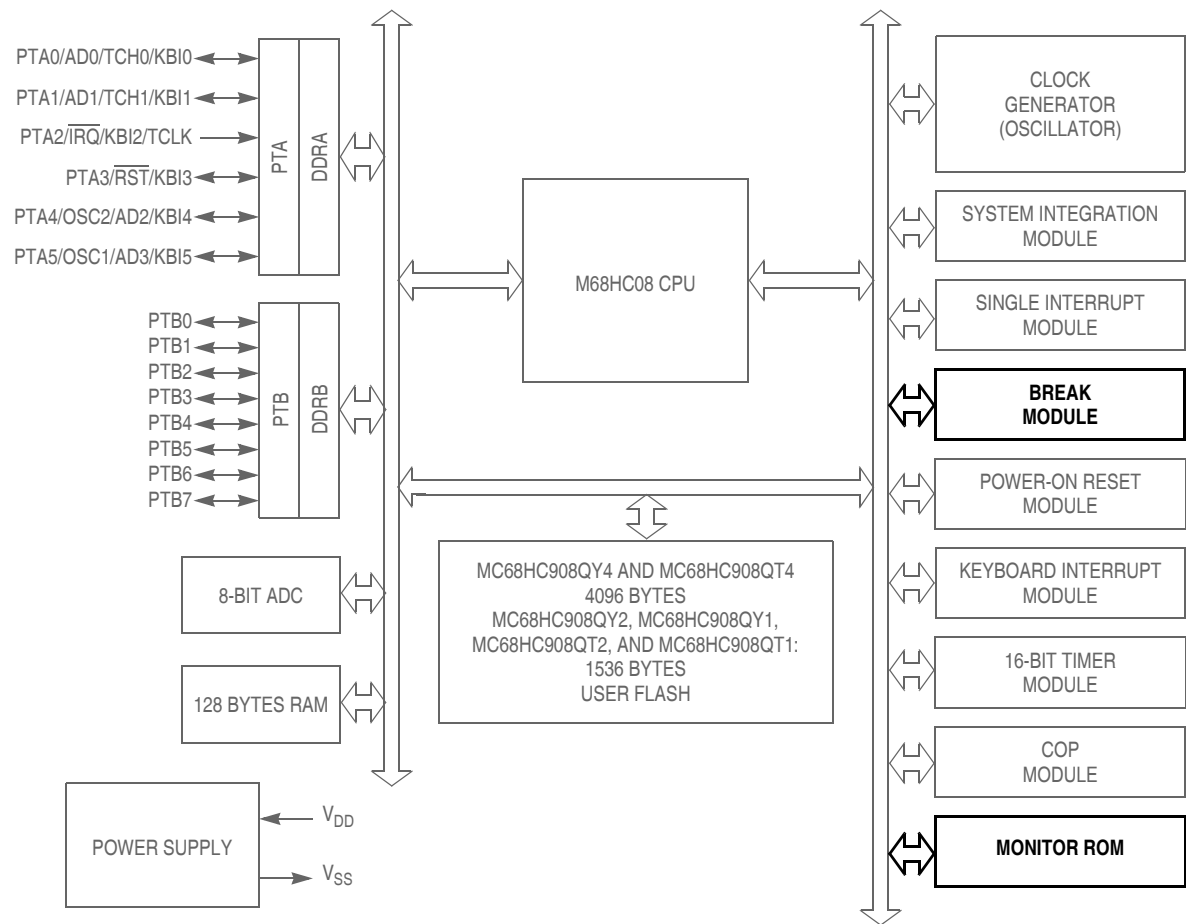
*Reset the TIM counter before writing to the TIM counter modulo registers.*

### 14.9.4 TIM Channel Status and Control Registers

Each of the TIM channel status and control registers does the following:

- Flags input captures and output compares
- Enables input capture and output compare interrupts
- Selects input capture, output compare, or PWM operation
- Selects high, low, or toggling output on output compare
- Selects rising edge, falling edge, or any edge as the active input capture trigger
- Selects output toggling on TIM overflow
- Selects 0% and 100% PWM duty cycle
- Selects buffered or unbuffered output compare/PWM operation





