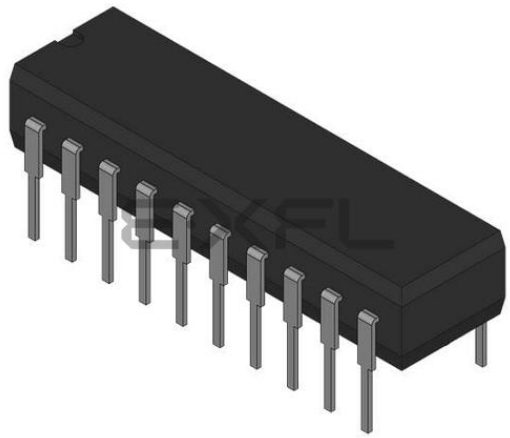

Welcome to E-XFL.COM

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 ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68hc908lb8cpe

Chapter 4
Op Amp/Comparator Module

Chapter 5
Configuration Register (CONFIG)

Chapter 6
Computer Operating Properly (COP) Module

Chapter 19
Development Support

Chapter 20
Electrical Specifications

Chapter 21
Ordering Information and Mechanical Specifications

  - Three shared with op amp/comparator
  - Seven shared with ADC module (AD[0:6])
  - One shared with timer channel 0
  - Two shared with OSC1 and OSC2
  - One shared with reset
  - Seven shared with keyboard interrupt
  - One input-only pin shared with external interrupt (IRQ)
- Available packages:
  - 20-pin small outline integrated chip (SOIC) package
  - 20-pin plastic dual in-line package (PDIP)
- On-chip programming firmware for use with host personal computer which does not require high voltage for entry
- System protection features:
  - Optional computer operating properly (COP) reset
  - Low-voltage reset
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- Low-power design; fully static with stop and wait modes
- Standard low-power modes of operation:
  - Wait mode
  - Stop mode
- Master reset pin and power-on reset (POR)
- 674 bytes of FLASH programming routines read-only memory (ROM)
- Break module (BRK) to allow single breakpoint setting during in-circuit debugging
- Internal pullup on $\overline{\text{RST}}$ pin to reduce customer system cost

Chapter 2
Memory

2.1 Introduction

The CPU08 can address 64 Kbytes of memory space. The memory map, shown in Figure 2-1, includes:

- System registers
- 8192 bytes of user FLASH memory
- 128 bytes of random-access memory (RAM)
- 674 bytes of FLASH programming routines read-only memory (ROM)
- 34 bytes of user-defined vectors

2.2 Unimplemented Memory Locations

Accessing an unimplemented location can cause an illegal address reset. In the memory map (Figure 2-1) and in register figures in this document, unimplemented locations are shaded.

2.3 Reserved Memory Locations

Accessing a reserved location can have unpredictable effects on microcontroller (MCU) operation. In the Figure 2-1 and in register figures in this document, reserved locations are marked with the word Reserved or with the letter R.

2.4 Register Section

Most of the control, status, and data registers are in the zero page area of $0000–$0058. Additional I/O registers have these addresses:

- $FE00; break status register, BSR
- $FE01; SIM reset status register, SRSR
- $FE02; break auxiliary register, BRKAR
- $FE03; break flag control register, BFCR
- $FE04; interrupt status register 1, INT1
- $FE05; interrupt status register 2, INT2
- $FE06; reserved
- $FE07; reserved
- $FE08; FLASH control register, FLCR
- $FE09; break address register high, BRKH
- $FE0A; break address register low, BRKL
- $FE0B; break status and control register, BRKSCR
- $FE0C; LVI status register, LVISR
- $FF7E; FLASH block protect register, FLBPR

