



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	2MHz
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)	2.4V ~ 3.6V
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	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mcl908qy1cdwe">https://www.e-xfl.com/product-detail/nxp-semiconductors/mcl908qy1cdwe</a>

## Table of Contents

3.6	Input/Output Signals	43
3.7	Input/Output Registers	43
3.7.1	ADC Status and Control Register	43
3.7.2	ADC Data Register	44
3.7.3	ADC Input Clock Register	45

## Chapter 4 Auto Wakeup Module (AWU)

4.1	Introduction	47
4.2	Features	47
4.3	Functional Description	47
4.4	Wait Mode	49
4.5	Stop Mode	49
4.6	Input/Output Registers	49
4.6.1	Port A I/O Register	49
4.6.2	Keyboard Status and Control Register	50
4.6.3	Keyboard Interrupt Enable Register	51

## Chapter 5 Configuration Register (CONFIG)

5.1	Introduction	53
5.2	Functional Description	53

## Chapter 6 Computer Operating Properly (COP)

6.1	Introduction	57
6.2	Functional Description	57
6.3	I/O Signals	58
6.3.1	BUSCLKX4	58
6.3.2	STOP Instruction	58
6.3.3	COPCTL Write	58
6.3.4	Power-On Reset	58
6.3.5	Internal Reset	58
6.3.6	COPD (COP Disable)	58
6.3.7	COPRS (COP Rate Select)	59
6.4	COP Control Register	59
6.5	Interrupts	59
6.6	Monitor Mode	59
6.7	Low-Power Modes	59
6.7.1	Wait Mode	59
6.7.2	Stop Mode	59
6.8	COP Module During Break Mode	59

Features of the CPU08 include the following:

- Enhanced HC05 programming model
- Extensive loop control functions
- 16 addressing modes (eight more than the HC05)
- 16-bit index register and stack pointer
- Memory-to-memory data transfers
- Fast  $8 \times 8$  multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Efficient C language support

### 1.3 MCU Block Diagram

[Figure 1-1](#) shows the structure of the MC68HLC908QY4.

### 1.4 Pin Assignments

The MC68HLC908QT4, MC68HLC908QT2, and MC68HLC908QT1 are available in 8-pin packages and the MC68HLC908QY4, MC68HLC908QY2, and MC68HLC908QY1 in 16-pin packages. [Figure 1-2](#) shows the pin assignment for these packages.

## 1.5 Pin Functions

Table 1-2 provides a description of the pin functions.

**Table 1-2. Pin Functions**

Pin Name	Description	Input/Output
V <sub>DD</sub>	Power supply	Power
V <sub>SS</sub>	Power supply ground	Power
PTA0	PTA0 — General purpose I/O port	Input/Output
	AD0 — A/D channel 0 input	Input
	TCH0 — Timer Channel 0 I/O	Input/Output
	KBI0 — Keyboard interrupt input 0	Input
PTA1	PTA1 — General purpose I/O port	Input/Output
	AD1 — A/D channel 1 input	Input
	TCH1 — Timer Channel 1 I/O	Input/Output
	KBI1 — Keyboard interrupt input 1	Input
PTA2	PTA2 — General purpose input-only port	Input
	$\overline{\text{IRQ}}$ — External interrupt with programmable pullup and Schmitt trigger input	Input
	KBI2 — Keyboard interrupt input 2	Input
	TCLK — Timer clock input	Input
PTA3	PTA3 — General purpose I/O port	Input/Output
	$\overline{\text{RST}}$ — Reset input, active low with internal pullup and Schmitt trigger	Input
	KBI3 — Keyboard interrupt input 3	Input
PTA4	PTA4 — General purpose I/O port	Input/Output
	OSC2 — XTAL oscillator output (XTAL option only) RC or internal oscillator output (OSC2EN = 1 in PTAPUE register)	Output Output
	AD2 — A/D channel 2 input	Input
	KBI4 — Keyboard interrupt input 4	Input
PTA5	PTA5 — General purpose I/O port	Input/Output
	$\overline{\text{OSC1}}$ — XTAL, RC, or external oscillator input	Input
	AD3 — A/D channel 3 input	Input
	KBI5 — Keyboard interrupt input 5	Input
PTB[0:7] <sup>(1)</sup>	8 general-purpose I/O ports	Input/Output

1. The PTB pins are not available on the 8-pin packages (see note in [12.1 Introduction](#)).

### 2.6.3 FLASH Mass Erase Operation

Use the following procedure to erase the entire FLASH memory to read as a 1:

1. Set both the ERASE bit and the MASS bit in the FLASH control register.
2. Read the FLASH block protect register.
3. Write any data to any FLASH address<sup>(1)</sup> within the FLASH memory address range.
4. Wait for a time,  $t_{NVS}$  (minimum 10  $\mu$ s).
5. Set the HVEN bit.
6. Wait for a time,  $t_{MErase}$  (minimum 4 ms).
7. Clear the ERASE and MASS bits.

