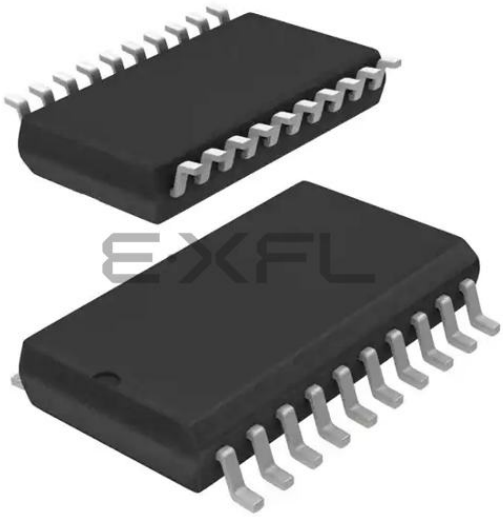


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LVR, POR, PWM
Number of I/O	18
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 7x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908lb8vdwer

Chapter 11
Low-Power Modes

  - 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

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

= Unimplemented R = Reserved

Bold = Buffered U = Unaffected

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

Table 2-1. Vector Addresses

Vector Priority	Address	Vector
Highest	$FFFF	Reset vector (low)
↑	$FFFE	Reset vector (high)
	$FFFD	SWI vector (low)
	$FFFC	SWI vector (high)
	$FFFB	$\overline{\text{IRQ}}$ vector (low)
	$FFFA	$\overline{\text{IRQ}}$ vector (high)
	$FFF9 ↓ $FFF8	Not used
	$FFF7	TIM Channel 0 vector (low)
	$FFF6	TIM Channel 0 vector (high)
	$FFF5	TIM Channel 1 vector (low)
	$FFF4	TIM Channel 1 vector (high)
	$FFF3	TIM overflow vector (low)
	$FFF2	TIM overflow vector (high)
	$FFF1	FAULT (PWM vector) (low)
	$FFF0	FAULT (PWM vector) (high)
	$FFEF	PWMINT (PWM vector) (low)
	$FFEE	PWMINT (PWM vector) (high)
	$FFED	SHTDWN (HRP vector) (low)
	$FFEC	SHTDWN (HRP vector) (high)
	$FFEB ↓ $FFE2	Not used
	$FFE1	Keyboard vector (low)
	$FFE0	Keyboard vector (high)
	$FFDF	ADC conversion complete vector (low)
Lowest	$FFDE	ADC conversion complete vector (high)

2.5 Random-Access Memory (RAM)

Addresses $0080 through $00FF are RAM locations. The location of the stack RAM is programmable. The 16-bit stack pointer allows the stack to be anywhere in the 64-Kbyte memory space.

2.6.7 Wait Mode

Putting the MCU into wait mode while the FLASH is in read mode does not affect the operation of the FLASH memory directly, but there will not be any memory activity since the CPU is inactive.

The WAIT instruction should not be executed while performing a program or erase operation on the FLASH, otherwise the operation will discontinue, and the FLASH will be on standby mode.

2.6.8 Stop Mode

Putting the MCU into stop mode while the FLASH is in read mode does not affect the operation of the FLASH memory directly, but there will not be any memory activity since the CPU is inactive.

The STOP instruction should not be executed while performing a program or erase operation on the FLASH, otherwise the operation will discontinue, and the FLASH will be on standby mode

NOTE

Standby mode is the power saving mode of the FLASH module in which all internal control signals to the FLASH are inactive and the current consumption of the FLASH is at a minimum.

Chapter 4
Op Amp/Comparator Module

4.1 Introduction

This section describes the functionality of the op amp/comparator.

4.2 Features

Features of the op amp/comparator include:

- Software enable/disable
- Op amp and comparator modes for optimized performance
- Shared output pin with ADC input pin and PWM fault pin to allow a op amp/comparator circuit to be inputs to these modules

4.3 Pin Name Conventions

The op amp/comparator shares two input pins and an output pin with the port B input/output (I/O). The full names of the op amp/comparator pins are listed in
Table 4-1. Note that the generic pin names appear in the text that follows.

