

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	<a href="https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mc68hc908lb8mdwe">https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mc68hc908lb8mdwe</a>



## Chapter 17

### System Integration Module (SIM)

17.1	Introduction	171
17.2	SIM Bus Clock Control and Generation	173
17.2.1	Bus Timing	173
17.2.2	Clock Start-Up from POR	173
17.2.3	Clocks in Stop Mode and Wait Mode	173
17.3	Reset and System Initialization	174
17.3.1	External Pin Reset	174
17.3.2	Active Resets from Internal Sources	174
17.3.2.1	Power-On Reset	175
17.3.2.2	Computer Operating Properly (COP) Reset	176
17.3.2.3	Illegal Opcode Reset	176
17.3.2.4	Illegal Address Reset	176
17.3.2.5	Low-Voltage Inhibit (LVI) Reset	176
17.3.2.6	Monitor Mode Entry Module Reset (MODRST)	177
17.4	SIM Counter	177
17.4.1	SIM Counter During Power-On Reset	177
17.4.2	SIM Counter During Stop Mode Recovery	177
17.4.3	SIM Counter and Reset States	177
17.5	Exception Control	177
17.5.1	Interrupts	177
17.5.1.1	Hardware Interrupts	178
17.5.1.2	SWI Instruction	180
17.5.2	Reset	180
17.5.3	Break Interrupts	180
17.5.4	Status Flag Protection in Break Mode	181
17.6	Low-Power Modes	181
17.6.1	Wait Mode	181
17.6.2	Stop Mode	182
17.7	SIM Registers	183
17.7.1	Break Status Register	183
17.7.2	SIM Reset Status Register	184
17.7.3	Break Flag Control Register	185

## Chapter 18

### Timer Interface Module (TIM)

18.1	Introduction	187
18.2	Features	188
18.3	Functional Description	188
18.3.1	TIM Counter Prescaler	190
18.3.2	Input Capture	190
18.3.3	Output Compare	190
18.3.3.1	Unbuffered Output Compare	190
18.3.3.2	Buffered Output Compare	191
18.3.4	Pulse Width Modulation (PWM)	191
18.3.4.1	Unbuffered PWM Signal Generation	192

the input signal. The ADC overrides the port I/O logic by forcing that pin as input to the ADC. The remaining ADC channels/port pins are controlled by the port I/O logic and can be used as general-purpose I/O. Writes to the port register or data direction register (DDR) will not have any effect on the port pin that is selected by the ADC. Read of a port pin in use by the ADC will return a logic 0. If the DDR bit is at 1, the value in the port data latch is read.

### 3.3.2 Voltage Conversion

When the input voltage to the ADC equals  $V_{REFH}$ , the ADC converts the signal to \$FF (full scale). If the input voltage equals  $V_{REFL}$ , the ADC converts it to \$00. Input voltages between  $V_{REFH}$  and  $V_{REFL}$  are a straight-line linear conversion.

$V_{REFH}$  and  $V_{REFL}$  are internally connected to  $V_{DD}$  and  $V_{SS}$  respectively.

### 3.3.3 Conversion Time

Conversion starts after a write to the ADC status and control register (ADSCR). One conversion will take between 16 and 17 ADC clock cycles. The ADIVx bit should be set to provide a 1-MHz ADC clock frequency.

$$\text{Conversion time} = \frac{16 \text{ to } 17 \text{ ADC cycles}}{\text{ADC frequency}}$$

$$\text{Number of bus cycles} = \text{conversion time} \times \text{bus frequency}$$

### 3.3.4 Conversion

In continuous conversion mode, the ADC data register will be filled with new data after each conversion. Data from the previous conversion will be overwritten whether that data has been read or not. Conversions will continue until the ADCO bit is cleared. The COCO bit is set after the first conversion and will stay set until the next write of the ADC status and control register or the next read of the ADC data register.

In single conversion mode, conversion begins with a write to the ADSCR. Only one conversion occurs between writes to the ADSCR.