#### NOTE

*Mass erase is disabled whenever any block is protected (FLBPR does not equal \$FF).*

8. Wait for a time,  $t_{NVH}$  (minimum 100  $\mu$ s).
9. Clear the HVEN bit.
10. After time,  $t_{RCV}$  (typical 1  $\mu$ 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 mass erase will erase the internal oscillator trim value at \$FFC0.*

### 2.6.4 FLASH Program Operation

Programming of the FLASH memory is done on a row basis. A row consists of 32 consecutive bytes starting from addresses \$XX00, \$XX20, \$XX40, \$XX60, \$XX80, \$XXA0, \$XXC0, or \$XXE0. Use the following step-by-step procedure to program a row of FLASH memory

Figure 2-4 shows a flowchart of the programming algorithm.

#### NOTE

*Only bytes which are currently \$FF may be programmed.*

1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
2. Read the FLASH block protect register.
3. Write any data to any FLASH location within the address range desired.
4. Wait for a time,  $t_{NVS}$  (minimum 10  $\mu$ s).
5. Set the HVEN bit.
6. Wait for a time,  $t_{PGS}$  (minimum 5  $\mu$ s).
7. Write data to the FLASH address being programmed<sup>(2)</sup>.

1. When in monitor mode, with security sequence failed (see 15.3.2 Security), write to the FLASH block protect register instead of any FLASH address.
2. The time between each FLASH address change, or the time between the last FLASH address programmed to clearing PGM bit, must not exceed the maximum programming time,  $t_{PROG}$  maximum.

8. Wait for time,  $t_{\text{PROG}}$  (minimum 30  $\mu\text{s}$ ).
9. Repeat step 7 and 8 until all desired bytes within the row are programmed.
10. Clear the PGM bit<sup>(1)</sup>.
11. Wait for time,  $t_{\text{NVH}}$  (minimum 5  $\mu\text{s}$ ).
12. Clear the HVEN bit.
13. After time,  $t_{\text{RCV}}$  (typical 1  $\mu\text{s}$ ), the memory can be accessed in read mode again.

#### NOTE

*The COP register at location \$FFFF should not be written between steps 5-12, when the HVEN bit is set. Since this register is located at a valid FLASH address, unpredictable behavior may occur if this location is written while HVEN is set.*

This program sequence is repeated throughout the memory until all data is programmed.

#### 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 shown, other unrelated operations may occur between the steps. Do not exceed  $t_{\text{PROG}}$  maximum, see [16.12 Memory Characteristics](#).*

### 2.6.5 FLASH Protection

Due to the ability of the on-board charge pump to erase and program the FLASH memory in the target application, provision is made to protect blocks of memory from unintentional erase or program operations due to system malfunction. This protection is done by use of a FLASH block protect register (FLBPR). The FLBPR determines the range of the FLASH memory which is to be protected. The range of the protected area starts from a location defined by FLBPR and ends to the bottom of the FLASH memory (\$FFFF). When the memory is protected, the HVEN bit cannot be set in either ERASE or PROGRAM operations.

#### NOTE

*In performing a program or erase operation, the FLASH block protect register must be read after setting the PGM or ERASE bit and before asserting the HVEN bit.*

When the FLBPR is programmed with all 0s, the entire memory is protected from being programmed and erased. When all the bits are erased (all 1s), the entire memory is accessible for program and erase.

When bits within the FLBPR are programmed, they lock a block of memory. The address ranges are shown in [2.6.6 FLASH Block Protect Register](#). Once the FLBPR is programmed with a value other than \$FF, any erase or program of the FLBPR or the protected block of FLASH memory is prohibited. Mass erase is disabled whenever any block is protected (FLBPR does not equal \$FF). The FLBPR itself can be erased or programmed only with an external voltage,  $V_{\text{TST}}$ , present on the  $\overline{\text{IRQ}}$  pin. This voltage also allows entry from reset into the monitor mode.

### 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, or 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, or 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.*

## 3.6 Input/Output Signals

The ADC module has four channels that are shared with I/O port A.

ADC voltage in (ADCVIN) is the input voltage signal from one of the four ADC channels to the ADC module.

## 3.7 Input/Output Registers

These I/O registers control and monitor ADC operation:

- ADC status and control register (ADSCR)
- ADC data register (ADR)
- ADC clock register (ADICLK)

### 3.7.1 ADC Status and Control Register

The following paragraphs describe the function of the ADC status and control register (ADSCR). When a conversion is in process and the ADSCR is written, the current conversion data should be discarded to prevent an incorrect reading.

Address: \$003C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COCO	AIEN	ADCO	CH4	CH3	CH2	CH1	CH0
Write:	R							
Reset:	0	0	0	1	1	1	1	1
	R	= Reserved						

**Figure 3-3. ADC Status and Control Register (ADSCR)**

#### COCO — Conversions Complete Bit

In non-interrupt mode (AIEN = 0), COCO is a read-only bit that is set at the end of each conversion. COCO will stay set until cleared by a read of the ADC data register. Reset clears this bit.

In interrupt mode (AIEN = 1), COCO is a read-only bit that is not set at the end of a conversion. It always reads as a 0.

1 = Conversion completed (AIEN = 0)

0 = Conversion not completed (AIEN = 0) or CPU interrupt enabled (AIEN = 1)

#### NOTE

*The write function of the COCO bit is reserved. When writing to the ADSCR register, always have a 0 in the COCO bit position.*

#### AIEN — ADC Interrupt Enable Bit

When this bit is set, an interrupt is generated at the end of an ADC conversion. The interrupt signal is cleared when ADR is read or ADSCR is written. Reset clears the AIEN bit.

