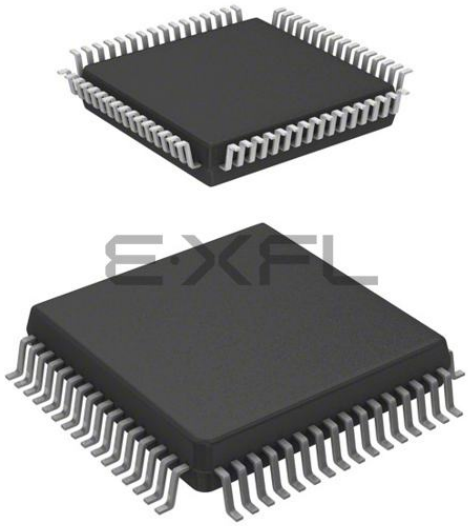


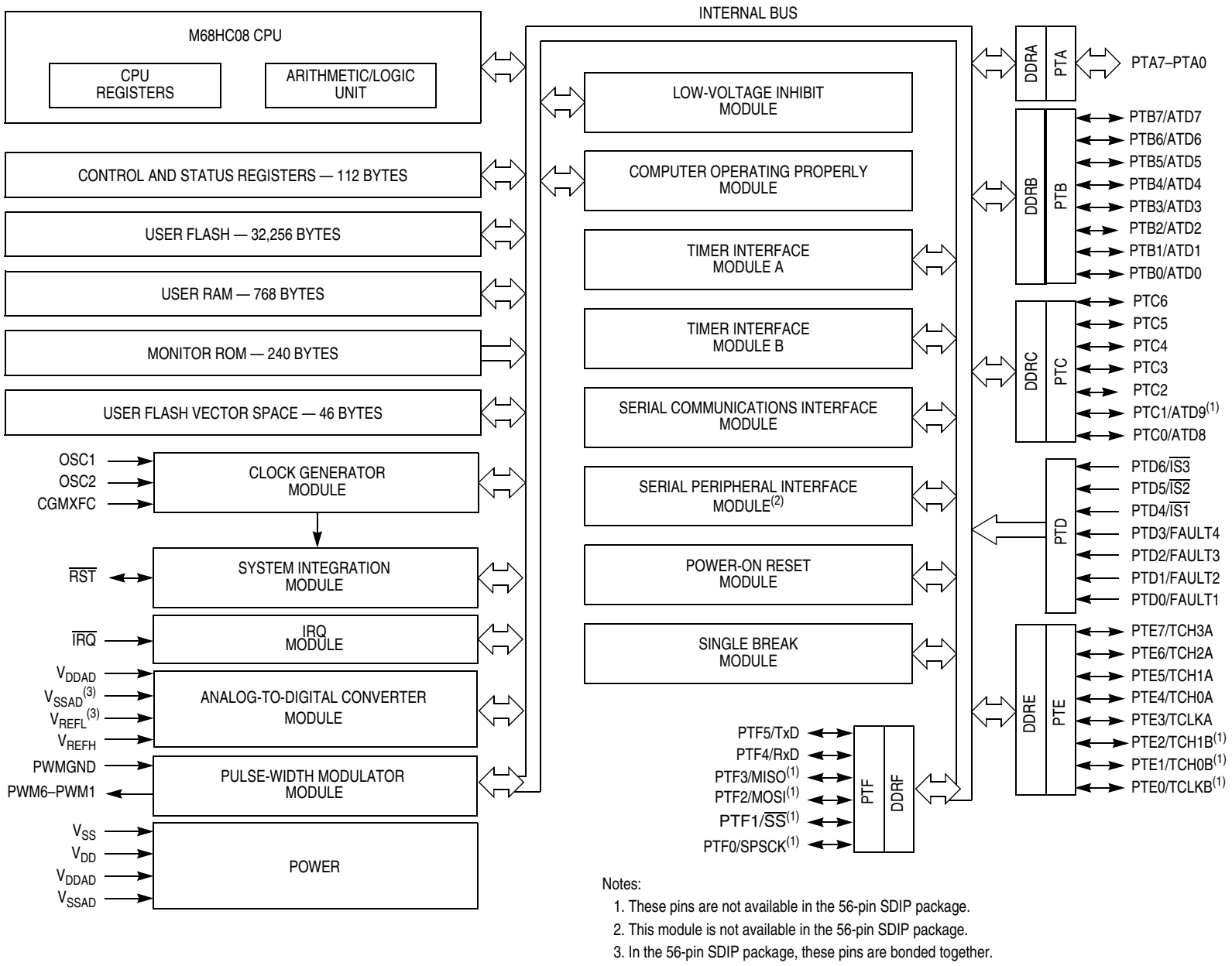
Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	44
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908mr16cfue



Notes:
1. These pins are not available in the 56-pin SDIP package.
2. This module is not available in the 56-pin SDIP package.
3. In the 56-pin SDIP package, these pins are bonded together.