Table 4-1. Pin Name Conventions

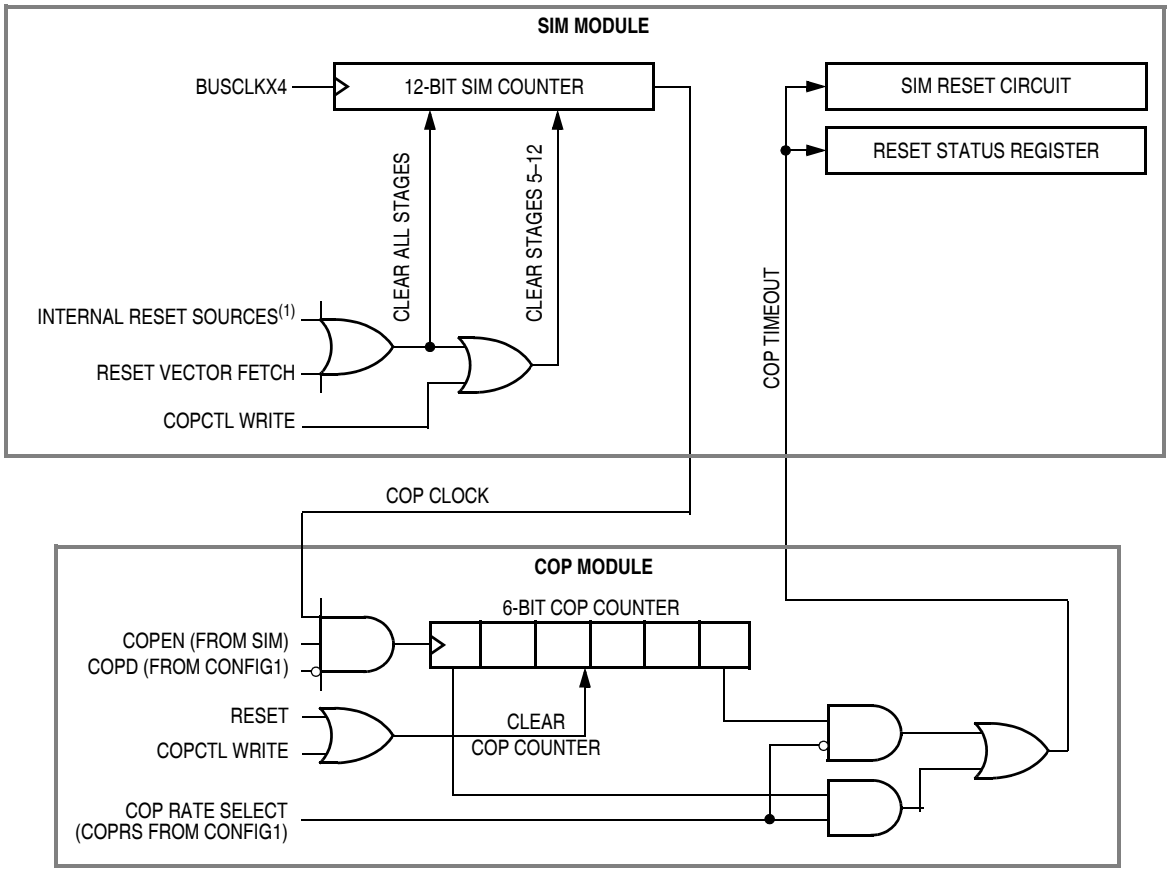
Generic Pin Name	Full Pin Name
V_{OUT}	PTB7/V_{OUT}/ADC6/FAULT
V–	PTB6/V–
V+	PTB5/V+

Chapter 6
Computer Operating Properly (COP) Module

6.1 Introduction

The computer operating properly (COP) module contains a free-running counter that generates a reset if allowed to overflow. The COP module helps software recover from runaway code. Prevent a COP reset by clearing the COP counter periodically. The COP module can be disabled through the COPD bit in the configuration 1 (CONFIG1) register.