1 = ADC interrupt enabled

0 = ADC interrupt disabled

#### ADCO — ADC Continuous Conversion Bit

When set, the ADC will convert samples continuously and update ADR at the end of each conversion. Only one conversion is allowed when this bit is cleared. Reset clears the ADCO bit.

1 = Continuous ADC conversion

0 = One ADC conversion



# Chapter 5

## Configuration Register (CONFIG)

### 5.1 Introduction

This section describes the configuration registers (CONFIG1 and CONFIG2). The configuration registers enable or disable the following options:

- Stop mode recovery time ( $32 \times \text{BUSCLKX4}$  cycles or  $4096 \times \text{BUSCLKX4}$  cycles)
- STOP instruction
- Computer operating properly module (COP)
- COP reset period (COPRS):  $8176 \times \text{BUSCLKX4}$  or  $262,128 \times \text{BUSCLKX4}$
- Low-voltage inhibit (LVI) enable and trip voltage selection
- OSC option selection
- $\overline{\text{IRQ}}$  pin
- $\overline{\text{RST}}$  pin
- Auto wakeup timeout period

### 5.2 Functional Description

The configuration registers are used in the initialization of various options. The configuration registers can be written once after each reset. Exceptions are bits LVDLVR and LVIRSTD which may be written at any time. Most of the configuration register bits are cleared during reset. Since the various options affect the operation of the microcontroller unit (MCU) it is recommended that this register be written immediately after reset. The configuration registers are located at \$001E and \$001F, and may be read at anytime.

Address: \$001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IRQPUD	IRQEN	R	OSCOPT1	OSCOPT0	R	R	RSTEN
Write:								
Reset:	0	0	0	0	0	0	0	U
POR:	0	0	0	0	0	0	0	0

R = Reserved      U = Unaffected

**Figure 5-1. Configuration Register 2 (CONFIG2)**

### 6.3.7 COPRS (COP Rate Select)

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

## 6.4 COP Control Register

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

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 6-2. COP Control Register (COPCTL)**

## 6.5 Interrupts

The COP does not generate CPU interrupt requests.

## 6.6 Monitor Mode

The COP is disabled in monitor mode when  $V_{TST}$  is present on the  $\overline{IRQ}$  pin.

## 6.7 Low-Power Modes

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

### 6.7.1 Wait Mode

The COP continues to operate during wait mode. To prevent a COP reset during wait mode, periodically clear the COP counter.

### 6.7.2 Stop Mode

Stop mode turns off the BUSCLKX4 input to the COP and clears the SIM counter. Service the COP immediately before entering or after exiting stop mode to ensure a full COP timeout period after entering or exiting stop mode.

## 6.8 COP Module During Break Mode

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

**Z — Zero Flag**

The CPU sets the zero flag when an arithmetic operation, logic operation, or data manipulation produces a result of \$00.

1 = Zero result

0 = Non-zero result

**C — Carry/Borrow Flag**

The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some instructions — such as bit test and branch, shift, and rotate — also clear or set the carry/borrow flag.

1 = Carry out of bit 7

0 = No carry out of bit 7

## 7.4 Arithmetic/Logic Unit (ALU)

The ALU performs the arithmetic and logic operations defined by the instruction set.

Refer to the *CPU08 Reference Manual* (document order number CPU08RM/AD) for a description of the instructions and addressing modes and more detail about the architecture of the CPU.

## 7.5 Low-Power Modes

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

### 7.5.1 Wait Mode

The WAIT instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling interrupts. After exit from wait mode by interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

### 7.5.2 Stop Mode

The STOP instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling external interrupts. After exit from stop mode by external interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

After exiting stop mode, the CPU clock begins running after the oscillator stabilization delay.

## 7.6 CPU During Break Interrupts

If a break module is present on the MCU, the CPU starts a break interrupt by:

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

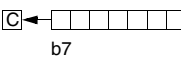
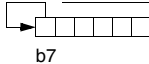
The break interrupt begins after completion of the CPU instruction in progress. If the break address register match occurs on the last cycle of a CPU instruction, the break interrupt begins immediately.

A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation if the break interrupt has been deasserted.