### 3.3.5 Accuracy and Precision

The conversion process is monotonic and has no missing codes.

## 3.4 Monotonicity

The conversion process is monotonic and has no missing codes.

## 3.5 Interrupts

When the AIEN bit is set, the ADC module is capable of generating CPU interrupts after each ADC conversion. A CPU interrupt is generated if the COCO bit is at 0. The COCO bit is not used as a conversion complete flag when interrupts are enabled.

The COP counter is a free-running 6-bit counter preceded by the 12-bit system integration module (SIM) counter. If not cleared by software, the COP counter overflows and generates an asynchronous reset after  $2^{18} - 2^4$  or  $2^{13} - 2^4$  BUSCLKX4 cycles; depending on the state of the COP rate select bit, COPRS, in configuration register 1. With a  $2^{18} - 2^4$  BUSCLKX4 cycle overflow option, using the internal clock to produce bus speed of 4 MHz gives a COP timeout period of 16.383 ms. Writing any value to location \$FFFF before an overflow occurs prevents a COP reset by clearing the COP counter and stages 12–5 of the SIM counter.

### NOTE

*Service the COP immediately after reset and before entering or after exiting stop mode to guarantee the maximum time before the first COP counter overflow.*

A COP reset pulls the  $\overline{\text{RST}}$  pin low for  $32 \times \text{BUSCLKX4}$  cycles and sets the COP bit in the reset status register (RSR). See [17.7.2 SIM Reset Status Register](#).

### NOTE

*Place COP clearing instructions in the main program and not in an interrupt subroutine. Such an interrupt subroutine could keep the COP from generating a reset even while the main program is not working properly.*

## 6.3 I/O Signals

The following paragraphs describe the signals shown in [Figure 6-1](#).

### 6.3.1 BUSCLKX4

BUSCLKX4 is the oscillator output signal. BUSCLKX4 frequency is equal to the crystal frequency, the internal oscillator frequency, or the RC oscillator frequency.

### 6.3.2 COPCTL Write

Writing any value to the COP control register (COPCTL) (see [6.4 COP Control Register](#)) clears the COP counter and clears bits 12–5 of the SIM counter. Reading the COP control register returns the low byte of the reset vector.

### 6.3.3 Power-On Reset

The power-on reset (POR) circuit in the SIM clears the SIM counter  $4096 \times \text{BUSCLKX4}$  cycles after power up.

### 6.3.4 Internal Reset

An internal reset clears the SIM counter and the COP counter.

### 6.3.5 Reset Vector Fetch

A reset vector fetch occurs when the vector address appears on the data bus. A reset vector fetch clears the SIM counter.

### 6.3.6 COPD (COP Disable)

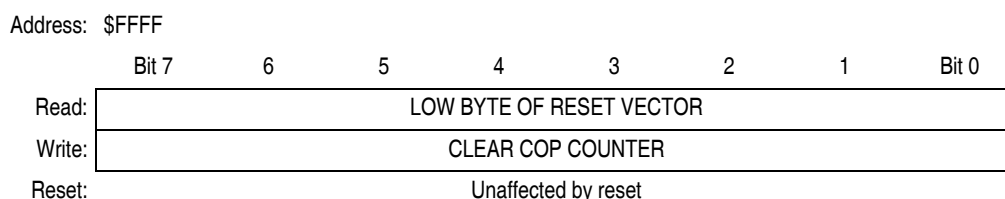
The COPD signal reflects the state of the COP disable bit (COPD) in the configuration register 1 (CONFIG1). See [Chapter 5 Configuration Register \(CONFIG\)](#).

