



Welcome to [E-XFL.COM](https://www.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	LVD, POR, PWM
Number of I/O	13
Program Memory Size	1.5KB (1.5K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-SOIC (0.295", 7.50mm Width)
Supplier Device Package	16-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908qy1cdw

Chapter 12

Input/Output Ports (PORTS)

12.1	Introduction	97
12.2	Port A	97
12.2.1	Port A Data Register	98
12.2.2	Data Direction Register A	98
12.2.3	Port A Input Pullup Enable Register	99
12.3	Port B	100
12.3.1	Port B Data Register	100
12.3.2	Data Direction Register B	101
12.3.3	Port B Input Pullup Enable Register	102

Chapter 13

System Integration Module (SIM)

13.1	Introduction	103
13.2	RST and IRQ Pins Initialization	104
13.3	SIM Bus Clock Control and Generation	104
13.3.1	Bus Timing	105
13.3.2	Clock Start-Up from POR	105
13.3.3	Clocks in Stop Mode and Wait Mode	105
13.4	Reset and System Initialization	105
13.4.1	External Pin Reset	106
13.4.2	Active Resets from Internal Sources	106
13.4.2.1	Power-On Reset	107
13.4.2.2	Computer Operating Properly (COP) Reset	108
13.4.2.3	Illegal Opcode Reset	108
13.4.2.4	Illegal Address Reset	108
13.4.2.5	Low-Voltage Inhibit (LVI) Reset	108
13.5	SIM Counter	108
13.5.1	SIM Counter During Power-On Reset	108
13.5.2	SIM Counter During Stop Mode Recovery	109
13.5.3	SIM Counter and Reset States	109
13.6	Exception Control	109
13.6.1	Interrupts	109
13.6.1.1	Hardware Interrupts	111
13.6.1.2	SWI Instruction	112
13.6.2	Interrupt Status Registers	112
13.6.2.1	Interrupt Status Register 1	113
13.6.2.2	Interrupt Status Register 2	113
13.6.2.3	Interrupt Status Register 3	113
13.6.3	Reset	114
13.6.4	Break Interrupts	114
13.6.5	Status Flag Protection in Break Mode	114
13.7	Low-Power Modes	114
13.7.1	Wait Mode	114
13.7.2	Stop Mode	115

Table of Contents

13.8	SIM Registers	116
13.8.1	SIM Reset Status Register	117
13.8.2	Break Flag Control Register	118

Chapter 14 Timer Interface Module (TIM)

14.1	Introduction	119
14.2	Features	119
14.3	Pin Name Conventions	119
14.4	Functional Description	121
14.4.1	TIM Counter Prescaler	122
14.4.2	Input Capture	122
14.4.3	Output Compare	122
14.4.3.1	Unbuffered Output Compare	122
14.4.3.2	Buffered Output Compare	122
14.4.4	Pulse Width Modulation (PWM)	123
14.4.4.1	Unbuffered PWM Signal Generation	124
14.4.4.2	Buffered PWM Signal Generation	124
14.4.4.3	PWM Initialization	125
14.5	Interrupts	125
14.6	Wait Mode	126
14.7	TIM During Break Interrupts	126
14.8	Input/Output Signals	126
14.8.1	TIM Clock Pin (PTA2/TCLK)	126
14.8.2	TIM Channel I/O Pins (PTA0/TCH0 and PTA1/TCH1)	126
14.9	Input/Output Registers	126
14.9.1	TIM Status and Control Register	127
14.9.2	TIM Counter Registers	128
14.9.3	TIM Counter Modulo Registers	129
14.9.4	TIM Channel Status and Control Registers	129
14.9.5	TIM Channel Registers	132

Chapter 15 Development Support

15.1	Introduction	133
15.2	Break Module (BRK)	133
15.2.1	Functional Description	133
15.2.1.1	Flag Protection During Break Interrupts	135
15.2.1.2	TIM During Break Interrupts	135
15.2.1.3	COP During Break Interrupts	135
15.2.2	Break Module Registers	135
15.2.2.1	Break Status and Control Register	136
15.2.2.2	Break Address Registers	136
15.2.2.3	Break Auxiliary Register	137
15.2.2.4	Break Status Register	137
15.2.2.5	Break Flag Control Register	138
15.2.3	Low-Power Modes	138

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 5)

ERASE — Erase Control Bit

This read/write bit configures the memory for erase operation. ERASE is interlocked with the PGM bit such that both bits cannot be equal to 1 or set to 1 at the same time.