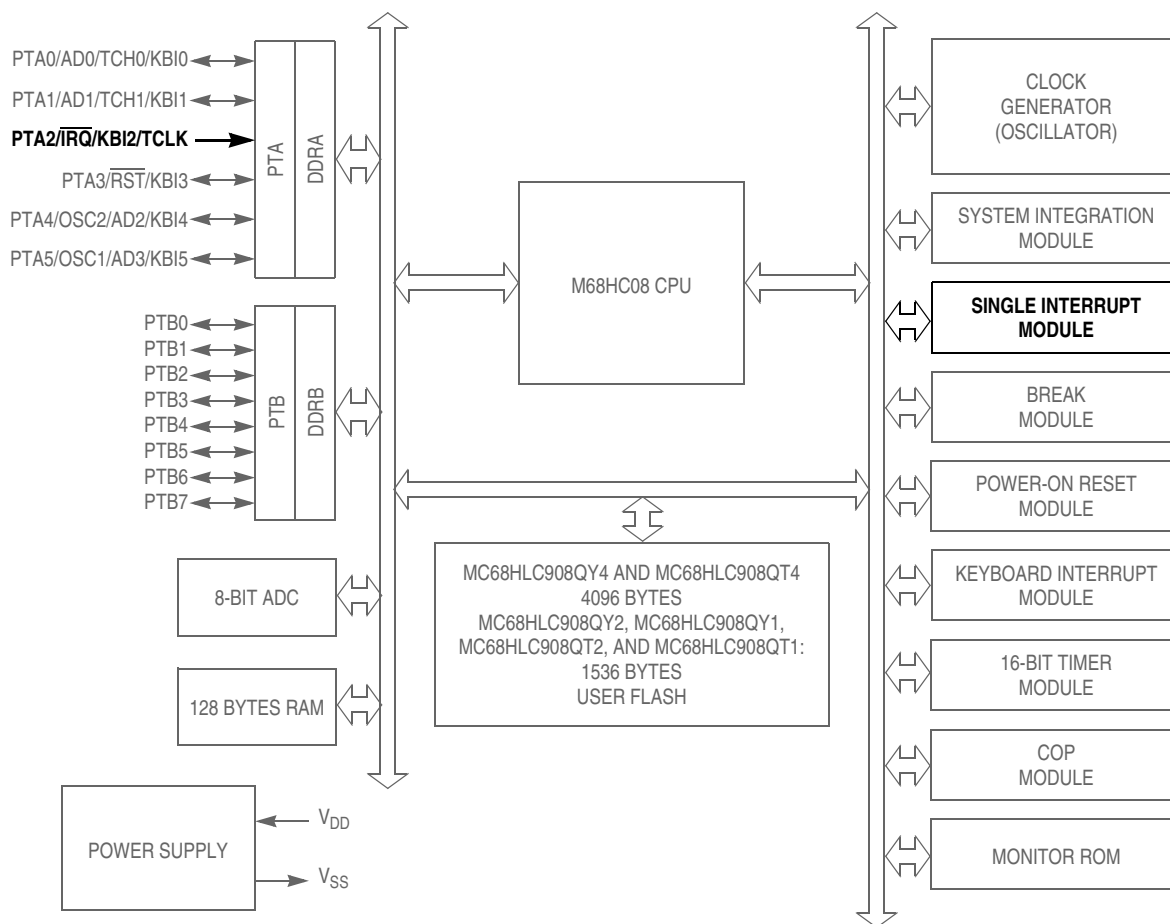
## 7.7 Instruction Set Summary

Table 7-1 provides a summary of the M68HC08 instruction set.

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

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ADC #opr ADC opr ADC opr ADC opr,X ADC opr,X ADC ,X ADC opr,SP ADC opr,SP	Add with Carry	$A \leftarrow (A) + (M) + (C)$	†	†	–	†	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	A9 B9 C9 D9 E9 F9 9EE9 9ED9	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
ADD #opr ADD opr ADD opr ADD opr,X ADD opr,X ADD ,X ADD opr,SP ADD opr,SP	Add without Carry	$A \leftarrow (A) + (M)$	†	†	–	†	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	AB BB CB DB EB FB 9EEB 9EDB	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
AIS #opr	Add Immediate Value (Signed) to SP	$SP \leftarrow (SP) + (16 \ll M)$	–	–	–	–	–	–	IMM	A7	ii	2
AIX #opr	Add Immediate Value (Signed) to H:X	$H:X \leftarrow (H:X) + (16 \ll M)$	–	–	–	–	–	–	IMM	AF	ii	2
AND #opr AND opr AND opr AND opr,X AND opr,X AND ,X AND opr,SP AND opr,SP	Logical AND	$A \leftarrow (A) \& (M)$	0	–	–	†	†	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A4 B4 C4 D4 E4 F4 9EE4 9ED4	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
ASL opr ASLA ASLX ASL opr,X ASL ,X ASL opr,SP	Arithmetic Shift Left (Same as LSL)		†	–	–	†	†	†	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff ff	4 1 1 4 3 5
ASR opr ASRA ASRX ASR opr,X ASR opr,X ASR opr,SP	Arithmetic Shift Right		†	–	–	†	†	†	DIR INH INH IX1 IX SP1	37 47 57 67 77 9E67	dd ff ff ff ff	4 1 1 4 3 5
BCC rel	Branch if Carry Bit Clear	$PC \leftarrow (PC) + 2 + rel ? (C) = 0$	–	–	–	–	–	–	REL	24	rr	3
BCLR n, opr	Clear Bit n in M	$M_n \leftarrow 0$	–	–	–	–	–	–	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	4 4 4 4 4 4 4 4
BCS rel	Branch if Carry Bit Set (Same as BLO)	$PC \leftarrow (PC) + 2 + rel ? (C) = 1$	–	–	–	–	–	–	REL	25	rr	3
BEQ rel	Branch if Equal	$PC \leftarrow (PC) + 2 + rel ? (Z) = 1$	–	–	–	–	–	–	REL	27	rr	3
BGE opr	Branch if Greater Than or Equal To (Signed Operands)	$PC \leftarrow (PC) + 2 + rel ? (N \oplus V) = 0$	–	–	–	–	–	–	REL	90	rr	3
BGT opr	Branch if Greater Than (Signed Operands)	$PC \leftarrow (PC) + 2 + rel ? (Z) \mid (N \oplus V) = 0$	–	–	–	–	–	–	REL	92	rr	3
BHCC rel	Branch if Half Carry Bit Clear	$PC \leftarrow (PC) + 2 + rel ? (H) = 0$	–	–	–	–	–	–	REL	28	rr	3
BHCS rel	Branch if Half Carry Bit Set	$PC \leftarrow (PC) + 2 + rel ? (H) = 1$	–	–	–	–	–	–	REL	29	rr	3
BHI rel	Branch if Higher	$PC \leftarrow (PC) + 2 + rel ? (C) \mid (Z) = 0$	–	–	–	–	–	–	REL	22	rr	3