Figure 1-1. MCU Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0026	PWM Counter Register High (PCNTH) See page 143.	Read:	0	0	0	0	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0027	PWM Counter Register Low (PCNTL) See page 143.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0028	PWM Counter Modulo Register High (PMODH) See page 144.	Read:	0	0	0	0	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	0	0	0	0	X	X	X	X
$0029	PWM Counter Modulo Register Low (PMODL) See page 144.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	X	X	X	X	X	X	X	X
$002A	PWM 1 Value Register High (PVAL1H) See page 145.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$002B	PWM 1 Value Register Low (PVAL1L) See page 145.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$002C	PWM 2 Value Register High (PVAL2H) See page 145.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$002D	PWM 2 Value Register Low (PVAL2L) See page 145.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$002E	PWM 3 Value Register High (PVAL3H) See page 145.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$002F	PWM 3 Value Register Low (PVAL3L) See page 145.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0030	PWM 4 Value Register High (PVAL4H) See page 145.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0031	PWM 4 Value Register Low (PVAL4L) See page 145.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

U = Unaffected X = Indeterminate R = Reserved **Bold** = Buffered = Unimplemented

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

Table 2-1 is a list of vector locations.

Table 2-1. Vector Addresses

Low

Priority

Address	Vector
$FFD2	SCI transmit vector (high)
$FFD3	SCI transmit vector (low)
$FFD4	SCI receive vector (high)
$FFD5	SCI receive vector (low)
$FFD6	SCI error vector (high)
$FFD7	SCI error vector (low)
$FFD8	SPI transmit vector (high)(1)
$FFD9	SPI transmit vector (low)(1)
$FFDA	SPI receive vector (high)(1)
$FFDB	SPI receive vector (low)(1)
$FFDC	A/D vector (high)
$FFDD	A/D vector (low)
$FFDE	TIMB overflow vector (high)
$FFDF	TIMB overflow vector (low)
$FFE0	TIMB channel 1 vector (high)
$FFE1	TIMB channel 1 vector (low)
$FFE2	TIMB channel 0 vector (high)
$FFE3	TIMB channel 0 vector (low)
$FFE4	TIMA overflow vector (high)
$FFE5	TIMA overflow vector (low)
$FFE6	TIMA channel 3 vector (high)
$FFE7	TIMA channel 3 vector (low)
$FFE8	TIMA channel 2 vector (high)
$FFE9	TIMA channel 2 vector (low)
$FFEA	TIMA channel 1 vector (high)
$FFEB	TIMA channel 1 vector (low)
$FFEC	TIMA channel 0 vector (high)
$FFED	TIMA channel 0 vector (low)

1. The SPI module is not available in the 56-pin SDIP package.

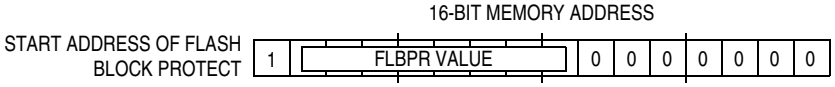


Figure 2-6. FLASH Block Protect Start Address

Refer to Table 2-2 for examples of the protect start address.

Table 2-2. Examples of Protect Start Address

BPR[7:0]	Start of Address of Protect Range
$00	The entire FLASH memory is protected.
$01 (**0000 0001**)	$8080 (1**000 0000 1**000 0000)
$02 (**0000 0010**)	$8100 (1**000 0001 0**000 0000)
and so on...	
$FE (**1111 1110**)	$FF00 (1**111 1111 0**000 0000)
$FF	The entire FLASH memory is not protected.

Note: The end address of the protected range is always $FFFF.

2.8.7 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.

2.8.8 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.

The PLL also may operate in manual mode (AUTO = 0). Manual mode is used by systems that do not require an indicator of the lock condition for proper operation. Such systems typically operate well below f_{BUSMAX} and require fast startup. These conditions apply when in manual mode:

- $\overline{ACQ}$ is a writable control bit that controls the mode of the filter. Before turning on the PLL in manual mode, the $\overline{ACQ}$ bit must be clear.
- Before entering tracking mode ($\overline{ACQ}$ = 1), software must wait a given time, t_{ACQ} (see 4.8 Acquisition/Lock Time Specifications), after turning on the PLL by setting PLLON in the PLL control register (PCTL).
- Software must wait a given time, t_{AL}, after entering tracking mode before selecting the PLL as the clock source to CGMOUT (BCS = 1).
- The LOCK bit is disabled.
- CPU interrupts from the CGM are disabled.

4.3.2.4 Programming the PLL

Use this 9-step procedure to program the PLL. Table 4-1 lists the variables used and their meaning.