- 1 = Erase operation selected
- 0 = Erase operation unselected

PGM — Program Control Bit

This read/write bit configures the memory for program operation. PGM is interlocked with the ERASE bit such that both bits cannot be equal to 1 or set to 1 at the same time.

- 1 = Program operation selected
- 0 = Program operation unselected

2.6.2 FLASH Page Erase Operation

Use the following procedure to erase a page of FLASH memory. A page consists of 64 consecutive bytes starting from addresses \$XX00, \$XX40, \$XX80, or \$XXC0. The 48-byte user interrupt vectors area also forms a page. Any FLASH memory page can be erased alone.

1. Set the ERASE bit and clear the MASS bit in the FLASH control register.
2. Read the FLASH block protect register.
3. Write any data to any FLASH location within the address range of the block to be erased.
4. Wait for a time, t_{NVS} (minimum 10 μ s).
5. Set the HVEN bit.
6. Wait for a time, t_{Erase} (minimum 1 ms or 4 ms).
7. Clear the ERASE bit.
8. Wait for a time, t_{NVH} (minimum 5 μ s).
9. Clear the HVEN bit.
10. After time, t_{RCV} (typical 1 μ s), the memory can be accessed in read mode again.

NOTE

Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps.

CAUTION

A page erase of the vector page will erase the internal oscillator trim values at \$FFC0 and \$FFC1.

In applications that require more than 1000 program/erase cycles, use the 4 ms page erase specification to get improved long-term reliability. Any application can use this 4 ms page erase specification. However, in applications where a FLASH location will be erased and reprogrammed less than 1000 times, and speed is important, use the 1 ms page erase specification to get a shorter cycle time.



Auto Wakeup Module (AWU)