### 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.



**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.

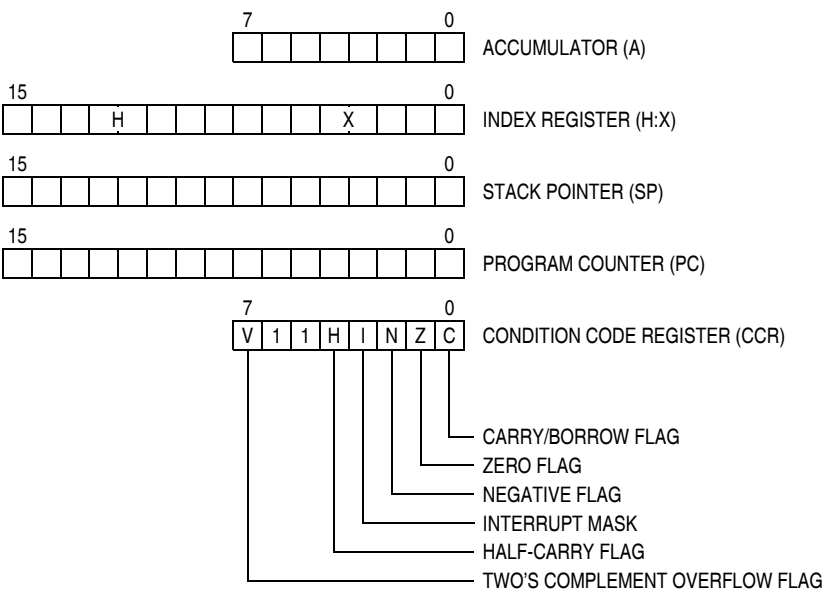


Figure 7-1. CPU Registers

7.3.1 Accumulator

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and the results of arithmetic/logic operations.

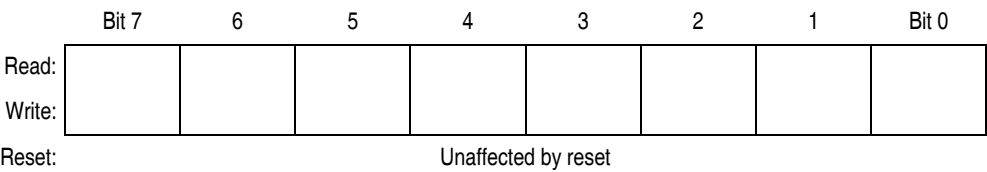


Figure 7-2. Accumulator (A)

7.3.2 Index Register

The 16-bit index register allows indexed addressing of a 64-Kbyte memory space. H is the upper byte of the index register, and X is the lower byte. H:X is the concatenated 16-bit index register.

In the indexed addressing modes, the CPU uses the contents of the index register to determine the conditional address of the operand.

The index register can serve also as a temporary data storage location.

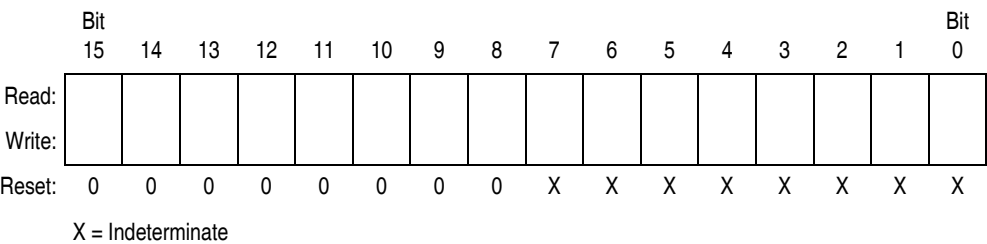
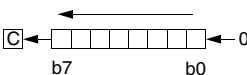


Figure 7-3. Index Register (H:X)