Table 4-1. Variable Definitions

Variable	Definition
f_{BUSDES}	Desired bus clock frequency
$f_{VCLKDES}$	Desired VCO clock frequency
f_{RCLK}	Chosen reference crystal frequency
f_{VCLK}	Calculated VCO clock frequency
f_{BUS}	Calculated bus clock frequency
f_{NOM}	Nominal VCO center frequency
f_{VRS}	Shifted FCO center frequency

1. Choose the desired bus frequency, f_{BUSDES}.

   Example: f_{BUSDES} = 8 MHz
2. Calculate the desired VCO frequency, $f_{VCLKDES}$.

   $$f_{VCLKDES} = 4 \times f_{BUSDES}$$

   Example: $f_{VCLKDES} = 4 \times 8\ MHz = 32\ MHz$
3. Using a reference frequency, f_{RCLK}, equal to the crystal frequency, calculate the VCO frequency multiplier, N. Round the result to the nearest integer.

   $$N = \frac{f_{VCLKDES}}{f_{RCLK}}$$

   $$\text{Example: } N = \frac{32\ MHz}{4\ MHz} = 8\ MHz$$
4. Calculate the VCO frequency, f_{VCLK}.

   $$f_{VCLK} = N \times f_{RCLK}$$

   Example: $f_{VCLK} = 8 \times 4\ MHz = 32\ MHz$

4.8 Acquisition/Lock Time Specifications

The acquisition and lock times of the PLL are, in many applications, the most critical PLL design parameters. Proper design and use of the PLL ensures the highest stability and lowest acquisition/lock times.

4.8.1 Acquisition/Lock Time Definitions

Typical control systems refer to the acquisition time or lock time as the reaction time, within specified tolerances, of the system to a step input. In a PLL, the step input occurs when the PLL is turned on or when it suffers a noise hit. The tolerance is usually specified as a percent of the step input or when the output settles to the desired value plus or minus a percent of the frequency change. Therefore, the reaction time is constant in this definition, regardless of the size of the step input. For example, consider a system with a 5 percent acquisition time tolerance. If a command instructs the system to change from 0 Hz to 1 MHz, the acquisition time is the time taken for the frequency to reach 1 MHz ± 50 kHz. Fifty kHz = 5% of the 1-MHz step input. If the system is operating at 1 MHz and suffers a –100-kHz noise hit, the acquisition time is the time taken to return from 900 kHz to 1 MHz ±5 kHz. Five kHz = 5% of the 100-kHz step input.

Other systems refer to acquisition and lock times as the time the system takes to reduce the error between the actual output and the desired output to within specified tolerances. Therefore, the acquisition or lock time varies according to the original error in the output. Minor errors may not even be registered. Typical PLL applications prefer to use this definition because the system requires the output frequency to be within a certain tolerance of the desired frequency regardless of the size of the initial error.

The discrepancy in these definitions makes it difficult to specify an acquisition or lock time for a typical PLL. Therefore, the definitions for acquisition and lock times for this module are:

- Acquisition time, t_{ACQ}, is the time the PLL takes to reduce the error between the actual output frequency and the desired output frequency to less than the tracking mode entry tolerance, Δ_{TRK}. Acquisition time is based on an initial frequency error, $(f_{DES} - f_{ORIG})/f_{DES}$, of not more than ±100 percent. In automatic bandwidth control mode (see 4.3.2.3 Manual and Automatic PLL Bandwidth Modes), acquisition time expires when the $\overline{ACQ}$ bit becomes set in the PLL bandwidth control register (PBWC).
- Lock time, t_{Lock}, is the time the PLL takes to reduce the error between the actual output frequency and the desired output frequency to less than the lock mode entry tolerance, Δ_{Lock}. Lock time is based on an initial frequency error, $(f_{DES} - f_{ORIG})/f_{DES}$, of not more than ±100 percent. In automatic bandwidth control mode, lock time expires when the LOCK bit becomes set in the PLL bandwidth control register (PBWC). See 4.3.2.3 Manual and Automatic PLL Bandwidth Modes.

Obviously, the acquisition and lock times can vary according to how large the frequency error is and may be shorter or longer in many cases.

4.8.2 Parametric Influences on Reaction Time

Acquisition and lock times are designed to be as short as possible while still providing the highest possible stability. These reaction times are not constant, however. Many factors directly and indirectly affect the acquisition time.

The most critical parameter which affects the reaction times of the PLL is the reference frequency, f_{RDV}. This frequency is the input to the phase detector and controls how often the PLL makes corrections. For stability, the corrections must be small compared to the desired frequency, so several corrections are

6.3.6 COPD (COP Disable)

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

6.4 COP Control Register

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

Address:	$FFFF							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Low byte of reset vector							
Write:	Clear COP counter							
Reset:	Unaffected by reset							

Figure 6-3. 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_{HI} is present on the $\overline{IRQ}$ pin or on the $\overline{RST}$ pin.

6.7 Wait Mode

The WAIT instruction puts the MCU in low power-consumption standby mode.

The COP continues to operate during wait mode.