6.2 Functional Description



1. See Chapter 17 System Integration Module (SIM) for more details.

Figure 6-1. COP Block Diagram

9.4 Keyboard Initialization

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

To prevent a false interrupt on keyboard initialization:

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

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

Another way to avoid a false interrupt:

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

9.5 Low-Power Modes

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

9.5.1 Wait Mode

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

9.5.2 Stop Mode

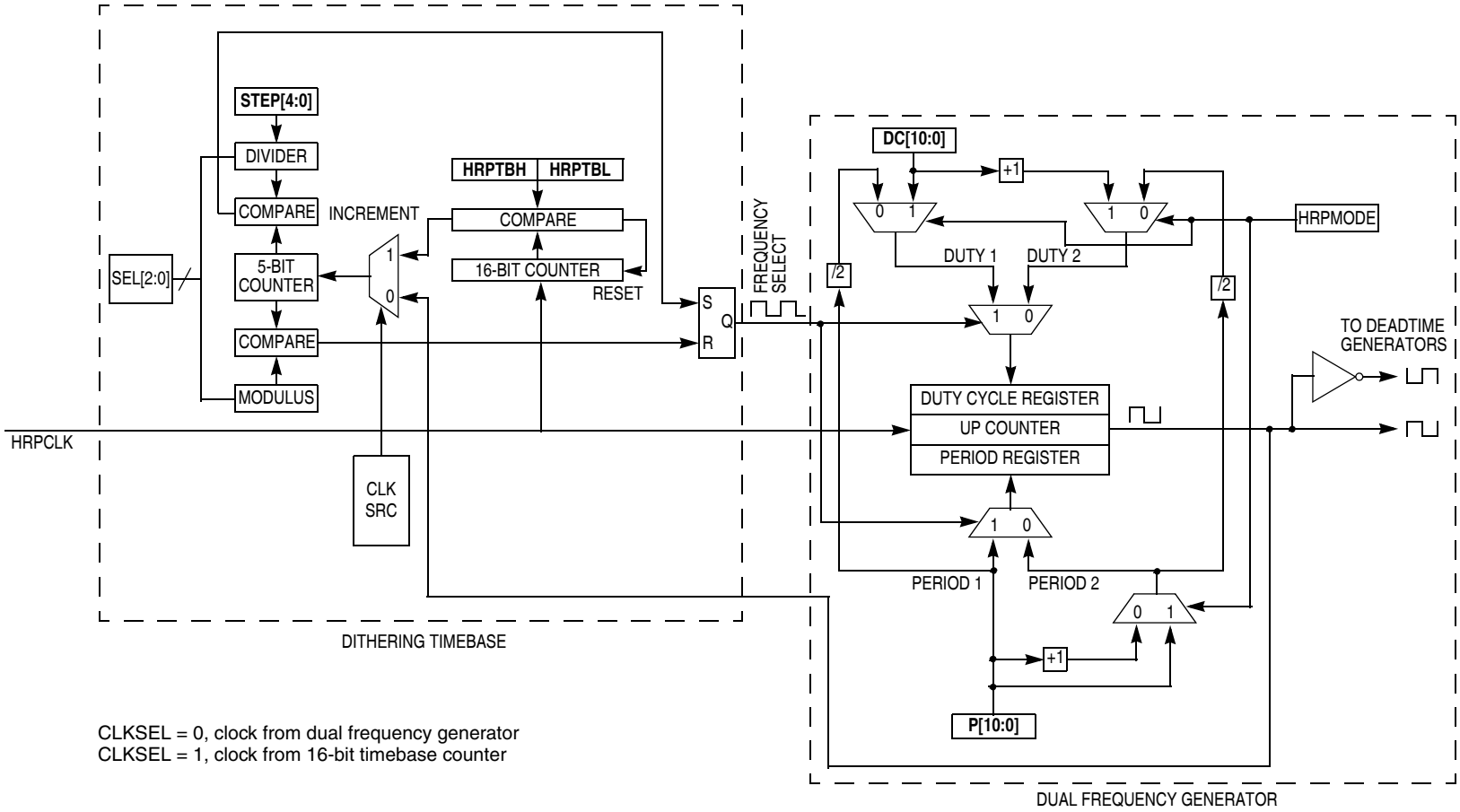
The keyboard module remains active in stop mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of stop mode.

9.6 Keyboard Module During Break Interrupts

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

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

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



CLKSEL = 0, clock from dual frequency generator
CLKSEL = 1, clock from 16-bit timebase counter

Figure 10-6. Dithering Controller and Dual Frequency Generator Block

Address: HRPDCH — $0052 HRPDCL — $0053

	Bit 15	14	13	12	11	10	9	Bit 8
Read: Write:	DC10	DC9	DC8	DC7	DC6	DC5	DC4	DC3
Reset:	0	0	0	0	0	0	0	0
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DC2	DC1	DC0	STEP4	STEP3	STEP2	STEP1	STEP0
Reset:	0	0	0	0	0	0	0	0

Figure 10-11. HRP Duty Cycle Registers (HRPDCH:HRPDCL)

DC[10:0] — 11-Bit Duty Cycle Value

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

10.8.3 HRP Period Registers

The two read/write period registers contain the 16-bit period of the PWM output after dithering. It is split into two parts:

1. 11-bit period value (P[10:0]) used to generate the HRP's output waveforms.
2. 5-bit step value (STEP[4:0]) the lower five bits of HRPPERH:HRPPERL, specifies how much time is spent on the longer period (PERIOD2).