Table 7-1. Instruction Set Summary (Sheet 4 of 7)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
DEC <i>opr</i> DECA DECX DEC <i>opr</i> ,X DEC ,X DEC <i>opr</i> ,SP	Decrement	$M \leftarrow (M) - 1$ $A \leftarrow (A) - 1$ $X \leftarrow (X) - 1$ $M \leftarrow (M) - 1$ $M \leftarrow (M) - 1$ $M \leftarrow (M) - 1$	†	–	–	†	†	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff	4 1 1 4 3 5
DIV	Divide	$A \leftarrow (H:A)/(X)$ $H \leftarrow \text{Remainder}$	–	–	–	–	†	†	INH	52		7
EOR # <i>opr</i> EOR <i>opr</i> EOR <i>opr</i> EOR <i>opr</i> ,X EOR <i>opr</i> ,X EOR ,X EOR <i>opr</i> ,SP EOR <i>opr</i> ,SP	Exclusive OR M with A	$A \leftarrow (A \oplus M)$	0	–	–	†	†	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A8 B8 C8 D8 E8 F8 9EE8 9ED8	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
INC <i>opr</i> INCA INCX INC <i>opr</i> ,X INC ,X INC <i>opr</i> ,SP	Increment	$M \leftarrow (M) + 1$ $A \leftarrow (A) + 1$ $X \leftarrow (X) + 1$ $M \leftarrow (M) + 1$ $M \leftarrow (M) + 1$ $M \leftarrow (M) + 1$	†	–	–	†	†	–	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff	4 1 1 4 3 5
JMP <i>opr</i> JMP <i>opr</i> JMP <i>opr</i> ,X JMP <i>opr</i> ,X JMP ,X	Jump	$PC \leftarrow \text{Jump Address}$	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff	2 3 4 3 2
JSR <i>opr</i> JSR <i>opr</i> JSR <i>opr</i> ,X JSR <i>opr</i> ,X JSR ,X	Jump to Subroutine	$PC \leftarrow (PC) + n$ ( $n = 1, 2, \text{ or } 3$ ) Push (PCL); $SP \leftarrow (SP) - 1$ Push (PCH); $SP \leftarrow (SP) - 1$ $PC \leftarrow \text{Unconditional Address}$	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	4 5 6 5 4
LDA # <i>opr</i> LDA <i>opr</i> LDA <i>opr</i> LDA <i>opr</i> ,X LDA <i>opr</i> ,X LDA ,X LDA <i>opr</i> ,SP LDA <i>opr</i> ,SP	Load A from M	$A \leftarrow (M)$	0	–	–	†	†	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A6 B6 C6 D6 E6 F6 9EE6 9ED6	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LDHX # <i>opr</i> LDHX <i>opr</i>	Load H:X from M	$H:X \leftarrow (M:M + 1)$	0	–	–	†	†	–	IMM DIR	45 55	ii jj dd	3 4
LDX # <i>opr</i> LDX <i>opr</i> LDX <i>opr</i> LDX <i>opr</i> ,X LDX <i>opr</i> ,X LDX ,X LDX <i>opr</i> ,SP LDX <i>opr</i> ,SP	Load X from M	$X \leftarrow (M)$	0	–	–	†	†	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	AE BE CE DE EE FE 9EEE 9EDE	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LSL <i>opr</i> LSLA LSLX LSL <i>opr</i> ,X LSL ,X LSL <i>opr</i> ,SP	Logical Shift Left (Same as ASL)		†	–	–	†	†	†	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5



## High Resolution PWM (HRP)

Read:	P2	P1	P0	STEP4	STEP3	STEP2	STEP1	STEP0
Write:								
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.

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

Address:	\$0056							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
Write:								
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:

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

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:	TB15	TB14	TB13	TB12	TB11	TB10	TB9	TB8
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
Write:								
Reset:	0	0	0	0	0	0	0	0

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



### 13.3.1.1 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 will allow the internal clock frequency value fit in a  $\pm 5\%$  range around 16 MHz.

The oscillator will be trimmed at the factory. The trimming value will be provided in a known FLASH location, \$FFC0. So that the user would be able to copy this byte from the FLASH to the OSCTRIM register right at the beginning of assembly code.

Reset loads OSCTRIM with a default value of \$80.