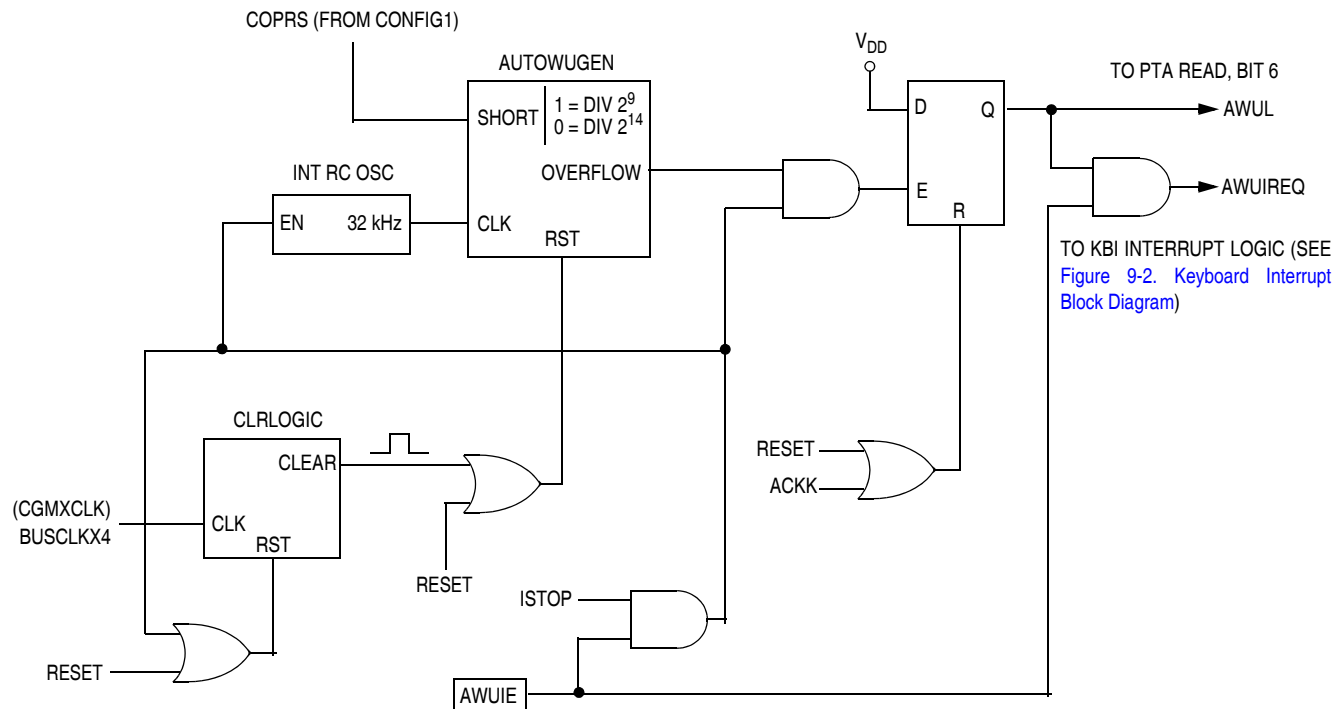


Figure 4-1. Auto Wakeup Interrupt Request Generation Logic

The auto wakeup RC oscillator is highly dependent on operating voltage and temperature. This feature is not recommended for use as a time-keeping function.

The wakeup request is latched to allow the interrupt source identification. The latched value, AWUL, can be read directly from the bit 6 position of PTA data register. This is a read-only bit which is occupying an empty bit position on PTA. No PTA associated registers, such as PTA6 data, PTA6 direction, and PTA6 pullup exist for this bit. The latch can be cleared by writing to the ACKK bit in the KBSCR register. Reset also clears the latch. AWUIE bit in KBI interrupt enable register (see [Figure 4-1](#)) has no effect on AWUL reading.

The AWU oscillator and counters are inactive in normal operating mode and become active only upon entering stop mode.

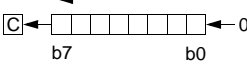
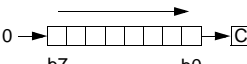
4.4 Wait Mode

The AWU module remains inactive in wait mode.

4.5 Stop Mode

When the AWU module is enabled (AWUIE = 1 in the keyboard interrupt enable register) it is activated automatically upon entering 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. The AWU counters start from '0' each time stop mode is entered.

Table 7-1. Instruction Set Summary (Sheet 4 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z				
JMP <i>opr</i> JMP <i>opr</i> JMP <i>opr</i> ,X JMP <i>opr</i> ,X JMP ,X	Jump	PC ← 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 ← (PC) + <i>n</i> (<i>n</i> = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← 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 ← (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 ← (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 ← (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 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
LSR <i>opr</i> LSRA LSRX LSR <i>opr</i> ,X LSR ,X LSR <i>opr</i> ,SP	Logical Shift Right		↑	–	–	0	↑	DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff ff	4 1 1 4 3 5
MOV <i>opr</i> , <i>opr</i> MOV <i>opr</i> ,X+ MOV # <i>opr</i> , <i>opr</i> MOV X+, <i>opr</i>	Move	(M) _{Destination} ← (M) _{Source} H:X ← (H:X) + 1 (IX+D, DIX+)	0	–	–	↑	↑	DD DIX+ IMD IX+D	4E 5E 6E 7E	dd dd dd ii dd dd	5 4 4 4
MUL	Unsigned multiply	X:A ← (X) × (A)	–	0	–	–	–	INH	42		5
NEG <i>opr</i> NEGA NEGX NEG <i>opr</i> ,X NEG ,X NEG <i>opr</i> ,SP	Negate (Two's Complement)	M ← –(M) = \$00 – (M) A ← –(A) = \$00 – (A) X ← –(X) = \$00 – (X) M ← –(M) = \$00 – (M) M ← –(M) = \$00 – (M)	↑	–	–	↑	↑	DIR INH INH IX1 IX SP1	30 40 50 60 70 9E60	dd ff ff ff	4 1 1 4 3 5
NOP	No Operation	None	–	–	–	–	–	INH	9D		1
NSA	Nibble Swap A	A ← (A[3:0]:A[7:4])	–	–	–	–	–	INH	62		3
ORA # <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ,X ORA <i>opr</i> ,X ORA ,X ORA <i>opr</i> ,SP ORA <i>opr</i> ,SP	Inclusive OR A and M	A ← (A) (M)	0	–	–	↑	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	AA BA CA DA EA FA 9EEA 9EDA	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
PSHA	Push A onto Stack	Push (A); SP ← (SP) – 1	–	–	–	–	–	INH	87		2
PSHH	Push H onto Stack	Push (H); SP ← (SP) – 1	–	–	–	–	–	INH	8B		2
PSHX	Push X onto Stack	Push (X); SP ← (SP) – 1	–	–	–	–	–	INH	89		2