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0020	Timer Status and Control Register (TSC) See page 195.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0021	Timer Counter Register High (TCNTH) See page 196.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0022	Timer Counter Register Low (TCNTL) See page 196.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0023	Timer Counter Modulo Register High (TMODH) See page 197.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0024	Timer Counter Modulo Register Low (TMODL) See page 197.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0025	Timer Channel 0 Status and Control Register (TSC0) See page 198.	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0026	Timer Channel 0 Register High (TCH0H) See page 201.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0027	Timer Channel 0 Register Low (TCH0L) See page 201.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0028	Timer Channel 1 Status and Control Register (TSC1) See page 198.	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0029	Timer Channel 1 Register High (TCH1H) See page 201.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$002A	Timer Channel 1 Register Low (TCH1L) See page 201.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$002B ↓ $0029	Unimplemented									

= Unimplemented R = Reserved

Bold = Buffered U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 3 of 8)

Address: $001B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:		KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 9-5. Keyboard Interrupt Enable Register (INTKBIER)

KBIE6–KBIE0 — Keyboard Interrupt Enable Bits

Each of these read/write bits enables the corresponding keyboard interrupt pin to latch interrupt requests. Reset clears the keyboard interrupt enable register.

1 = PTAx pin enabled as keyboard interrupt pin
0 = PTAx pin not enabled as keyboard interrupt pin

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)

11.13 Timer Interface Module (TIM)

11.13.1 Wait Mode

The timer interface module (TIM) remains active in wait mode. Any enabled CPU interrupt request from the TIM can bring the MCU out of wait mode.

If TIM functions are not required during wait mode, reduce power consumption by stopping the TIM before executing the WAIT instruction.

11.13.2 Stop Mode

The TIM is inactive in stop mode. The STOP instruction does not affect register states or the state of the TIM counter. TIM operation resumes when the MCU exits stop mode after an external interrupt.

11.14 Exiting Wait Mode

These events restart the CPU clock and load the program counter with the reset vector or with an interrupt vector:

- External reset — A logic 0 on the $\overline{\text{RST}}$ pin resets the MCU and loads the program counter with the contents of locations: $FFFE and $FFFF.
- External interrupt — A high-to-low transition on an external interrupt pin ($\overline{\text{IRQ}}$ pin) loads the program counter with the contents of locations: $FFFA and $FFFB.
- Break (BRK) interrupt — A break interrupt loads the program counter with the contents of: $FFFC and $FFFD.
- Computer operating properly (COP) module reset — A timeout of the COP counter resets the MCU and loads the program counter with the contents of: $FFFE and $FFFF.
- Low-voltage inhibit (LVI) module reset — A power supply voltage below the V_{TRIPF} voltage resets the MCU and loads the program counter with the contents of locations: $FFFE and $FFFF.
- Keyboard interrupt (KBI) module — A CPU interrupt request from the KBI module loads the program counter with the contents of: $FFE0 and $FFE1.
- Timer interface (TIM) module interrupt — A CPU interrupt request from the TIM loads the program counter with the contents of:
  - $FFF2 and $FFF3; TIM overflow
  - $FFF4 and $FFF5; TIM channel 1
  - $FFF6 and $FFF7; TIM channel 0
- Analog-to-digital converter (ADC) module interrupt — A CPU interrupt request from the ADC loads the program counter with the contents of: $FFDF and $FFDE.
- Pulse-Width Modulator with Fault Input (PWM) — A CPU interrupt request from the PWM load the program counter with the contents of:
  - $FFF1 and $FFF0; FAULT
  - $FFEF and $FFEE; PWMINT
- High Resolution PWM (HRP) — A CPU interrupt request from the HRP loads the program counter with the contents of: $FFED and $FFEC

Address:	$0006							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	DDRC1	DDRC0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-10. Data Direction Register C (DDRC)

DDRC1–DDRC0 — Data Direction Register C Bits

These read/write bits control port C data direction. Reset clears DDRC1–DDRC0, configuring all port C pins as inputs.

1 = Corresponding port C pin configured as output
0 = Corresponding port C pin configured as input

NOTE

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

Figure 14-11 shows the port C I/O logic.