6.8 Stop Mode

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

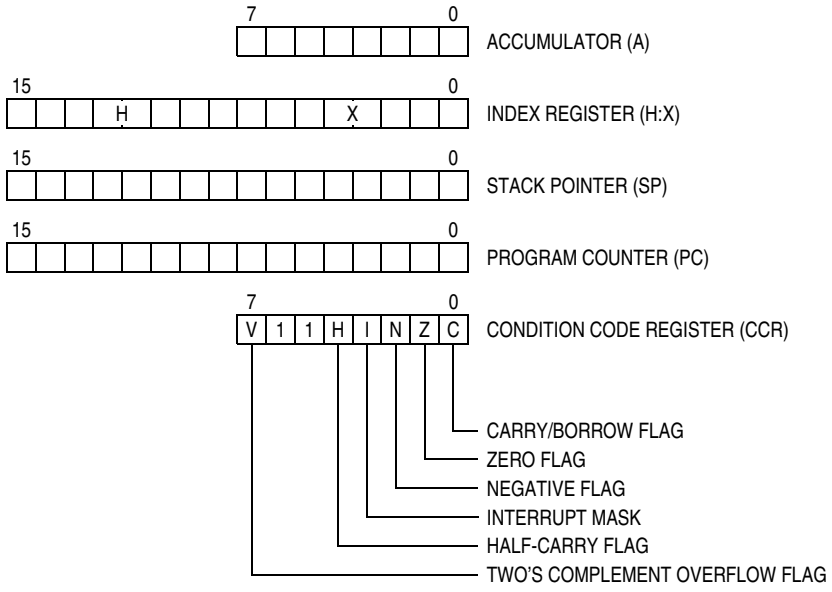


Figure 7-1. CPU Registers

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.

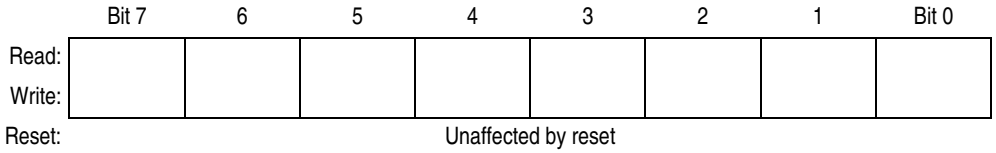


Figure 7-2. Accumulator (A)

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.

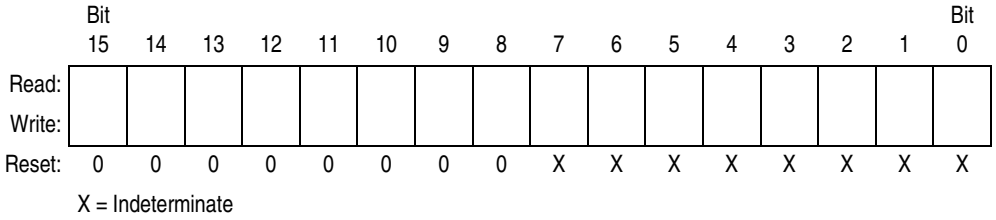


Figure 7-3. Index Register (H:X)

Chapter 10
Input/Output (I/O) Ports (PORTS)

10.1 Introduction

Thirty-seven bidirectional input-output (I/O) pins and seven input pins form six parallel ports. All I/O pins are programmable as inputs or outputs.

When using the 56-pin package version:

- Set the data direction register bits in DDRC such that bit 1 is written to a logic 1 (along with any other output bits on port C).
- Set the data direction register bits in DDRE such that bits 0, 1, and 2 are written to a logic 1 (along with any other output bits on port E).
- Set the data direction register bits in DDRF such that bits 0, 1, 2, and 3 are written to a logic 1 (along with any other output bits on port F).

NOTE

Connect any unused I/O pins to an appropriate logic level, either V_{DD} or V_{SS}. Although PWM6–PWM1 do not require termination for proper operation, termination reduces excess current consumption and the possibility of electrostatic damage.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA) See page 103.	Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PTB) See page 104.	Read: Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PTC) See page 106.	Read:	0	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
		Write:	R							
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PTD) See page 107.	Read:	0	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Write:	R	R	R	R	R	R	R	R
		Reset:	Unaffected by reset							
$0004	Data Direction Register A (DDRA) See page 103.	Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Reset:	0	0	0	0	0	0	0	0
			R = Reserved				= Unimplemented			

Figure 10-1. I/O Port Register Summary

Figure 10-18 shows the port F I/O logic.

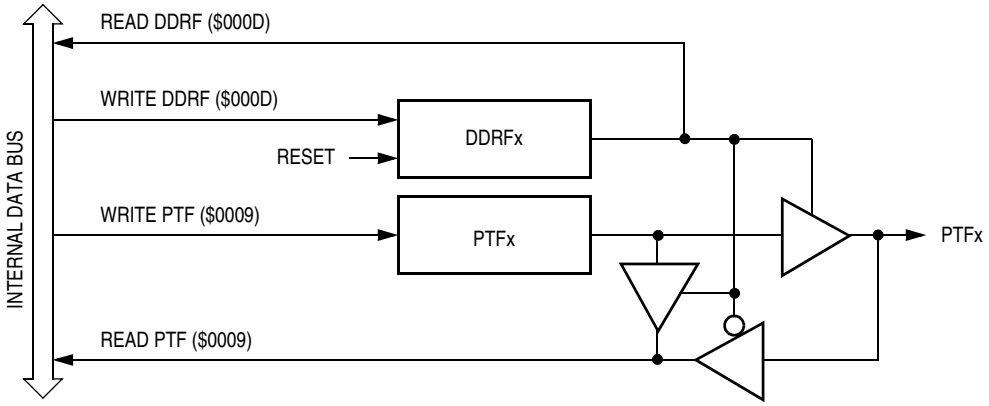


Figure 10-18. Port F I/O Circuit

When bit DDRFx is a logic 1, reading address $0009 reads the PTFx data latch. When bit DDRFx is a logic 0, reading address $0009 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 10-6 summarizes the operation of the port F pins.