The output period including dithering in variable frequency mode is:

$$\text{Output Period (seconds)} = \frac{P[10:0]}{HRPCLK} + \frac{INT\left(\frac{STEP[4:0]}{2^{SEL[2:0]}}\right)}{\frac{32}{2^{SEL[2:0]}} \text{¥ HRPCLK}} \qquad \textbf{(EQ 10-12)}$$

where $2^{SEL[2:0]}$ is the STEP[4:0] scaling factor.

The output period in variable duty cycle mode does not include dithering. The period value is:

$$\text{Period} = \frac{P[10:0]}{HRPCLK} \qquad \textbf{(EQ 10-13)}$$

Writes to the high byte (HRPPERH) are stored in a latch until the low byte (HRPPERL) is written. Both registers are then updated simultaneously. This prevents glitches in the output period.

Address: HRPPERH — $0054 HRPPERL — $0055

	Bit 15	14	13	12	11	10	9	Bit 8
Read: Write:	P10	P9	P8	P7	P6	P5	P4	P3
Reset:	0	0	0	0	0	0	0	0
	Bit 7	6	5	4	3	2	1	Bit 0

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

Chapter 11
Low-Power Modes

11.1 Introduction

The microcontroller (MCU) may enter two low-power modes: wait mode and stop mode. They are common to all HC08 MCUs and are entered through instruction execution. This section describes how each module acts in the low-power modes.

11.1.1 Wait Mode

The WAIT instruction puts the MCU in a low-power standby mode in which the central processor unit (CPU) clock is disabled but the bus clock continues to run. Power consumption can be further reduced by disabling the low-voltage inhibit (LVI) module through bits in the CONFIG1 register. See Chapter 5 Configuration Register (CONFIG).

11.1.2 Stop Mode

Stop mode is entered when a STOP instruction is executed. The CPU clock is disabled and the bus clock is disabled.

11.2 Analog-to-Digital Converter (ADC)

11.2.1 Wait Mode

The analog-to-digital converter (ADC) continues normal operation during wait mode. Any enabled CPU interrupt request from the ADC can bring the MCU out of wait mode. If the ADC is not required to bring the MCU out of wait mode, power down the ADC by setting ADCH4–ADCH0 bits in the ADC status and control register before executing the WAIT instruction.

11.2.2 Stop Mode

The ADC module is inactive after the execution of a STOP instruction. Any pending conversion is aborted. ADC conversions resume when the MCU exits stop mode after an external interrupt. Allow one conversion cycle to stabilize the analog circuitry.

11.3 Break Module (BRK)

11.3.1 Wait Mode

If enabled, the break (BRK) module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if the SBSW bit in the break status register is set.

Table 14-3. Port C Pin Functions

PTCPUE Bit	DDRC Bit	PTC Bit	I/O Pin Mode	Accesses to DDRC	Accesses to PTC	
				Read/Write	Read	Write(1)
1	0	X(2)	Input, V_{DD}(3)	DDRC1–DDRC0	Pin	PTC1–PTC0(4)
0	0	X	Input, Hi-Z(5)	DDRC1–DDRC0	Pin	PTC1–PTC0(4)
X	1	X	Output	DDRC1–DDRC0	PTC2–PTC0	PTC1–PTC0

NOTES:
1. Output does not apply to PTC2.
2. X = Don't care
3. I/O pin pulled up to V_{DD} by internal pullup device.
4. Writing affects data register, but does not affect input.
5. Hi-Z = High impedance