### 13.3.1.2 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 PTC1 (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 [Table 13-2](#). 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 ms.
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).*

### 13.3.2 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 C pullup enable register will be clear to enable PTC1 I/O functions on the pin.

**Table 13-2. Oscillator Modes**

OSCOPT1	OSCOPT0	Oscillator Modes
0	0	Internal Oscillator
0	1	External Oscillator
1	0	External RC
1	1	External Crystal

## 13.8 Input/Output (I/O) Registers

The oscillator module contains these two registers:

1. Oscillator status register (OSCSTAT)
2. Oscillator trim register (OSCTRIM)

### 13.8.1 Oscillator Status Register

The oscillator status register (OSCSTAT) contains the bits for switching from internal to external clock sources.

Address: \$0036

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	R	R	R	R	R	ECGON	ECGST
Write:								
Reset:	0	0	0	0	0	0	0	0

R = Reserved

**Figure 13-4. Oscillator Status Register (OSCSTAT)**

#### ECGON — External Clock Generator On Bit

This read/write bit enables external clock generator, so that the switching process can be initiated. This bit is forced low during reset. This bit is ignored in monitor mode when the internal oscillator is bypassed.

- 1 = External clock generator enabled
- 0 = External clock generator disabled

### 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:		PTA6PUE	PTA5PUE	PTA4PUE	PTA3PUE	PTA2PUE	PTA1PUE	PTA0PUE
Write:								
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:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Write:								
Reset:	Unaffected by reset							

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

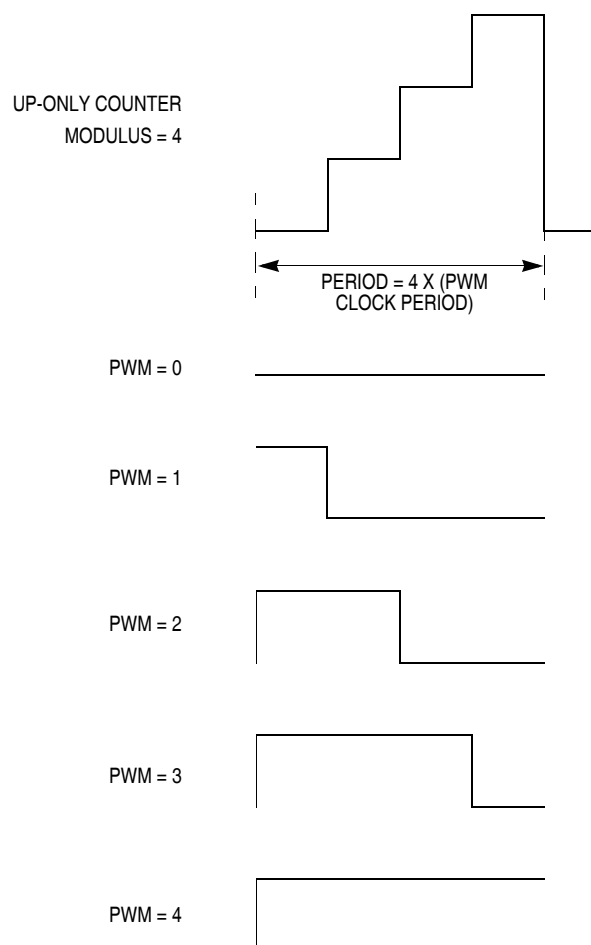
#### PTB7–PTB0 — Port B Data Bits

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.



in [Figure 15-4](#). Again, the timer modulus register is used to determine the maximum count. The PWM period will equal:

$(\text{timer modulus}) \times (\text{PWM clock period})$



**Figure 15-4. Edge-Aligned PWM (Positive Polarity)**

### 15.3.2 Prescaler

To permit lower PWM frequencies, a prescaler is provided which will divide the PWM clock frequency by 1, 2, 4, or 8. [Table 15-1](#) shows how setting the prescaler bits in PWM control register 2 affects the PWM clock frequency. This prescaler is buffered and will not be used by the PWM generator until the LDOK bit is set and a new PWM reload cycle begins.