$\overline{RST}$ ,  $\overline{IRQ}$ : Pins have internal (about 30K Ohms) pull up  
 PTA[0:5]: High current sink and source capability  
 PTA[0:5]: Pins have programmable keyboard interrupt and pull up  
 PTB[0:7]: Not available on 8-pin devices – MC68HC908QT1, MC68HC908QT2, and MC68HC908QT4 (see note in [12.1 Introduction](#))  
 ADC: Not available on the MC68HC908QY1 and MC68HC908QT1

Figure 15-1. Block Diagram Highlighting BRK and MON Blocks

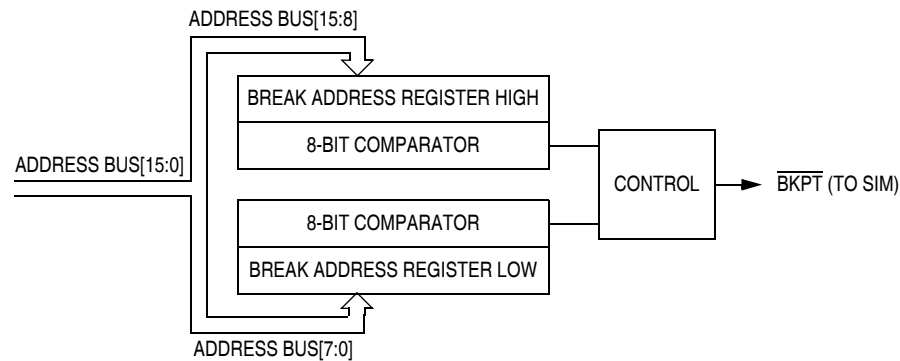


Figure 15-2. Break Module Block Diagram

### 15.3.2 Security

A security feature discourages unauthorized reading of FLASH locations while in monitor mode. The host can bypass the security feature at monitor mode entry by sending eight security bytes that match the bytes at locations \$FFF6–\$FFFD. Locations \$FFF6–\$FFFD contain user-defined data.

#### NOTE

*Do not leave locations \$FFF6–\$FFFD blank. For security reasons, program locations \$FFF6–\$FFFD even if they are not used for vectors.*

During monitor mode entry, the MCU waits after the power-on reset for the host to send the eight security bytes on pin PTA0. If the received bytes match those at locations \$FFF6–\$FFFD, the host bypasses the security feature and can read all FLASH locations and execute code from FLASH. Security remains bypassed until a power-on reset occurs. If the reset was not a power-on reset, security remains bypassed and security code entry is not required. See [Figure 15-18](#).

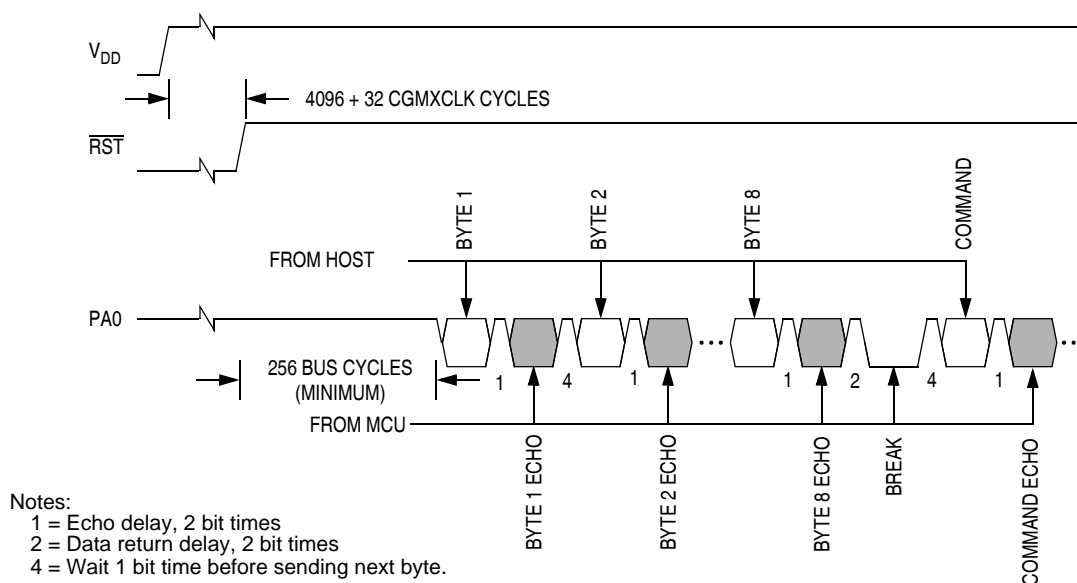
Upon power-on reset, if the received bytes of the security code do not match the data at locations \$FFF6–\$FFFD, the host fails to bypass the security feature. The MCU remains in monitor mode, but reading a FLASH location returns an invalid value and trying to execute code from FLASH causes an illegal address reset. After receiving the eight security bytes from the host, the MCU transmits a break character, signifying that it is ready to receive a command.