Table 10-6. Port F Pin Functions

DDRF Bit	PTF Bit	I/O Pin Mode	Accesses to DDRF	Accesses to PTF	
			Read/Write	Read	Write
0	X[1]	Input, Hi-Z[2]	DDRF[6:0]	Pin	PTF[6:0][3]
1	X	Output	DDRF[6:0]	PTF[6:0]	PTF[6:0]

1. X = don't care
2. Hi-Z = high impedance
3. Writing affects data register, but does not affect input.

Chapter 12
Pulse-Width Modulator for Motor Control (PWMMC)

12.1 Introduction

This section describes the pulse-width modulator for motor control (PWMMC, version A). The PWM module can generate three complementary PWM pairs or six independent PWM signals. These PWM signals can be center-aligned or edge-aligned. A block diagram of the PWM module is shown in Figure 12-2.

A12-bit timer PWM counter is common to all six channels. PWM resolution is one clock period for edge-aligned operation and two clock periods for center-aligned operation. The clock period is dependent on the internal operating frequency (f_{OP}) and a programmable prescaler. The highest resolution for edge-aligned operation is 125 ns (f_{OP} = 8 MHz). The highest resolution for center-aligned operation is 250 ns (f_{OP} = 8 MHz).

When generating complementary PWM signals, the module features automatic dead-time insertion to the PWM output pairs and transparent toggling of PWM data based upon sensed motor phase current polarity.

A summary of the PWM registers is shown in Figure 12-3.

12.2 Features

Features of the PWMMC include:

- Three complementary PWM pairs or six independent PWM signals
- Edge-aligned PWM signals or center-aligned PWM signals
- PWM signal polarity control
- 20-mA current sink capability on PWM pins
- Manual PWM output control through software
- Programmable fault protection
- Complementary mode featuring:
  - Dead-time insertion
  - Separate top/bottom pulse width correction via current sensing or programmable software bits

12.9.4 PWM Control Register 1

PWM control register 1 (PCTL1) controls PWM enabling/disabling, the loading of new modulus, prescaler, PWM values, and the PWM correction method. In addition, this register contains the software disable bits to force the PWM outputs to their inactive states (according to the disable mapping register).

Address: $0020

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DISX	DISY	PWMINT	PWMF	ISENS1	ISENS0	LDOK	PWMEN
Reset:	0	0	0	0	0	0	0	0

Figure 12-39. PWM Control Register 1 (PCTL1)

DISX — Software Disable Bit for Bank X Bit

This read/write bit allows the user to disable one or more PWM pins in bank X. The pins that are disabled are determined by the disable mapping write-once register.

1 = Disable PWM pins in bank X.
0 = Re-enable PWM pins at beginning of next PWM cycle.

DISY — Software Disable Bit for Bank Y Bit

This read/write bit allows the user to disable one or more PWM pins in bank Y. The pins that are disabled are determined by the disable mapping write-once register.

1 = Disable PWM pins in bank Y.
0 = Re-enable PWM pins at beginning of next PWM cycle.

PWMINT — PWM Interrupt Enable Bit

This read/write bit allows the user to enable and disable PWM CPU interrupts. If set, a CPU interrupt will be pending when the PWMF flag is set.

1 = Enable PWM CPU interrupts.
0 = Disable PWM CPU interrupts.

NOTE

When PWMINT is cleared, pending CPU interrupts are inhibited.

PWMF — PWM Reload Flag

This read/write bit is set at the beginning of every reload cycle regardless of the state of the LDOK bit. This bit is cleared by reading PWM control register 1 with the PWMF flag set, then writing a logic 0 to PWMF. If another reload occurs before the clearing sequence is complete, then writing logic 0 to PWMF has no effect.

1 = New reload cycle began.
0 = New reload cycle has not begun.

NOTE

When PWMF is cleared, pending PWM CPU interrupts are cleared (not including fault interrupts).

ISENS1 and ISENS0 — Current Sense Correction Bits

These read/write bits select the top/bottom correction scheme as shown in Table 12-7.

IDLE — Receiver Idle Bit

This clearable, read-only bit is set when 10 or 11 consecutive 1s appear on the receiver input. IDLE generates an SCI error CPU interrupt request if the ILIE bit in SCC2 is also set. Clear the IDLE bit by reading SCS1 with IDLE set and then reading the SCDR. After the receiver is enabled, it must receive a valid character that sets the SCRF bit before an idle condition can set the IDLE bit. Also, after the IDLE bit has been cleared, a valid character must again set the SCRF bit before an idle condition can set the IDLE bit. Reset clears the
IDLE bit.