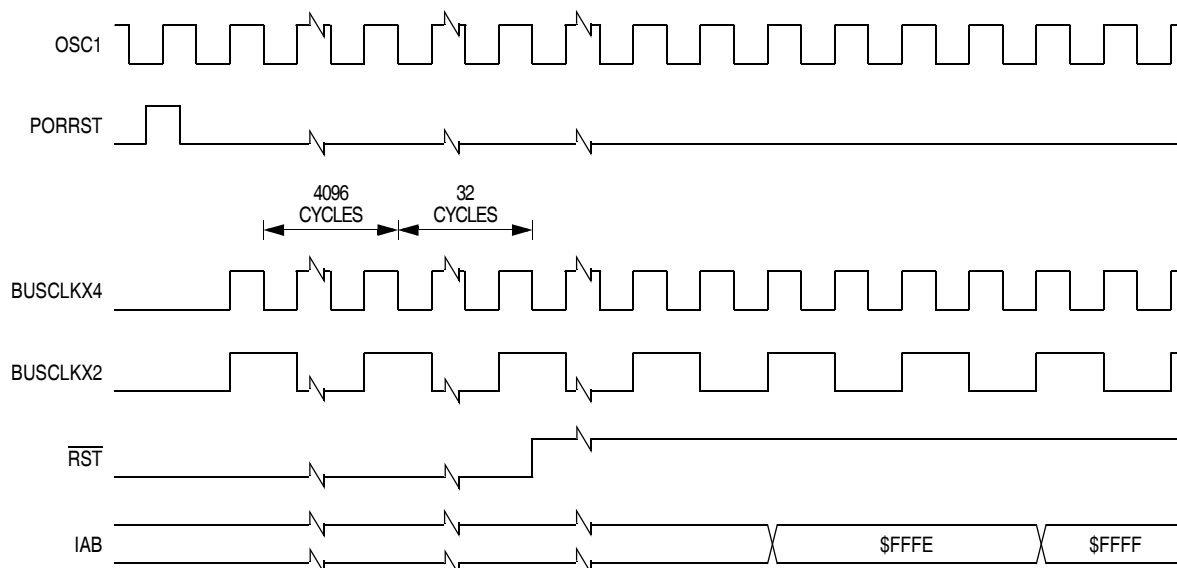


Figure 17-7. POR Recovery

### 17.3.2.2 Computer Operating Properly (COP) Reset

An input to the SIM is reserved for the COP reset signal. The overflow of the COP counter causes an internal reset and sets the COP bit in the SIM reset status register (SRSR). The SIM actively pulls down the  $\overline{\text{RST}}$  pin for all internal reset sources.

The COP module is disabled if the  $\overline{\text{IRQ}}$  pin is held at  $V_{\text{TST}}$  while the MCU is in monitor mode. The COP module can be disabled only through combinational logic conditioned with the high voltage signal on the  $\overline{\text{IRQ}}$  pin. This prevents the COP from becoming disabled as a result of external noise.

### 17.3.2.3 Illegal Opcode Reset

The SIM decodes signals from the CPU to detect illegal instructions. An illegal instruction sets the ILOP bit in the SIM reset status register (SRSR) and causes a reset.

If the stop enable bit, STOP, in the mask option register is 0, the SIM treats the STOP instruction as an illegal opcode and causes an illegal opcode reset. The SIM actively pulls down the  $\overline{\text{RST}}$  pin for all internal reset sources.