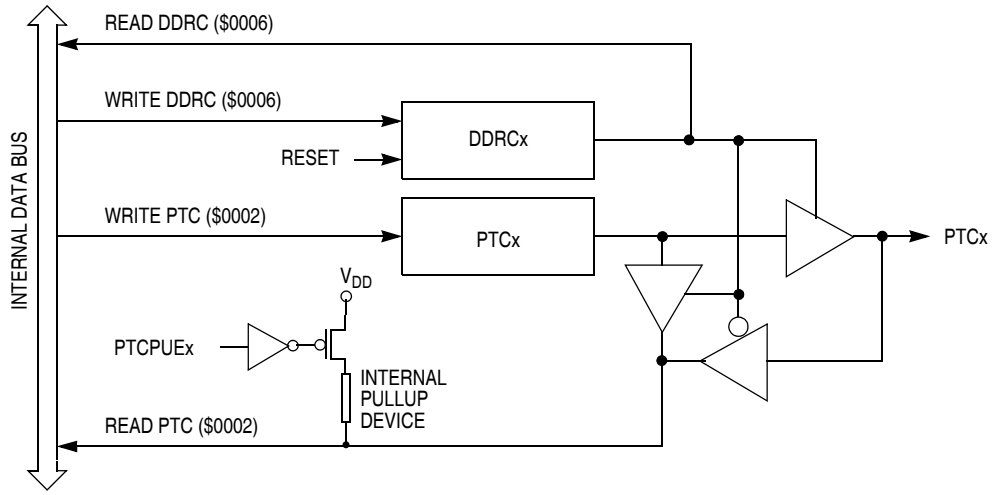

Figure 14-11. Port C I/O Circuit

NOTE

Figure 14-11 does not apply to PTC2.

When bit DDRCx is a 1, reading address $0002 reads the PTCx data latch. When bit DDRCx is a 0, reading address $0002 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 14-3 summarizes the operation of the port C pins.

Chapter 15
Pulse Width Modulator with Fault Input (PWM)

15.1 Introduction

This section describes the pulse-width modulator with fault input (PWM). The MC68HC908LB8 PWM module can generate two independent PWM signals. These PWM signals are edge-aligned. A block diagram of the PWM module is shown in Figure 15-2.

A 12-bit timer PWM counter is common to both channels. PWM resolution is one clock period for edge-aligned operation. The clock period is dependent on the internal operating frequency (BUSCLK) and a programmable prescaler.

The highest resolution for edge-aligned operation is 125 ns (BUSCLK = 8 MHz).

A summary of the PWM registers is shown in Figure 15-3.

15.2 Features

Features of the PWMMC include:

- Two independent PWM signals
- Edge-aligned PWM signals
- PWM signal polarity control
- Programmable fault protection

NOTE

The terms “active” and “inactive” refer to the asserted and negated states of the PWM signals and should not be confused with the high-impedance state of the PWM pins.

Table 15-3. PWM Data Overflow and Underflow Conditions

PWMVALxH:PWMVALxL	Condition	PWM Value Used
$0000–$0FFF	Normal	Per register contents
$1000–$7FFF	Overflow	$FFF
$8000–$FFFF	Underflow	$000

15.4.3 Output Polarity

The output polarity of the PWMs is determined by the POLx bits. Positive polarity means that when the PWM is active, the PWM output is high. Conversely, negative polarity means that when the PWM is active, PWM output is low. See Figure 15-9.

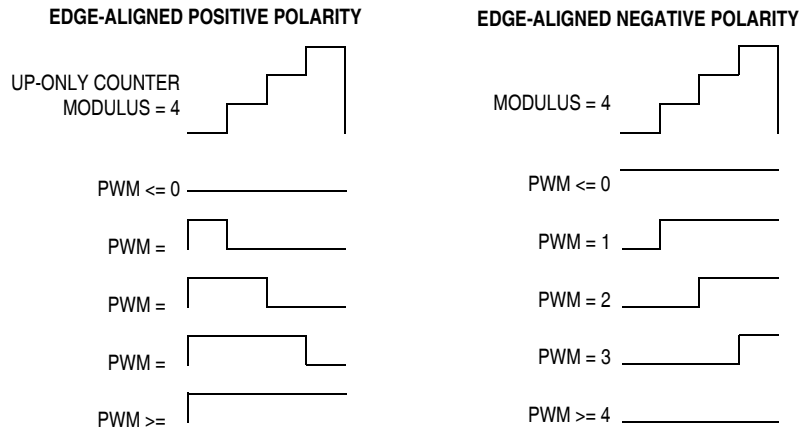

Figure 15-9. PWM Output Polarity

15.5 Fault Protection

Conditions may arise in the external drive circuitry which require that the PWM signals become inactive immediately. Furthermore, it may be desirable to selectively disable PWM(s) solely with software.