## External Interrupt (IRQ)



**RST, IRQ:** Pins have internal (about 30K Ohms) pull up

**PTA[0:5]:** High current sink and source capability

**PTA[0:5]:** Pins have programmable keyboard interrupt and pull up

**PTB[0:7]:** Not available on 8-pin devices – MC68HLC908QT1, MC68HLC908QT2, and MC68HLC908QT4 (see note in [12.1 Introduction](#))

**ADC:** Not available on the MC68HLC908QY1 and MC68HLC908QT1

**Figure 8-1. Block Diagram Highlighting IRQ Block and Pins**

When set, the IMASK bit in INTSCR masks the  $\overline{\text{IRQ}}$  interrupt request. A latched interrupt request is not presented to the interrupt priority logic unless IMASK is clear.

### NOTE

*The interrupt mask (I) in the condition code register (CCR) masks all interrupt requests, including the  $\overline{\text{IRQ}}$  interrupt request.*

A falling edge on the  $\overline{\text{IRQ}}$  pin can latch an interrupt request into the IRQ latch. An IRQ vector fetch, software clear, or reset clears the IRQ latch.

## Chapter 9

# Keyboard Interrupt Module (KBI)

### 9.1 Introduction

The keyboard interrupt module (KBI) provides six independently maskable external interrupts, which are accessible via the PTA0–PTA5 pins.

### 9.2 Features

Features of the keyboard interrupt module include:

- Six keyboard interrupt pins with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Software configurable pullup device if input pin is configured as input port bit
- Programmable edge-only or edge and level interrupt sensitivity
- Exit from low-power modes

### 9.3 Functional Description

The keyboard interrupt module controls the enabling/disabling of interrupt functions on the six port A pins. These six pins can be enabled/disabled independently of each other.

#### 9.3.1 Keyboard Operation

Writing to the KBIE0–KBIE5 bits in the keyboard interrupt enable register (KBIER) independently enables or disables each port A pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin in port A also enables its internal pullup device irrespective of PTAPUEx bits in the port A input pullup enable register (see [12.2.3 Port A Input Pullup Enable Register](#)). A logic 0 applied to an enabled keyboard interrupt pin latches a keyboard interrupt request.

A keyboard interrupt is latched when one or more keyboard interrupt inputs goes low after all were high. The MODEK bit in the keyboard status and control register controls the triggering mode of the keyboard interrupt.


- If the keyboard interrupt is edge-sensitive only, a falling edge on a keyboard interrupt input does not latch an interrupt request if another keyboard pin is already low. To prevent losing an interrupt request on one input because another input is still low, software can disable the latter input while it is low.
- If the keyboard interrupt is falling edge and low-level sensitive, an interrupt request is present as long as any keyboard interrupt input is low.

### 9.7.2 Keyboard Interrupt Enable Register

The port A keyboard interrupt enable register (KBIER) enables or disables each port A pin or auto wakeup to operate as a keyboard interrupt input.