1 = Receiver input idle
0 = Receiver input active or idle since the IDLE bit was cleared

OR — Receiver Overrun Bit

This clearable, read-only bit is set when software fails to read the SCDR before the receive shift register receives the next character. The OR bit generates an SCI error CPU interrupt request if the ORIE bit in SCC3 is also set. The data in the shift register is lost, but the data already in the SCDR is not affected. Clear the OR bit by reading SCS1 with OR set and then reading the SCDR. Reset clears the OR bit.

1 = Receive shift register full and SCRF = 1
0 = No receiver overrun

Software latency may allow an overrun to occur between reads of SCS1 and SCDR in the flag-clearing sequence. Figure 13-12 shows the normal flag-clearing sequence and an example of an overrun caused by a delayed flag-clearing sequence. The delayed read of SCDR does not clear the OR bit because OR was not set when SCS1 was read. Byte 2 caused the overrun and is lost. The next flag-clearing sequence reads byte 3 in the SCDR instead of byte 2.

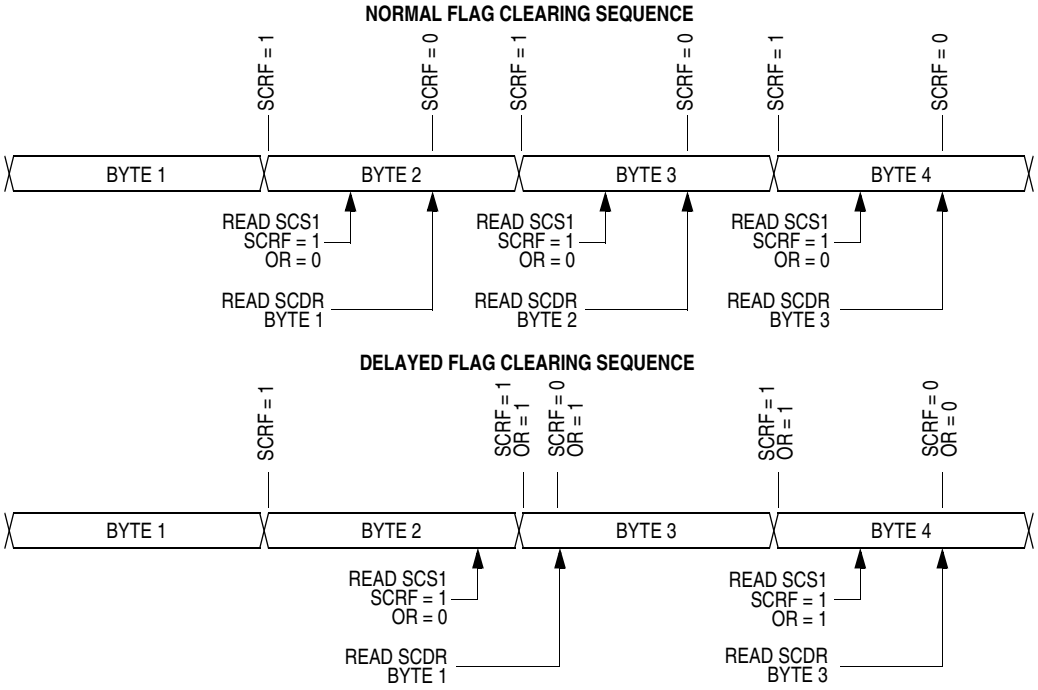


Figure 13-12. Flag Clearing Sequence

14.5.1.2 Software Interrupt (SWI) Instruction

The software interrupt (SWI) instruction is a non-maskable instruction that causes an interrupt regardless of the state of the interrupt mask
(I bit) in the condition code register.

14.5.2 Reset

All reset sources always have equal and highest priority and cannot be arbitrated.

14.6 Low-Power Mode

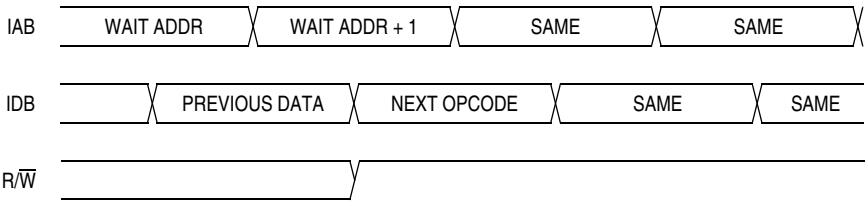
Executing the WAIT instruction puts the MCU in a low power-consumption mode for standby situations. The SIM holds the CPU in a non-clocked state. WAIT clears the interrupt mask (I) in the condition code register, allowing interrupts to occur.

14.6.1 Wait Mode

In wait mode, the CPU clocks are inactive while the peripheral clocks continue to run. Figure 14-11 shows the timing for wait mode entry.

A module that is active during wait mode can wake up the CPU with an interrupt if the interrupt is enabled. Stacking for the interrupt begins one cycle after the WAIT instruction during which the interrupt occurred. 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.