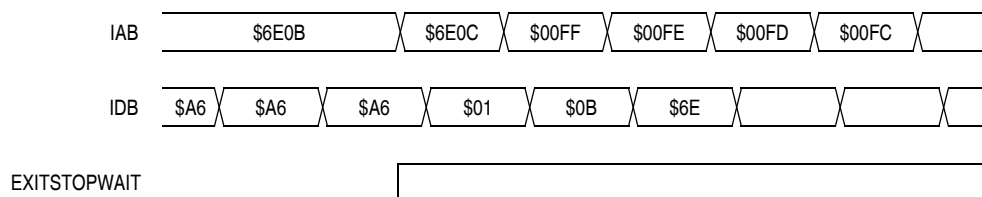
### 17.3.2.4 Illegal Address Reset

An opcode fetch from an unmapped address generates an illegal address reset. The SIM verifies that the CPU is fetching an opcode prior to asserting the ILAD bit in the SIM reset status register (SRSR) and resetting the MCU. A data fetch from an unmapped address does not generate a reset. The SIM actively pulls down the  $\overline{\text{RST}}$  pin for all internal reset sources.

### 17.3.2.5 Low-Voltage Inhibit (LVI) Reset

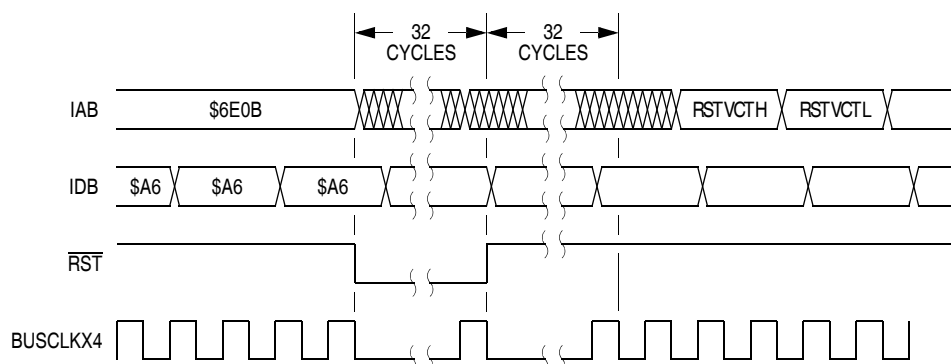
The low-voltage inhibit module (LVI) asserts its output to the SIM when the  $V_{\text{DD}}$  voltage falls to the  $\text{LVI}_{\text{TRIPF}}$  voltage. The LVI bit in the SIM reset status register (SRSR) is set, and the external reset pin ( $\overline{\text{RST}}$ ) is held low while the SIM counter counts out 4096 + 32 BUSCLKX4 cycles. Thirty-two BUSCLKX4





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.*

- 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.

### 18.3.3.2 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 control the output are the ones written to last. TSC0 controls and monitors the buffered output compare function, and TIM channel 1 status and control register (TSC1) is unused.

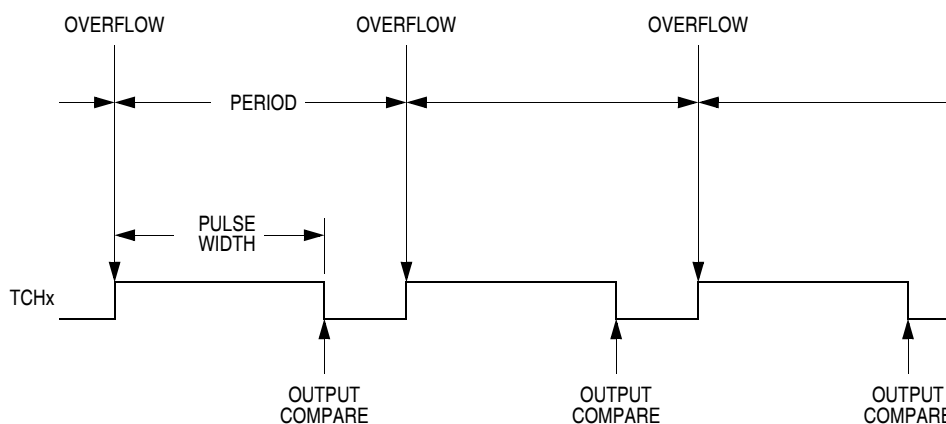
#### NOTE

*In buffered output compare operation, do not write new output compare values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered output compares.*

### 18.3.4 Pulse Width Modulation (PWM)

By using the toggle-on-overflow feature with an output compare channel, the TIM can generate a PWM signal. The value in the TIM counter modulo registers determines the period of the PWM signal. The channel pin toggles when the counter reaches the value in the TIM counter modulo registers. The time between overflows is the period of the PWM signal.