Address: \$001B

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

 = Unimplemented

**Figure 9-4. Keyboard Interrupt Enable Register (KBIER)**

#### KBIE5–KBIE0 — Port A Keyboard Interrupt Enable Bits

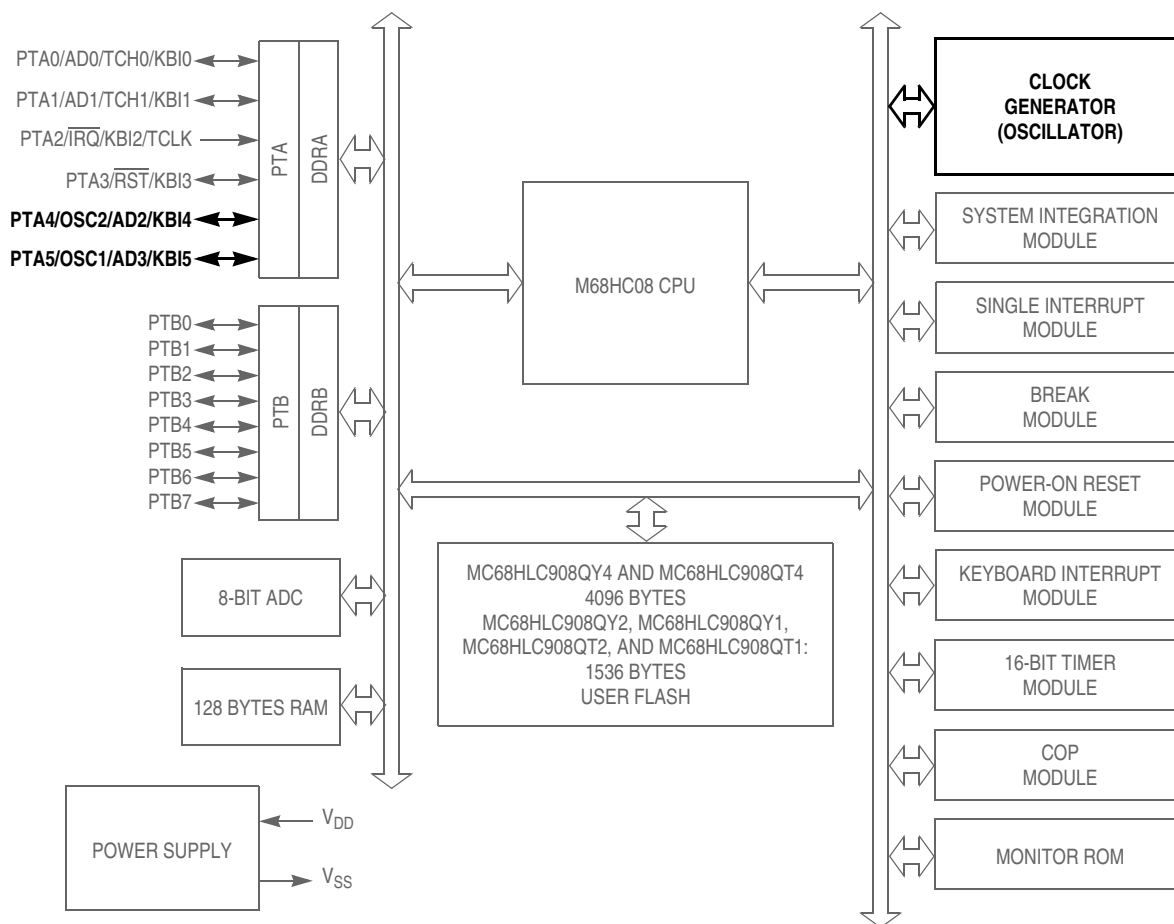
Each of these read/write bits enables the corresponding keyboard interrupt pin on port A to latch interrupt requests. Reset clears the keyboard interrupt enable register.

- 1 = KBIx pin enabled as keyboard interrupt pin
- 0 = KBIx pin not enabled as keyboard interrupt pin

**NOTE**

*AWUIE bit is not used in conjunction with the keyboard interrupt feature. To see a description of this bit, see [Chapter 4 Auto Wakeup Module \(AWU\)](#).*

## Oscillator Module (OSC)



$\overline{RST}$ ,  $\overline{IRQ}$ : Pins have internal (about 30K Ohms) pull up

**PTA[0:5]: High current sink and source capability**

**PTA[0:5]: Pins have programmable keyboard interrupt and pull up**

PTB[0:7]: Not available on 8-pin devices – MC68HLC908QT1, MC68HLC908QT2, and MC68HLC908QT4 (see note in [12.1 Introduction](#))

ADC: Not available on the MC68HLC908QY1 and MC68HLC908QT1

**Figure 11-1. Block Diagram Highlighting OSC Block and Pins**



### 11.3.1.1 Internal Oscillator Trimming

The 8-bit trimming register, OSCTRIM, allows a clock period adjust of +127 and –128 steps. Increasing OSCTRIM value increases the clock period. Trimming allows the internal clock frequency to be set to 4.0 MHz  $\pm 5\%$ .

All devices are programmed with a trim value in a reserved FLASH location, \$FFC0. This value can be copied from the FLASH to the OSCTRIM register (\$0038) during reset initialization.

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

#### **WARNING**

**Bulk FLASH erasure will set location \$FFC0 to \$FF and the factory programmed value will be lost.**

### 11.3.1.2 Internal to External Clock Switching

When external clock source (external OSC, RC, or XTAL) is desired, the user must perform the following steps:

1. For external crystal circuits only, OSCOPT[1:0] = 1:1: To help precharge an external crystal oscillator, set PTA4 (OSC2) as an output and drive high for several cycles. This may help the crystal circuit start more robustly.
2. Set CONFIG2 bits OSCOPT[1:0] according to . The oscillator module control logic will then set OSC1 as an external clock input and, if the external crystal option is selected, OSC2 will also be set as the clock output.
3. Create a software delay to wait the stabilization time needed for the selected clock source (crystal, resonator, RC) as recommended by the component manufacturer. A good rule of thumb for crystal oscillators is to wait 4096 cycles of the crystal frequency, i.e., for a 4-MHz crystal, wait approximately 1 msec.
4. After the manufacturer's recommended delay has elapsed, the ECGON bit in the OSC status register (OSCSTAT) needs to be set by the user software.
5. After ECGON set is detected, the OSC module checks for oscillator activity by waiting two external clock rising edges.
6. The OSC module then switches to the external clock. Logic provides a glitch free transition.
7. The OSC module first sets the ECGST bit in the OSCSTAT register and then stops the internal oscillator.