#### NOTE

*The MCU does not transmit a break character until after the host sends the eight security bytes.*

To determine whether the security code entered is correct, check to see if bit 6 of RAM address \$80 is set. If it is, then the correct security code has been entered and FLASH can be accessed.

If the security sequence fails, the device should be reset by a power-on reset and brought up in monitor mode to attempt another entry. After failing the security sequence, the FLASH module can also be mass erased by executing an erase routine that was downloaded into internal RAM. The mass erase operation clears the security code locations so that all eight security bytes become \$FF (blank).



**Figure 15-18. Monitor Mode Entry Timing**

# 16.5 5-V DC Electrical Characteristics

Characteristic <sup>(1)</sup>	Symbol	Min	Typ <sup>(2)</sup>	Max	Unit
Output high voltage $I_{Load} = -2.0$ mA, all I/O pins $I_{Load} = -10.0$ mA, all I/O pins $I_{Load} = -15.0$ mA, PTA0, PTA1, PTA3–PTA5 only	$V_{OH}$	$V_{DD}-0.4$ $V_{DD}-1.5$ $V_{DD}-0.8$	— — —	— — —	V
Maximum combined $I_{OH}$ (all I/O pins)	$I_{OHT}$	—	—	50	mA
Output low voltage $I_{Load} = 1.6$ mA, all I/O pins $I_{Load} = 10.0$ mA, all I/O pins $I_{Load} = 15.0$ mA, PTA0, PTA1, PTA3–PTA5 only	$V_{OL}$	— — —	— — —	0.4 1.5 0.8	V
Maximum combined $I_{OL}$ (all I/O pins)	$I_{OLT}$	—	—	50	mA
Input high voltage PTA0–PTA5, PTB0–PTB7	$V_{IH}$	$0.7 \times V_{DD}$	—	$V_{DD}$	V
Input low voltage PTA0–PTA5, PTB0–PTB7	$V_{IL}$	$V_{SS}$	—	$0.3 \times V_{DD}$	V
Input hysteresis	$V_{HYS}$	$0.06 \times V_{DD}$	—	—	V
DC injection current, all ports	$I_{INJ}$	–2	—	+2	mA
Total dc current injection (sum of all I/O)	$I_{INJTOT}$	–25	—	+25	mA
Ports Hi-Z leakage current	$I_{IL}$	–1	$\pm 0.1$	+1	$\mu$ A
Capacitance Ports (as input) Ports (as input)	$C_{IN}$ $C_{OUT}$	— —	— —	12 8	pF
POR rearm voltage <sup>(3)</sup>	$V_{POR}$	0	—	100	mV
POR rise time ramp rate <sup>(4)</sup>	$R_{POR}$	0.035	—	—	V/ms
Monitor mode entry voltage	$V_{TST}$	$V_{DD} + 2.5$	—	9.1	V
Pullup resistors <sup>(5)</sup> PTA0–PTA5, PTB0–PTB7	$R_{PU}$	16	26	36	k $\Omega$
Low-voltage inhibit reset, trip falling voltage	$V_{TRIPF}$	3.90	4.20	4.50	V
Low-voltage inhibit reset, trip rising voltage	$V_{TRIPR}$	4.00	4.30	4.60	V
Low-voltage inhibit reset/recover hysteresis	$V_{HYS}$	—	100	—	mV

1.  $V_{DD} = 4.5$  to  $5.5$  Vdc,  $V_{SS} = 0$  Vdc,  $T_A = T_L$  to  $T_H$ , unless otherwise noted.