As [Figure 18-4](#) shows, the output compare value in the TIM channel registers determines the pulse width of the PWM signal. The time between overflow and output compare is the pulse width. Program the TIM to clear the channel pin on output compare if the state of the PWM pulse is logic 1. Program the TIM to set the pin if the state of the PWM pulse is logic 0.



**Figure 18-4. PWM Period and Pulse Width**

The value in the TIM counter modulo registers and the selected prescaler output determines the frequency of the PWM output. The frequency of an 8-bit PWM signal is variable in 256 increments. Writing \$00FF (255) to the TIM counter modulo registers produces a PWM period of 256 times the internal bus clock period if the prescaler select value is \$000. See [18.8.1 TIM Status and Control Register](#).

The value in the TIM channel registers determines the pulse width of the PWM output. The pulse width of an 8-bit PWM signal is variable in 256 increments. Writing \$0080 (128) to the TIM channel registers produces a duty cycle of 128/256 or 50%.

### 18.3.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in [18.3.4 Pulse Width Modulation \(PWM\)](#). The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change a pulse width value could cause incorrect operation for up to two PWM 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 PWM period. Also, using a TIM overflow interrupt routine to write a new, smaller pulse width 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 PWM pulse width on channel x:

- When changing to a shorter pulse width, 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 pulse. The interrupt routine has until the end of the PWM period to write the new value.
- When changing to a longer pulse width, 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 PWM 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 PWM period.

#### 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 also can cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.*

### 18.3.4.2 Buffered PWM Signal Generation

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

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The TIM channel 0 registers initially control the pulse width on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIM channel registers (0 or 1) that control the pulse width are the ones written to last. TSC0 controls and monitors the buffered PWM function, and TIM channel 1 status and control register (TSC1) is unused.

## 20.10 Comparator Parameters

(Measured over -40°C to +125°C at operating voltage = 5V;  $R_L = 20k\Omega$  unless specified)

Parameter	Minimum	Typical	Maximum	Unit
Input offset current <sup>(1)(2)</sup>	—	—	$\pm 10$	nA
Input offset voltage	—	$\pm 5$	$\pm 15$	mV
Input bias current <sup>(1)(2)</sup>	—	—	$\pm 10$	nA
Common mode voltage range low <sup>(2)</sup>	—	1.2	1.5	V
Common mode voltage range high <sup>(2)</sup>	VDD-2.0	VDD-1.6	—	V
Input resistance <sup>(1)(2)</sup>	10	—	—	M $\Omega$
Input common mode rejection ratio (DC)	55	—	—	dB
Respond Time (0.4V to V <sub>DD</sub> -0.4V swing, $\Delta V_{IN}=100mV$ , $R_L=20k\Omega$ )	—	0.5	—	$\mu s$
DC open loop voltage gain <sup>(2)</sup>	60	—	—	dB
Output Voltage Range ( $I_L = \pm 8mA$ )	Same as PTB7	—	Same as PTB7	V
Output short circuit current	—	Same as PTB7	—	mA
Input capacitance <sup>(2)</sup>	—	—	5	pF
Supply current <sup>(2)(3)</sup>	—	0.5	—	mA

### NOTES:

1. Excludes pad leakage current.
2. These values are from design and are not tested.
3. Supply current measured with  $R_L = 20k\Omega$  at maximum output.

## 20.11 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>	—	$t_{CYC}$

### NOTES:

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