14.4.3 Port C Input Pullup Enable Register

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

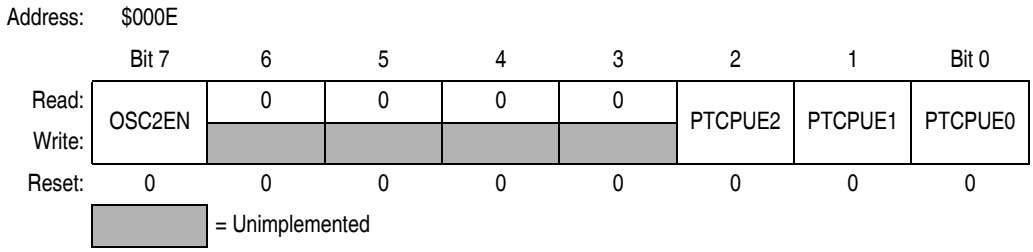


Figure 14-12. Port C Input Pullup Enable Register (PTCPUE)

OSC2EN — Enable PTC1 on OSC2 Pin

This read/write bit configures the OSC2 pin function when internal oscillator or RC oscillator option is selected. this bit has no effect for the XTAL or external oscillator options.

1 = OSC2 pin outputs the internal or RC oscillator clock (BUSCLKX4)

0 = OSC2 pin configured for PTC1 I/O, having all the interrupt and pullup functions

PTCPUE2–PTCPUE0 — Port C Input Pullup Enable Bits

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

1 = Corresponding port C pin configured to have internal pullup

0 = Corresponding port C pin internal pullup disconnected

POL0 — This read/write bit selects the polarity of the PWM waveform of PWM1. Positive polarity means that when the PWM is active the PWM output is high. Conversely, negative polarity means that when the PWM is active the PWM output is low.

1 = PWM0 has positive polarity
0 = PWM0 has negative polarity

PRSC1 and PRSC0 — PWM Prescaler Bits

These buffered read/write bits allow the PWM clock frequency to be modified as shown in Table 15-5.

NOTE

When reading these bits, the value read is the buffer value (not necessarily the value the PWM generator is currently using).

Table 15-5. PWM Prescaler

Prescaler Bits PRSC1 and PRSC0	PWM Clock Frequency
00	BUSCLK
01	BUSCLK/2
10	BUSCLK/4
11	BUSCLK/8

15.8.6 PWM Disable Mapping Write-Once Register

The PWM disable mapping write-once register (DISMAP) contains two bits that control the PWM pins that will be disabled if an external fault occurs. After this register is written for the first time, it cannot be rewritten unless a reset occurs.

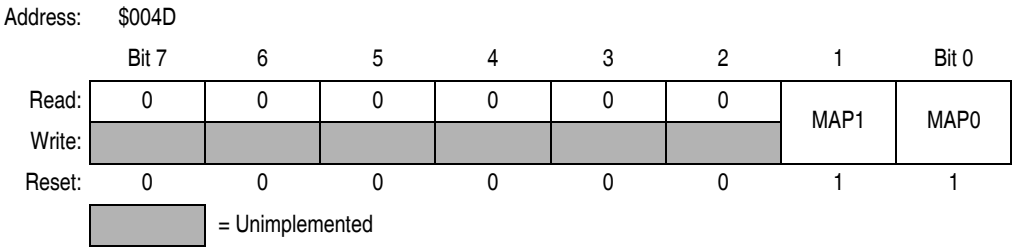


Figure 15-22. PWM Disable Mapping Write-Once Register (DISMAP)

MAP1 — Disable Map for PWM1 Bit

This write-once bit allows the user to select PWM1 to be disabled when a logic 1 is present on the FAULT pin.

1 = Disables PWM1 when an external fault occurs
0 = Prevents PWM1 from being disabled by hardware

MAP0 — Disable Map for PWM0 Bit

This write-once bit allows the user to select PWM0 to be disabled when a logic 1 is present on the FAULT pin.

1 = Disables PWM0 when an external fault occurs
0 = Prevents PWM0 from being disabled by hardware

15.8.7 Fault Control Register

The fault control register (FCR) controls the fault-protection circuitry.

Address: $0042

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

= Unimplemented

Figure 15-23. Fault Control Register (FCR)

FINT — Fault Interrupt Enable Bit

This read/write bit allows the CPU interrupt caused by faults on the fault pin to be enabled. The fault protection circuitry is independent of this bit and will always be active. If a fault is detected, the PWM pins will still be disabled according to the disable mapping register.