#### **NOTE**

*Once transition to the external clock is done, the internal oscillator will only be reactivated with reset. No post-switch clock monitor feature is implemented (clock does not switch back to internal if external clock dies).*

### 11.3.2 External Oscillator

The external clock option is designed for use when a clock signal is available in the application to provide a clock source to the microcontroller. The OSC1 pin is enabled as an input by the oscillator module. The clock signal is used directly to create BUSCLKX4 and also divided by two to create BUSCLKX2.

In this configuration, the OSC2 pin cannot output BUSCLKX4. So the OSC2EN bit in the port A pullup enable register will be clear to enable PTA4 I/O functions on the pin.

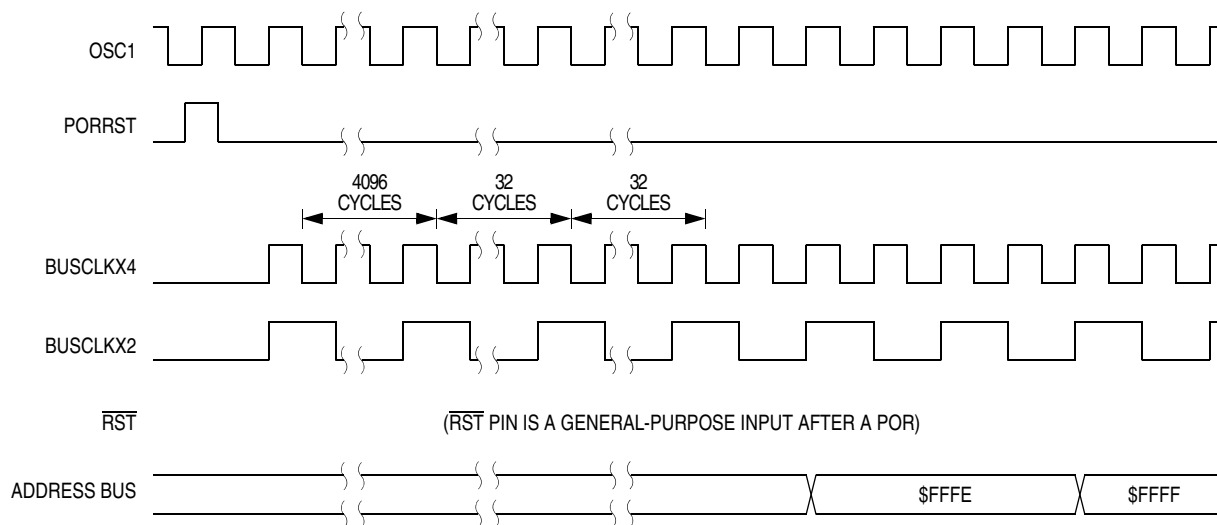
### 13.4.2.1 Power-On Reset

When power is first applied to the MCU, the power-on reset module (POR) generates a pulse to indicate that power on has occurred. The SIM counter counts out 4096 BUSCLKX4 cycles. Sixty-four BUSCLKX4 cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur.

At power on, the following events occur:

- A POR pulse is generated.
- The internal reset signal is asserted.
- The SIM enables the oscillator to drive BUSCLKX4.
- Internal clocks to the CPU and modules are held inactive for 4096 BUSCLKX4 cycles to allow stabilization of the oscillator.
- The POR bit of the SIM reset status register (SRSR) is set.

See [Figure 13-6](#).



**Figure 13-6. POR Recovery**

### 13.4.2.2 Computer Operating Properly (COP) Reset

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

To prevent a COP module time out, write any value to location \$FFFF. Writing to location \$FFFF clears the COP counter and stages 12–5 of the SIM counter. The SIM counter output, which occurs at least every 4080 BUSCLKX4 cycles, drives the COP counter. The COP should be serviced as soon as possible out of reset to guarantee the maximum amount of time before the first time out.

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

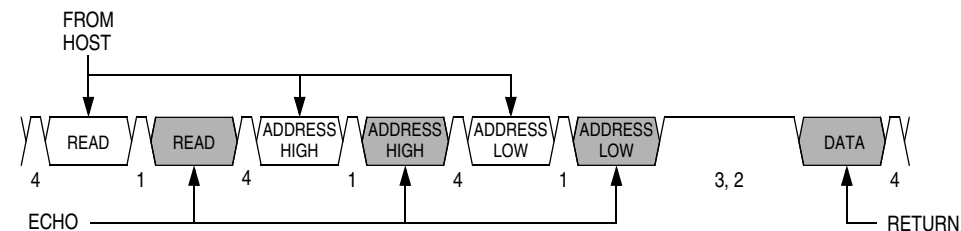
Address: \$0026	TCH0H							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:								
Reset:	Indeterminate after reset							

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							

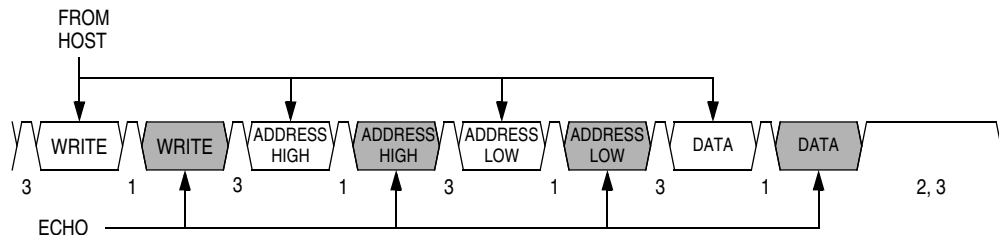
Address: \$02A	TCH1L							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:								
Reset:	Indeterminate after reset							

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



Notes:  
 1 = Echo delay, approximately 2 bit times  
 2 = Data return delay, approximately 2 bit times  
 3 = Cancel command delay, 11 bit times  
 4 = Wait 1 bit time before sending next byte.