One or more PWM pins can be disabled (forced to their inactive state) by applying a logic high to the external fault pin or by writing a logic high to either of the disable bits (DIS0 and DIS1 in PWM control register 1). Figure 15-10 shows the structure of the PWM disabling scheme. While the PWM pins are disabled, they are forced to their inactive state. The PWM generator continues

A fault can also generate a CPU interrupt. The fault pin has its own interrupt vector.

15.5.1 Fault Condition Input Pin

A logic high level on a fault pin disables the PWM(s) determined by the disable map bits (MAPx). The external fault pin is software-configurable to re-enable the PWMs either with the fault pin (automatic

PWM Load Frequency

Frequency at which new PWM parameters get loaded into the PWM. See Figure 15-27.

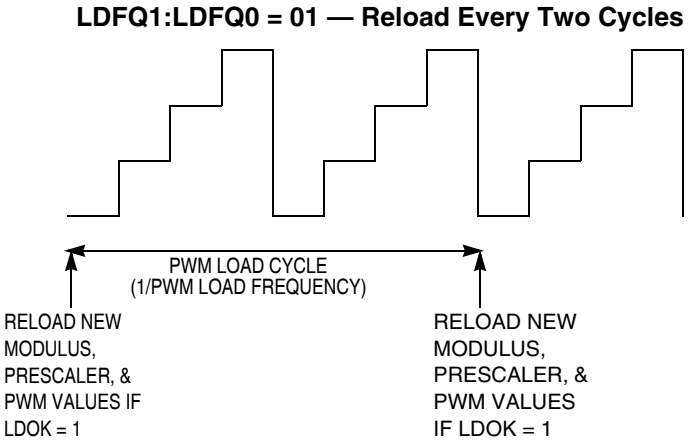

Figure 15-27. PWM Load Cycle/Frequency Definition

In wait mode, the CPU clocks are inactive. The SIM also produces two sets of clocks for other modules. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

17.3 Reset and System Initialization

The MCU has these reset sources:

- Power-on reset module (POR)
- External reset pin ($\overline{RST}$)
- Computer operating properly module (COP)
- Low-voltage inhibit module (LVI)
- Illegal opcode
- Illegal address
- Forced monitor mode entry reset (MODRST)

All of these resets produce the vector $FFFE:$FFFF ($FEFE:$FEFF in monitor mode) and assert the internal reset signal (IRST). IRST causes all registers to be returned to their default values and all modules to be returned to their reset states.

An internal reset clears the SIM counter (see 17.4 SIM Counter), but an external reset does not. Each of the resets sets a corresponding bit in the SIM reset status register (SRSR). See 17.7 SIM Registers.

17.3.1 External Pin Reset

The $\overline{RST}$ pin circuit includes an internal pullup device. Pulling the asynchronous $\overline{RST}$ pin low halts all processing. The PIN bit of the SIM reset status register (SRSR) is set as long as $\overline{RST}$ is held low for a minimum of 67 BUSCLKX4 cycles, assuming that neither the POR nor the LVI was the source of the reset. See Table 17-2 for details. Figure 17-4 shows the relative timing.