1 = Fault pin will cause CPU interrupts
0 = Fault pin will not cause CPU interrupts

FMODE — Fault Mode Selection for Fault Pin Bit (automatic versus manual mode)

This read/write bit allows the user to select between automatic and manual mode faults. For further descriptions of each mode, see 15.5 Fault Protection.

1 = Automatic mode
0 = Manual mode

15.8.8 Fault Status Register

The fault status register (FSR) is a read-only register that indicates the current fault status.

Address: $0043

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	FPIN	FFLAG
Write:								
Reset:	0	0	0	0	0	0	U	0

= Unimplemented U = Unaffected

Figure 15-24. Fault Status Register (FSR)

FPIN — State of Fault Pin Bit

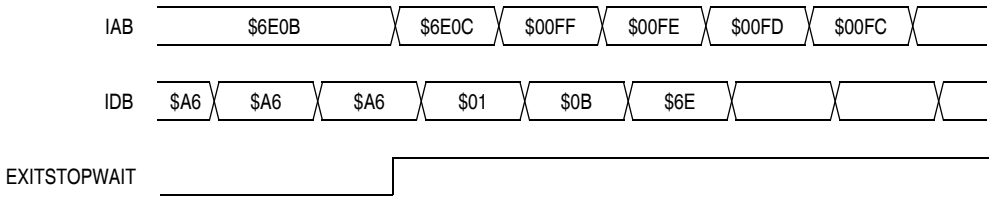
This read-only bit allows the user to read the current state of the fault pin.

1 = Fault pin is at logic 1
0 = Fault pin is at logic 0

FFLAG — Fault Event Flag

The FFLAG event bit is set immediately when a rising edge is seen on the fault pin. To clear the FFLAG bit, the user must write a 1 to the FTACK bit in the fault acknowledge register.

1 = A fault has occurred on the fault pin
0 = No new fault on the fault pin



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

Figure 17-13. Wait Recovery from Interrupt

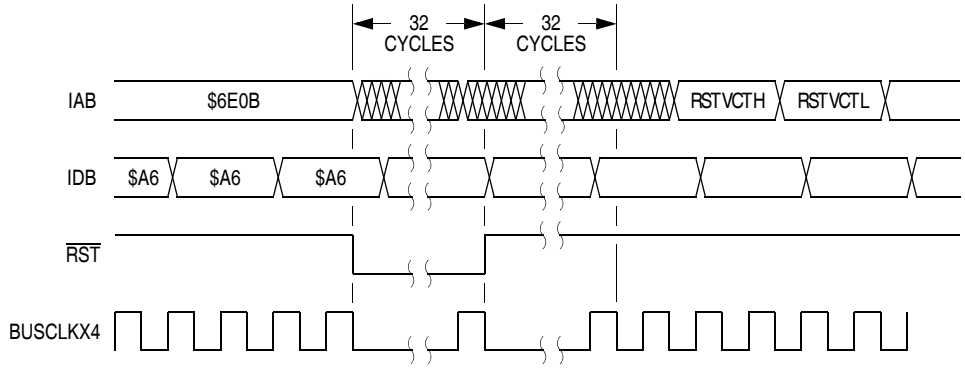


Figure 17-14. Wait Recovery from Internal Reset

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.

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

- Selects output toggling on TIM overflow
- Selects 0% and 100% PWM duty cycle
- Selects buffered or unbuffered output compare/PWM operation

Address: $0025

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Figure 18-10. TIM Channel 0 Status and Control Register (TSC0)

Address: $0028

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Figure 18-11. TIM Channel 1 Status and Control Register (TSC1)

CHxF — Channel x Flag Bit

When channel x is an input capture channel, this read/write bit is set when an active edge occurs on the channel x pin. When channel x is an output compare channel, CHxF is set when the value in the TIM counter registers matches the value in the TIM channel x registers.

When TIM CPU interrupt requests are enabled (CHxIE = 1), clear CHxF by reading TIM channel x status and control register with CHxF set and then writing a 0 to CHxF. If another interrupt request occurs before the clearing sequence is complete, then writing 0 to CHxF has no effect. Therefore, an interrupt request cannot be lost due to inadvertent clearing of CHxF.

Reset clears the CHxF bit. Writing a 1 to CHxF has no effect.