Chapter 11

Oscillator Module (OSC)

11.1 Introduction

The oscillator module is used to provide a stable clock source for the microcontroller system and bus. The oscillator module generates two output clocks, BUSCLKX2 and BUSCLKX4. The BUSCLKX4 clock is used by the system integration module (SIM) and the computer operating properly module (COP). The BUSCLKX2 clock is divided by two in the SIM to be used as the bus clock for the microcontroller. Therefore the bus frequency will be one fourth of the BUSCLKX4 frequency.

11.2 Features

The oscillator has these four clock source options available:

1. Internal oscillator: An internally generated, fixed frequency clock, trimmable to $\pm 5\%$. This is the default option out of reset.
2. External oscillator: An external clock that can be driven directly into OSC1.
3. External RC: A built-in oscillator module (RC oscillator) that requires an external R connection only. The capacitor is internal to the chip.
4. External crystal: A built-in oscillator module (XTAL oscillator) that requires an external crystal or ceramic-resonator.

11.3 Functional Description

The oscillator contains these major subsystems:

- Internal oscillator circuit
- Internal or external clock switch control
- External clock circuit
- External crystal circuit
- External RC clock circuit

Chapter 13

System Integration Module (SIM)

13.1 Introduction

This section describes the system integration module (SIM), which supports up to 24 external and/or internal interrupts. Together with the central processor unit (CPU), the SIM controls all microcontroller unit (MCU) activities. A block diagram of the SIM is shown in [Figure 13-1](#). The SIM is a system state controller that coordinates CPU and exception timing.

The SIM is responsible for:

- Bus clock generation and control for CPU and peripherals
 - Stop/wait/reset/break entry and recovery
 - Internal clock control
- Master reset control, including power-on reset (POR) and computer operating properly (COP) timeout
- Interrupt control:
 - Acknowledge timing
 - Arbitration control timing
 - Vector address generation
- CPU enable/disable timing

Table 13-1. Signal Name Conventions

Signal Name	Description
BUSCLKX4	Buffered clock from the internal, RC or XTAL oscillator circuit.
BUSCLKX2	The BUSCLKX4 frequency divided by two. This signal is again divided by two in the SIM to generate the internal bus clocks (bus clock = $BUSCLKX4 \div 4$).
Address bus	Internal address bus
Data bus	Internal data bus
PORRST	Signal from the power-on reset module to the SIM
IRST	Internal reset signal
R/ \overline{W}	Read/write signal

Chapter 14

Timer Interface Module (TIM)

14.1 Introduction

This section describes the timer interface module (TIM). The TIM is a two-channel timer that provides a timing reference with input capture, output compare, and pulse-width-modulation functions. [Figure 14-2](#) is a block diagram of the TIM.

14.2 Features

Features of the TIM include the following:

- Two input capture/output compare channels
 - Rising-edge, falling-edge, or any-edge input capture trigger
 - Set, clear, or toggle output compare action
- Buffered and unbuffered pulse width modulation (PWM) signal generation
- Programmable TIM clock input
 - 7-frequency internal bus clock prescaler selection
 - External TIM clock input
- Free-running or modulo up-count operation
- Toggle any channel pin on overflow
- TIM counter stop and reset bits

14.3 Pin Name Conventions

The TIM shares two input/output (I/O) pins with two port A I/O pins. The full names of the TIM I/O pins are listed in [Table 14-1](#). The generic pin name appear in the text that follows.

Table 14-1. Pin Name Conventions

TIM Generic Pin Names:	TCH0	TCH1	TCLK
Full TIM Pin Names:	PTA0/TCH0	PTA1/TCH1	PTA2/TCLK

14.4.4.3 PWM Initialization

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

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

NOTE

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

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

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

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

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

14.5 Interrupts

The following TIM sources can generate interrupt requests:

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

CHxMAX — Channel x Maximum Duty Cycle Bit

When the TOVx bit is at a 1, setting the CHxMAX bit forces the duty cycle of buffered and unbuffered PWM signals to 100%. As Figure 14-8 shows, the CHxMAX bit takes effect in the cycle after it is set or cleared. The output stays at the 100% duty cycle level until the cycle after CHxMAX is cleared.