2. Typical values reflect average measurements at midpoint of voltage range, 25°C only.

3. Maximum is highest voltage that POR is guaranteed.

4. If minimum  $V_{DD}$  is not reached before the internal POR reset is released, the LVI will hold the part in reset until minimum  $V_{DD}$  is reached.

5.  $R_{PU}$  is measured at  $V_{DD} = 5.0$  V.

# 16.15 Timer Interface Module Characteristics

Characteristic	Symbol	Min	Max	Unit
Timer input capture pulse width	$t_{TH}, t_{TL}$	2	—	$t_{cyc}$
Timer input capture period	$t_{TLTL}$	Note <sup>(1)</sup>	—	$t_{cyc}$
Timer input clock pulse width	$t_{TCL}, t_{TCH}$	$t_{cyc} + 5$	—	ns

1. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1  $t_{cyc}$ .

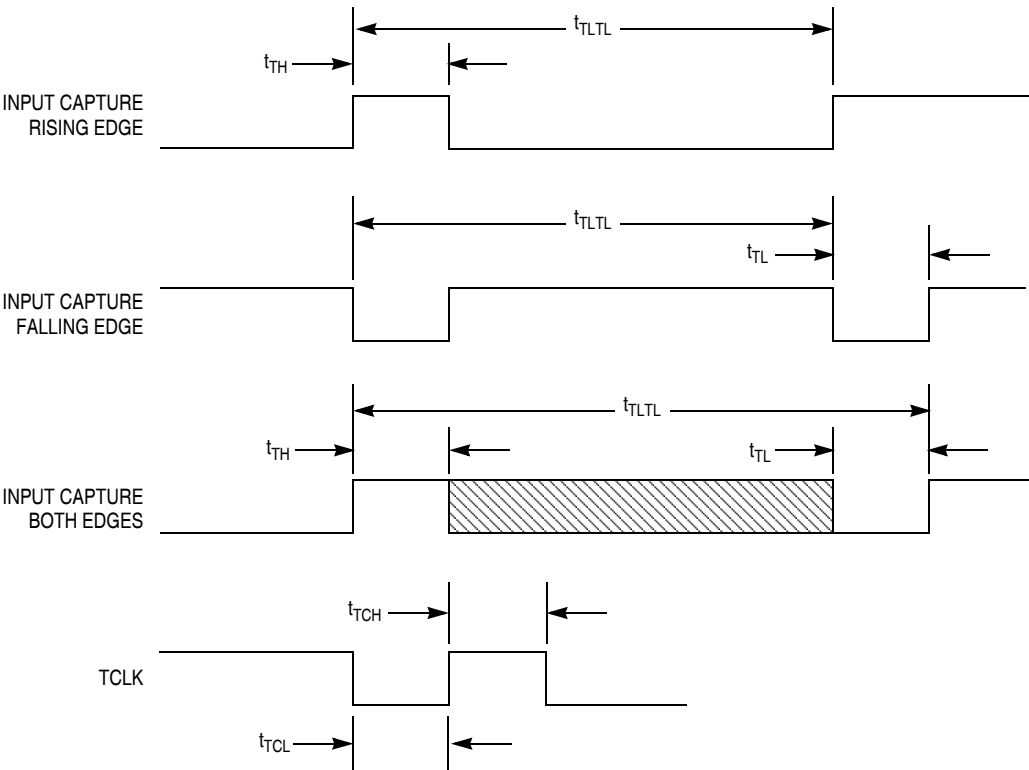


Figure 16-11. Timer Input Timing