**Figure 15-15. Read Transaction**



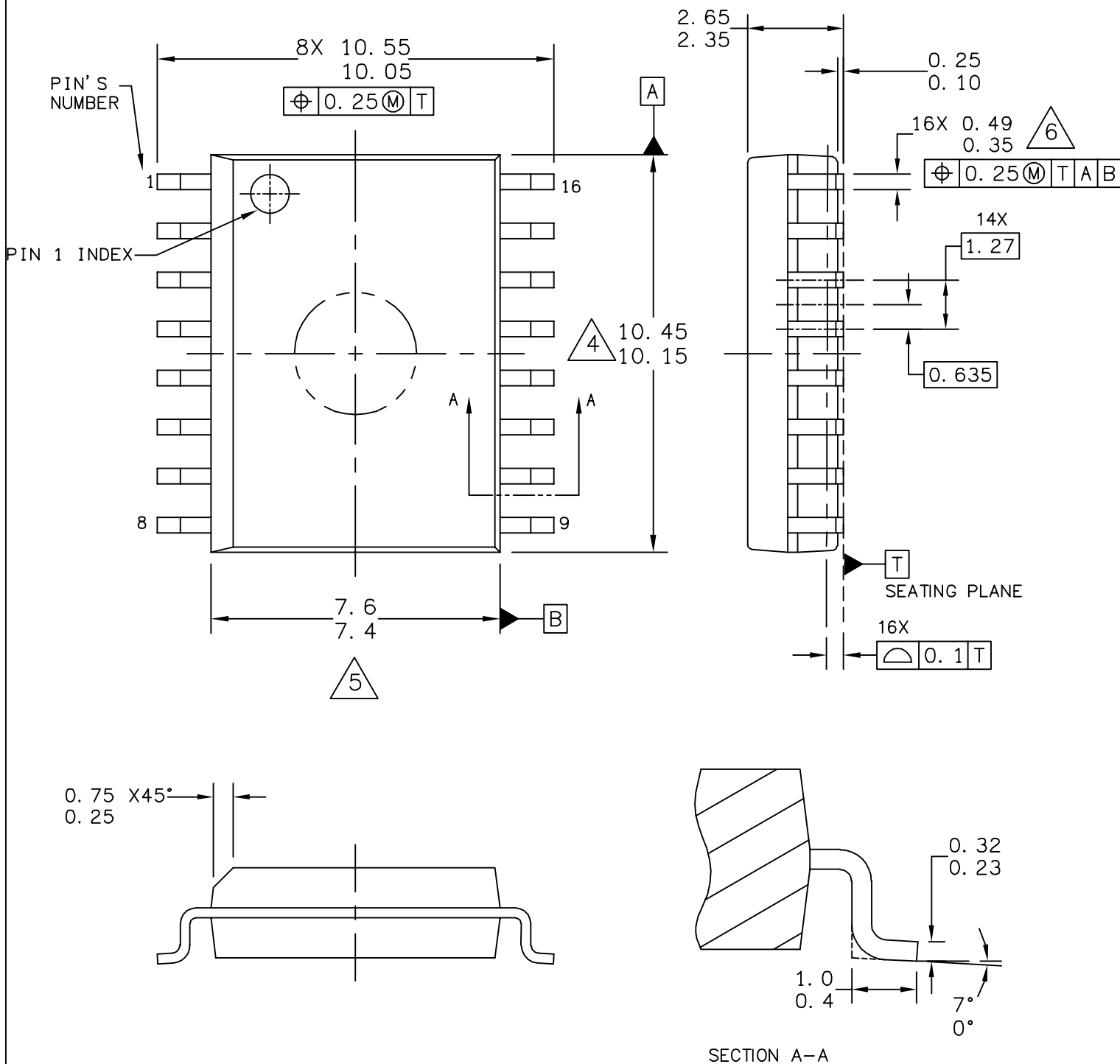
Notes:  
 1 = Echo delay, approximately 2 bit times  
 2 = Cancel command delay, 11 bit times  
 3 = Wait 1 bit time before sending next byte.

**Figure 15-16. Write Transaction**

A brief description of each monitor mode command is given in [Table 15-3](#) through [Table 15-8](#).

**Table 15-3. READ (Read Memory) Command**

<b>Description</b>	Read byte from memory
<b>Operand</b>	2-byte address in high-byte:low-byte order
<b>Data Returned</b>	Returns contents of specified address
<b>Opcode</b>	\$4A
<p style="text-align: center;"><b>Command Sequence</b></p>	



TITLE:  
16LD SOIC W/B, 1.27 PITCH  
CASE-OUTLINE

CASE NUMBER: 751G-05

STANDARD: JEDEC MS-013AA

PACKAGE CODE: 2003

SHEET: 1 OF 3