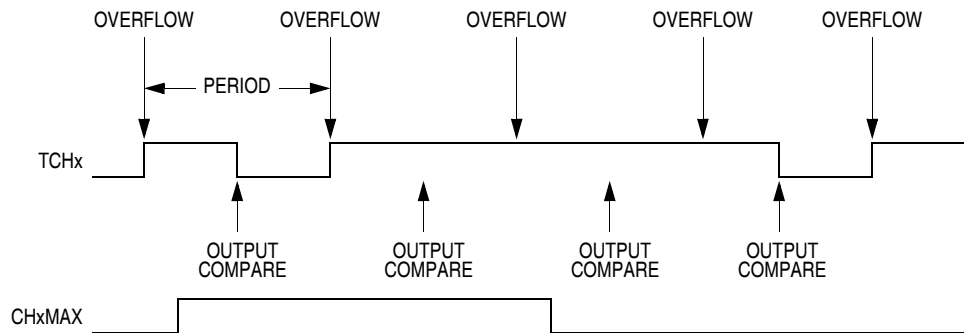


Figure 14-8. CHxMAX Latency

14.9.5 TIM Channel Registers

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

In input capture mode ($MSxB:MSxA = 0:0$), reading the high byte of the TIM channel x registers (TCHxH) inhibits input captures until the low byte (TCHxL) is read.

In output compare mode ($MSxB:MSxA \neq 0:0$), writing to the high byte of the TIM channel x registers (TCHxH) inhibits output compares until the low byte (TCHxL) is written.

Address: \$0026	TCH0H	Bit 7	6	5	4	3	2	1	Bit 0
Read:		Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:									
Reset:		Indeterminate after reset							

Address: \$0027	TCH0L	Bit 7	6	5	4	3	2	1	Bit 0
Read:		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:									
Reset:		Indeterminate after reset							

Address: \$0029	TCH1H	Bit 7	6	5	4	3	2	1	Bit 0
Read:		Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:									
Reset:		Indeterminate after reset							

Address: \$02A	TCH1L	Bit 7	6	5	4	3	2	1	Bit 0
Read:		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:									
Reset:		Indeterminate after reset							

Figure 14-9. TIM Channel Registers (TCH0H/L:TCH1H/L)

15.2.2.1 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 15-3. 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

15.2.2.2 Break Address Registers

The break address registers (BRKH and BRKL) contain the high and low bytes of the desired breakpoint address. Reset clears the break address registers.

Address: \$FE09

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

Figure 15-4. Break Address Register High (BRKH)

Address: \$FE0A

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

Figure 15-5. Break Address Register Low (BRKL)

15.2.2.5 Break Flag Control Register

The break control register (BFCR) contains a bit that enables software to clear status bits while the MCU is in a break state.

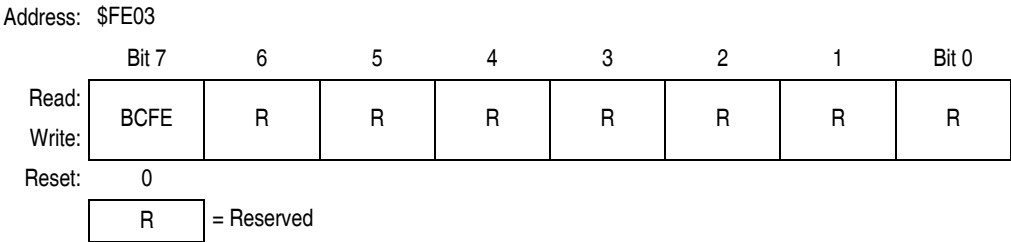


Figure 15-8. Break Flag Control Register (BFCR)

BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

- 1 = Status bits clearable during break
- 0 = Status bits not clearable during break

15.2.3 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power- consumption standby modes. If enabled, the break module will remain enabled in wait and stop modes. However, since the internal address bus does not increment in these modes, a break interrupt will never be triggered.

15.3 Monitor Module (MON)

This subsection describes the monitor module (MON) and the monitor mode entry methods. The monitor allows debugging and programming of the microcontroller unit (MCU) through a single-wire interface with a host computer. Monitor mode entry can be achieved without use of the higher test voltage, V_{TST} , as long as vector addresses \$FFFE and \$FFFF are blank, thus reducing the hardware requirements for in-circuit programming.