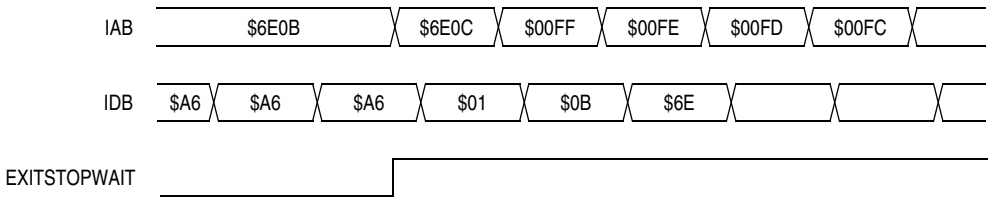
Wait mode can also be exited by a reset. If the COP disable bit, COPD, in the configuration register is logic 0, then the computer operating properly module (COP) is enabled and remains active in wait mode.



Note: Previous data can be operand data or the WAIT opcode, depending on the last instruction.

Figure 14-11. Wait Mode Entry Timing

Figure 14-12 and Figure 14-13 show the timing for wait recovery.



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

Figure 14-12. Wait Recovery from Interrupt

x referring to the active channel number). When an active edge occurs on the pin of an input capture channel, the TIMA latches the contents of the TIMA counter into the TIMA channel registers, TACHxH–TACHxL. Input captures can generate TIMA CPU interrupt requests. Software can determine that an input capture event has occurred by enabling input capture interrupts or by polling the status flag bit.

The free-running counter contents are transferred to the TIMA channel status and control register (TACHxH–TACHxL, see 16.7.5 TIMA Channel Registers) on each proper signal transition regardless of whether the TIMA channel flag (CH0F–CH3F in TASC0–TASC3 registers) is set or clear. When the status flag is set, a CPU interrupt is generated if enabled. The value of the count latched or “captured” is the time of the event. Because this value is stored in the input capture register two bus cycles after the actual event occurs, user software can respond to this event at a later time and determine the actual time of the event. However, this must be done prior to another input capture on the same pin; otherwise, the previous time value will be lost.

By recording the times for successive edges on an incoming signal, software can determine the period and/or pulse width of the signal. To measure a period, two successive edges of the same polarity are captured. To measure a pulse width, two alternate polarity edges are captured. Software should track the overflows at the 16-bit module counter to extend its range.

Another use for the input capture function is to establish a time reference. In this case, an input capture function is used in conjunction with an output compare function. For example, to activate an output signal a specified number of clock cycles after detecting an input event (edge), use the input capture function to record the time at which the edge occurred. A number corresponding to the desired delay is added to this captured value and stored to an output compare register (see 16.7.5 TIMA Channel Registers). Because both input captures and output compares are referenced to the same 16-bit modulo counter, the delay can be controlled to the resolution of the counter independent of software latencies.

Reset does not affect the contents of the input capture channel registers.

16.3.3 Output Compare

With the output compare function, the TIMA 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 TIMA can set, clear, or toggle the channel pin. Output compares can generate TIMA CPU interrupt requests.

16.3.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in 16.3.3 Output Compare. The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIMA channel registers.

An unsynchronized write to the TIMA 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 TIMA overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIMA may pass the new value before it is written.

16.7.5 TIMA Channel Registers

These read/write registers contain the captured TIMA counter value of the input capture function or the output compare value of the output compare function. The state of the TIMA channel registers after reset is unknown.

In input capture mode (MSxB:MSxA = 0:0), reading the high byte of the TIMA channel x registers (TACHxH) inhibits input captures until the low byte (TACHxL) is read.

In output compare mode (MSxB:MSxA ≠ 0:0), writing to the high byte of the TIMA channel x registers (TACHxH) inhibits output compares until the low byte (TACHxL) is written.