1 = Input capture or output compare on channel x
0 = No input capture or output compare on channel x

19.2.1.2 CPU During Break Interrupts

The CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with $FFFC:$FFFD ($FEFC:$FEFD in monitor mode)

The break interrupt begins after completion of the CPU instruction in progress. If the break address register match occurs on the last cycle of a CPU instruction, the break interrupt begins immediately.

19.2.1.3 TIM During Break Interrupts

A break interrupt stops the timer counter.

19.2.1.4 COP During Break Interrupts

The COP is disabled during a break interrupt with monitor mode when BDCOP bit is set in break auxiliary register (BRKAR).

19.2.2 Break Module Registers

These registers control and monitor operation of the break module:

- Break status and control register (BRKSCR)
- Break address register high (BRKH)
- Break address register low (BRKL)
- Break status register (BSR)
- Break flag control register (BFCR)

19.2.2.1 Break Status and Control Register

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

Address: $FE0B

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

= Unimplemented

Figure 19-3. Break Status and Control Register (BRKSCR)

BRKE — Break Enable Bit

This read/write bit enables breaks on break address register matches. Clear BRKE by writing a 0 to bit 7. Reset clears the BRKE bit.

1 = Breaks enabled on 16-bit address match
0 = Breaks disabled

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a 1 to BRKA generates a break interrupt. Clear BRKA by writing a 0 to it before exiting the break routine. Reset clears the BRKA bit.

1 = Break address match
0 = No break address match

If entering monitor mode without high voltage on $\overline{IRQ}$ (above condition set 2 or 3, where applied voltage is V_{DD} or V_{SS}), then startup port pin requirements and conditions, (PTA1/PTA4) are not in effect. This is to reduce circuit requirements when performing in-circuit programming.

19.3.1.1 Normal Monitor Mode

$\overline{RST}$ and OSC1 functions will be active on the PTA5 and PTC0 pins, respectively, as long as V_{TST} is applied to the $\overline{IRQ}$ pin. If the $\overline{IRQ}$ pin is lowered (no longer V_{TST}) then the chip will still be operating in monitor mode, but the pin functions will be determined by the settings in the configuration register when V_{TST} was lowered. See Chapter 5 Configuration Register (CONFIG).

When monitor mode is entered with V_{TST} on $\overline{IRQ}$, the computer operating properly (COP) is disabled as long as V_{TST} is applied to $\overline{IRQ}$. This condition states that as long as V_{TST} is maintained on the $\overline{IRQ}$ pin after entering monitor mode, then the COP will be disabled.

19.3.1.2 Forced Monitor Mode

If the voltage applied to the $\overline{IRQ1}$ is less than V_{TST}, the MCU will come out of reset in user mode. However, when the reset vector is erased ($FFFF), the MCU is forced into monitor mode without requiring high voltage on the $\overline{IRQ1}$ pin. Once out of reset, the monitor code is initially executing off the internal clock at its default frequency.

If $\overline{IRQ}$ is tied high (V_{DD}), all pins will default to regular input port functions except for PTA0 and PTC0 which will operate as a serial communication port and OSC1 input respectively (refer to Figure 19-11). That will allow the clock to be driven from an external source through OSC1 pin.

If $\overline{IRQ}$ is tied low, all pins will default to regular input port function except for PTA0 which will operate as serial communication port. Refer to Figure 19-12. Regardless of the state of the $\overline{IRQ}$ pin, it will not function as a port input pin in monitor mode.

The COP module is disabled in forced monitor mode.

NOTE

If the reset vector is blank and monitor mode is entered, the chip will see an additional reset cycle after the initial power-on reset (POR). Once the part has been programmed, the traditional method of applying a voltage, V_{TST}, to $\overline{IRQ}$ must be used to enter monitor mode.

19.3.1.3 Monitor Vectors

In monitor mode, the MCU uses different vectors for reset, SWI (software interrupt), and break interrupt than those for user mode. The alternate vectors are in the $FE page instead of the $FF page and allow code execution from the internal monitor firmware instead of user code.

Table 19-2 summarizes the differences between user mode and monitor mode regarding vectors.

Table 19-2. Mode Difference

Modes	Functions					
	Reset Vector High	Reset Vector Low	Break Vector High	Break Vector Low	SWI Vector High	SWI Vector Low
User	$FFFE	$FFFF	$FFFC	$FFFD	$FFFC	$FFFD
Monitor	$FEFE	$FEFF	$FEFC	$FEFD	$FEFC	$FEFD