Features include:

- Normal user-mode pin functionality on most pins
- One pin dedicated to serial communication between MCU and host computer
- Standard non-return-to-zero (NRZ) communication with host computer
- Execution of code in random-access memory (RAM) or FLASH
- FLASH memory security feature⁽¹⁾
- FLASH memory programming interface
- Use of external 9.8304 MHz oscillator to generate internal frequency of 2.4576 MHz
- Simple internal oscillator mode of operation (no external clock or high voltage)
- Monitor mode entry without high voltage, V_{TST} , if reset vector is blank (\$FFFE and \$FFFF contain \$FF)
- Standard monitor mode entry if high voltage is applied to \overline{IRQ}

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.

Table 15-7. READSP (Read Stack Pointer) Command

Description	Reads stack pointer
Operand	None
Data Returned	Returns incremented stack pointer value (SP + 1) in high-byte:low-byte order
Opcode	\$0C
<p style="text-align: center;">Command Sequence</p>	

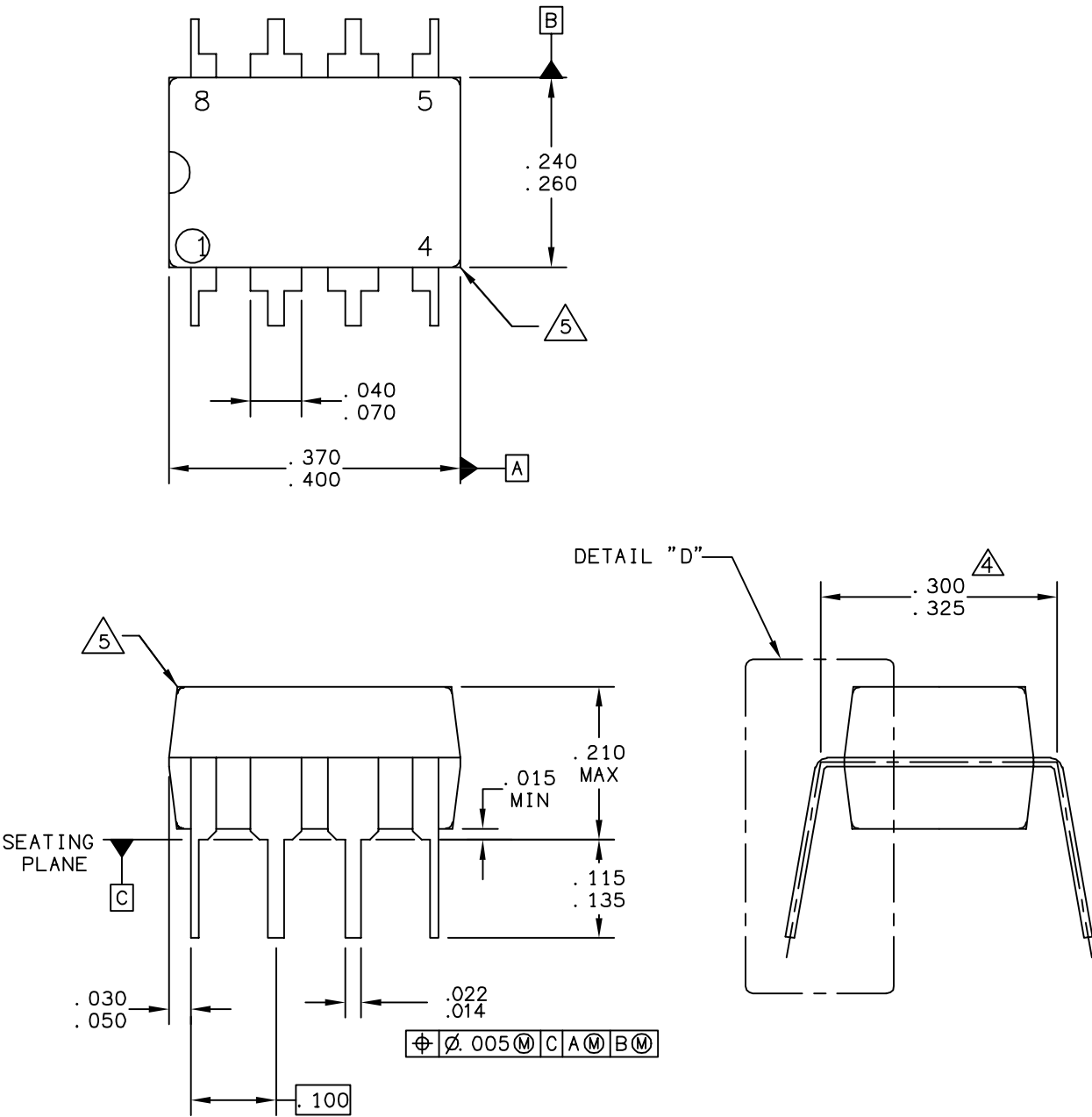
Table 15-8. RUN (Run User Program) Command