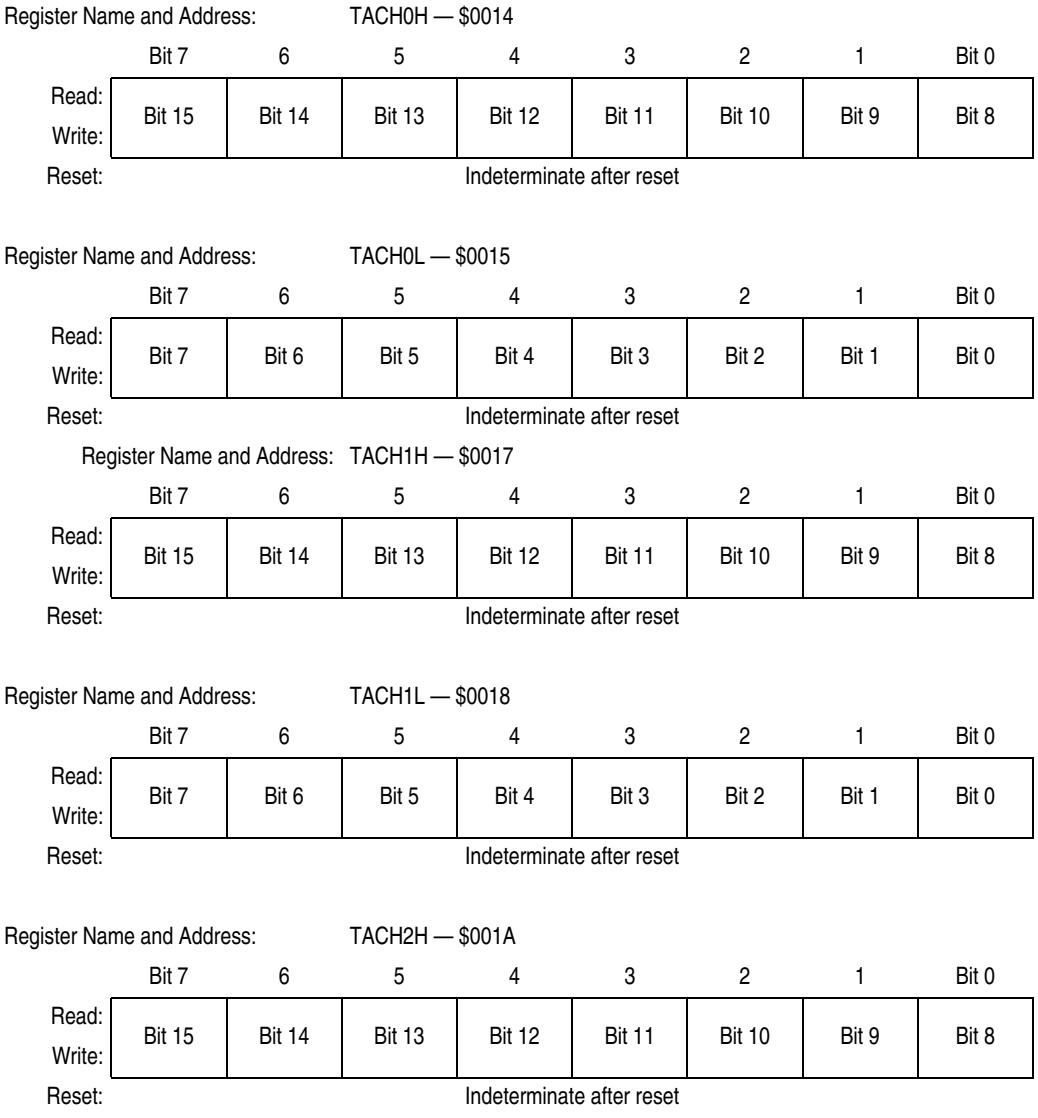


Figure 16-10. TIMA Channel Registers (TACH0H/L–TACH3H/L)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0056	TIMB Channel 0 Status/Control Register (TBSC0) See page 247.	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0057	TIMB Channel 0 Register High (TBCH0H) See page 250.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0058	TIMB Channel 0 Register Low (TBCH0L) See page 250.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0059	TIMB Channel 1 Status/Control Register (TBSC1) See page 247.	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0		R					
		Reset:	0	0	0	0	0	0	0	0
$005A	TIMB Channel 1 Register High (TBCH1H) See page 250.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$005B	TIMB Channel 1 Register Low (TBCH1L) See page 250.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
			R = Reserved							

Figure 17-3. TIMB I/O Register Summary (Continued)

17.3.1 TIMB Counter Prescaler

The TIMB clock source can be one of the seven prescaler outputs or the TIMB clock pin, PTE0/TCLKB. The prescaler generates seven clock rates from the internal bus clock. The prescaler select bits, PS[2:0], in the TIMB status and control register select the TIMB clock source.

17.3.2 Input Capture

An input capture function has three basic parts:

1. Edge select logic
2. Input capture latch
3. 16-bit counter

Two 8-bit registers, which make up the 16-bit input capture register, are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The polarity of the active edge is programmable. The level transition which triggers the counter transfer is defined by the corresponding input edge bits (ELSxB and ELSxA in TBSC0–TBSC1 control registers with x referring to the active channel number). When an active edge occurs on the pin of an input capture channel, the TIMB latches the contents of the TIMB counter into the TIMB channel registers, TCHxH–TCHxL. Input captures can generate TIMB CPU interrupt requests. Software can determine that an input capture event has occurred by enabling input capture interrupts or by polling the status flag bit.

The free-running counter contents are transferred to the TIMB channel status and control register (TBCHxH–TBCHxL, see 17.7.5 TIMB Channel Registers) on each proper signal transition regardless of

18.2.1.2 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 and $FFFD ($FEFC and $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.

18.2.1.3 TIM1 and TIM2 During Break Interrupts

A break interrupt stops the timer counters.

18.2.1.4 COP During Break Interrupts

The COP is disabled during a break interrupt when V_{TST} is present on the $\overline{RST}$ pin.

18.2.2 Low-Power Modes

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

18.2.2.1 Wait Mode

If enabled, the break module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if SBSW is set. Clear the BW bit by writing logic 0 to it.

18.2.2.2 Stop Mode

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

18.2.3 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)
- SIM break status register (SBSR)
- SIM break flag control register (SBFCR)

20.4 56-Pin Shrink Dual In-Line Package (SDIP)