Table 17-2. PIN Bit Set Timing

Reset Type	Number of Cycles Required to Set PIN
POR/LVI	4163 (4096 + 64 + 3)
All others	67 (64 + 3)

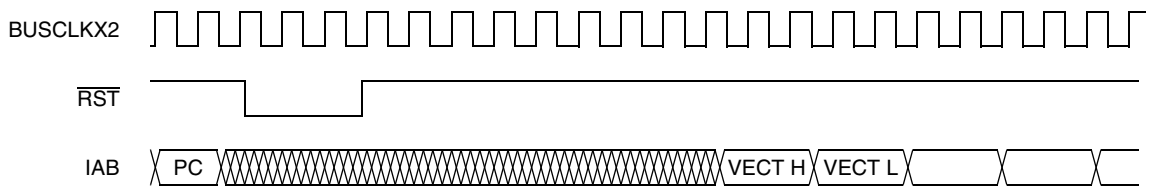

Figure 17-4. External Reset Timing

17.3.2 Active Resets from Internal Sources

All internal reset sources actively pull the $\overline{RST}$ pin low for 32 BUSCLKX4 cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles.

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.

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.

NOTE

In buffered PWM signal generation, do not write new pulse width 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 PWM signals.

18.3.4.3 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 18-2.
   b. Write 1 to the toggle-on-overflow bit, TOVx.
   c. Write 1:0 (to clear output on compare) or 1:1 (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 18-2.

   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.

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 18.8.4 TIM Channel Status and Control Registers.

18.4 Interrupts

The following TIM sources can generate interrupt requests:

18.8 I/O Registers

These I/O registers control and monitor operation of the TIM:

- TIM status and control register (TSC)
- TIM counter registers (TCNTH:TCNTL)
- TIM counter modulo registers (TMODH:TMODL)
- TIM channel status and control registers (TSC0 and TSC1)
- TIM channel registers (TCH0H:TCH0L, TCH1H:TCH1L)

18.8.1 TIM Status and Control Register

The TIM status and control register (TSC):

- Enables TIM overflow interrupts
- Flags TIM overflows
- Stops the TIM counter
- Resets the TIM counter
- Prescales the TIM counter clock

Address: $0020

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
Write:	0			TRST				
Reset:	0	0	1	0	0	0	0	0

= Unimplemented

Figure 18-5. TIM Status and Control Register (TSC)

TOF — TIM Overflow Flag Bit

This read/write flag is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. Clear TOF by reading the TIM status and control register when TOF is set and then writing a 0 to TOF. If another TIM overflow occurs before the clearing sequence is complete, then writing 0 to TOF has no effect. Therefore, a TOF interrupt request cannot be lost due to inadvertent clearing of TOF. Reset clears the TOF bit. Writing a 1 to TOF has no effect.

1 = TIM counter has reached modulo value
0 = TIM counter has not reached modulo value

TOIE — TIM Overflow Interrupt Enable Bit

This read/write bit enables TIM overflow interrupts when the TOF bit becomes set. Reset clears the TOIE bit.

1 = TIM overflow interrupts enabled
0 = TIM overflow interrupts disabled

TSTOP — TIM Stop Bit

This read/write bit stops the TIM counter. Counting resumes when TSTOP is cleared. Reset sets the TSTOP bit, stopping the TIM counter until software clears the TSTOP bit.