Description	Executes PULH and RTI instructions
Operand	None
Data Returned	None
Opcode	\$28
<p style="text-align: center;">Command Sequence</p>	

The MCU executes the SWI and PSHH instructions when it enters monitor mode. The RUN command tells the MCU to execute the PULH and RTI instructions. Before sending the RUN command, the host can modify the stacked CPU registers to prepare to run the host program. The READSP command returns the incremented stack pointer value, SP + 1. The high and low bytes of the program counter are at addresses SP + 5 and SP + 6.

	SP
HIGH BYTE OF INDEX REGISTER	SP + 1
CONDITION CODE REGISTER	SP + 2
ACCUMULATOR	SP + 3
LOW BYTE OF INDEX REGISTER	SP + 4
HIGH BYTE OF PROGRAM COUNTER	SP + 5
LOW BYTE OF PROGRAM COUNTER	SP + 6
	SP + 7

Figure 15-17. Stack Pointer at Monitor Mode Entry



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.		MECHANICAL OUTLINE		PRINT VERSION NOT TO SCALE	
TITLE: 8 LD PDIP			DOCUMENT NO: 98ASB42420B		REV: N
			CASE NUMBER: 626-06		19 MAY 2005
			STANDARD: NON-JEDEC		



NOTES:

- 1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M – 1994.
- 2. ALL DIMENSIONS ARE IN INCHES.
- 3. 626-03 TO 626-06 OBSOLETE. NEW STANDARD 626-07.

△ 4. DIMENSION TO CENTER OF LEAD WHEN FORMED PARALLEL.

△ 5. PACKAGE CONTOUR OPTIONAL (ROUND OR SQUARE CONERS).

STYLE 1:

PIN	1.	AC IN	5.	GROUND
	2.	DC + IN	6.	OUTPUT
	3.	DC – IN	7.	AUXILIARY
	4.	AC IN	8.	VCC

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.		MECHANICAL OUTLINE		PRINT VERSION NOT TO SCALE	
TITLE: 8 LD PDIP			DOCUMENT NO: 98ASB42420B		REV: N
			CASE NUMBER: 626-06		19 MAY 2005
			STANDARD: NON-JEDEC		



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.
ELECTRONIC VERSIONS ARE UNCONTROLLED EXCEPT WHEN ACCESSED
DIRECTLY FROM THE DOCUMENT CONTROL REPOSITORY. PRINTED
VERSIONS ARE UNCONTROLLED EXCEPT WHEN STAMPED "CONTROLLED
COPY" IN RED.

MECHANICAL OUTLINES DICTIONARY

DOCUMENT NO: 98ASH70107A

PAGE: 968

DO NOT SCALE THIS DRAWING

REV: A

NOTES:

1. DIMENSIONS AND TOLERANCES PER ASME Y14.5–1994.

2. CONTROLLING DIMENSION: MILLIMETER.

3. DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS AND ARE MEASURED AT THE PARTING LINE. MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED 0.15mm PER SIDE.

4. TERMINAL NUMBERS ARE SHOWN FOR REFERENCE ONLY.

5. THE LEAD WIDTH DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION SHALL BE 0.08mm TOTAL IN EXCESS OF THE LEAD WIDTH DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSIONS AND ADJACENT LEAD TO BE 0.46mm.

TITLE:

8 LEAD MFP

CASE NUMBER: 968–02

STANDARD: EIAJ

PACKAGE CODE: 6003

SHEET: 3 OF 4