1 = TIM counter stopped
0 = TIM counter active

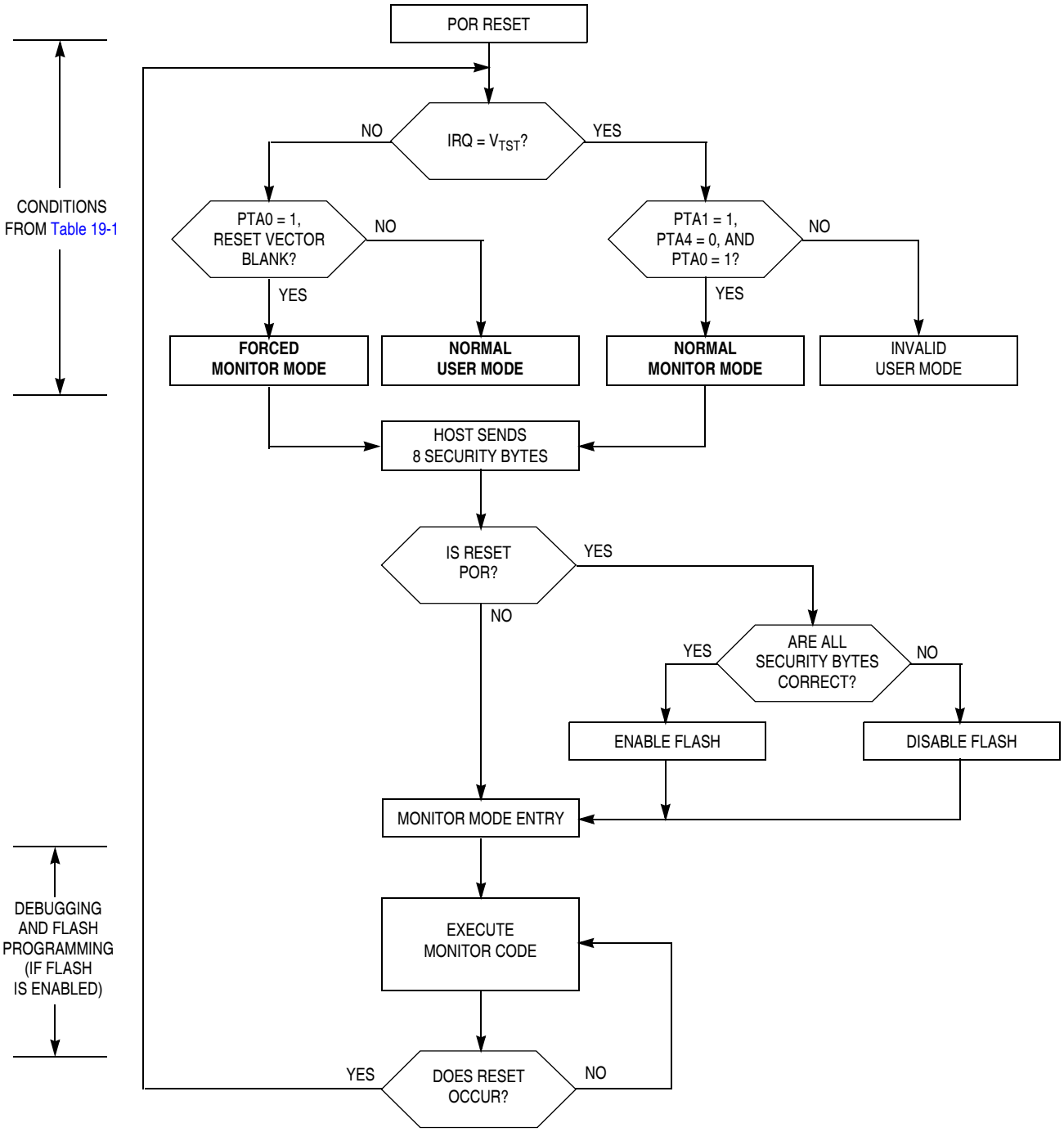

Figure 19-9. Simplified Monitor Mode Entry Flowchart

20.10 Comparator Parameters

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

Parameter	Minimum	Typical	Maximum	Unit
Input offset current(1)(2)	—	—	± 10	nA
Input offset voltage	—	± 5	± 15	mV
Input bias current(1)(2)	—	—	± 10	nA
Common mode voltage range low(2)	—	1.2	1.5	V
Common mode voltage range high(2)	VDD-2.0	VDD-1.6	—	V
Input resistance(1)(2)	10	—	—	MΩ
Input common mode rejection ratio (DC)	55	—	—	dB
Respond Time (0.4V to V_{DD}-0.4V swing, ΔV_{IN}=100mV, R_L=20kΩ)	—	0.5	—	μs
DC open loop voltage gain(2)	60	—	—	dB
Output Voltage Range (I_L= ±8mA)	Same as PTB7	—	Same as PTB7	V
Output short circuit current	—	Same as PTB7	—	mA
Input capacitance(2)	—	—	5	pF
Supply current(2)(3)	—	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 = 20kΩ 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(1)	—	t_{CYC}

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