

Welcome to [E-XFL.COM](http://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	Active
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	A/D 4x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	16-DIP (0.300", 7.62mm)
Supplier Device Package	16-PDIP
Purchase URL	<a href="https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mc68hc908qy2cp">https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=mc68hc908qy2cp</a>

## Revision History (Sheet 2 of 3)

Date	Revision Level	Description	Page Number(s)
August, 2003	1.0	Reformatted to meet latest M68HC08 documentation standards	N/A
		Figure 1-1. Block Diagram — Diagram redrawn to include keyboard interrupt module and TCLK pin designator.	20
		Figure 1-2. MCU Pin Assignments — Added TCLK pin designator.	21
		Table 1-2. Pin Functions — Added TCLK pin description.	22
		Table 1-3. Function Priority in Shared Pins — Revised table for clarity and to add TCLK.	23
		Figure 2-1. Memory Map — Corrected names for the IRQ status and control register (INTSCR) bits 3–0.	26
		3.7.3 ADC Input Clock Register — Clarified bit description for the ADC clock prescaler bits.	47
		4.3 Functional Description — Updated periodic wakeup request values.	51
		Figure 6-1. COP Block Diagram — Reworked for clarity	59
		Chapter 8 External Interrupt (IRQ) — Corrected bit names for MODE, IRQF, ACK, and IMASK	77–79
		Chapter 14 Timer Interface Module (TIM) — Added TCLK function.	131–139
		15.3 Monitor Module (MON) — Updated with additional data.	147
Chapter 16 Electrical Specifications — Updated with additional data.	169–173		
October, 2003	2.0	Figure 2-2. Control, Status, and Data Registers — Deleted unimplemented areas from \$FFB0–\$FFBD and \$FFC2–\$FFCF as they are actually available. Also corrected \$FFBF designation from unimplemented to reserved.	27
		Figure 6-1. COP Block Diagram — Reworked for clarity	59
		6.3.2 STOP Instruction — Added subsection	60
		13.4.2 Active Resets from Internal Sources — Reworked notes for clarity.	111
		Table 13-2. Reset Recovery Timing — Replaced previous table with new information.	112
		Chapter 14 Timer Interface Module (TIM) — Updated with additional data.	131
		Figure 15-3. Break I/O Register Summary — Corrected bit designators for the BRKAR register	143
		15.3 Monitor Module (MON) — Clarified seventh bullet.	147
Table 17-1. MC Order Numbers — Corrected temperature and package designators.	175		
January, 2004	3.0	Figure 2-2. Control, Status, and Data Registers — Corrected reset state for the FLASH Block Protect Register at address location \$FFBE and the Internal Oscillator Trim Value at \$FFC0.	32
		Figure 2-5. FLASH Block Protect Register (FLBPR) — Restated reset state for clarity.	38

## Table of Contents

3.6	Input/Output Signals	45
3.7	Input/Output Registers	45
3.7.1	ADC Status and Control Register	45
3.7.2	ADC Data Register	47
3.7.3	ADC Input Clock Register	47

### Chapter 4 Auto Wakeup Module (AWU)

4.1	Introduction	49
4.2	Features	49
4.3	Functional Description	49
4.4	Wait Mode	50
4.5	Stop Mode	50
4.6	Input/Output Registers	51
4.6.1	Port A I/O Register	51
4.6.2	Keyboard Status and Control Register	51
4.6.3	Keyboard Interrupt Enable Register	52

### 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

## Chapter 7 Central Processor Unit (CPU)

7.1	Introduction .....	61
7.2	Features .....	61
7.3	CPU Registers .....	61
7.3.1	Accumulator .....	62
7.3.2	Index Register .....	62
7.3.3	Stack Pointer .....	63
7.3.4	Program Counter .....	63
7.3.5	Condition Code Register .....	64
7.4	Arithmetic/Logic Unit (ALU) .....	65
7.5	Low-Power Modes .....	65
7.5.1	Wait Mode .....	65
7.5.2	Stop Mode .....	65
7.6	CPU During Break Interrupts .....	65
7.7	Instruction Set Summary .....	66
7.8	Opcode Map .....	71

## Chapter 8 External Interrupt (IRQ)

8.1	Introduction .....	73
8.2	Features .....	73
8.3	Functional Description .....	73
8.3.1	MODE = 1 .....	75
8.3.2	MODE = 0 .....	75
8.4	Interrupts .....	76
8.5	Low-Power Modes .....	76
8.5.1	Wait Mode .....	76
8.5.2	Stop Mode .....	76
8.6	IRQ Module During Break Interrupts .....	76
8.7	I/O Signals .....	76
8.7.1	IRQ Input Pins ( $\overline{\text{IRQ}}$ ) .....	77
8.8	Registers .....	77

## Chapter 9 Keyboard Interrupt Module (KBI)

9.1	Introduction .....	79
9.2	Features .....	79
9.3	Functional Description .....	79
9.3.1	Keyboard Operation .....	79
9.3.2	Keyboard Initialization .....	82
9.4	Wait Mode .....	82
9.5	Stop Mode .....	82
9.6	Keyboard Module During Break Interrupts .....	82



### 3.5.2 Stop Mode

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

## 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.*

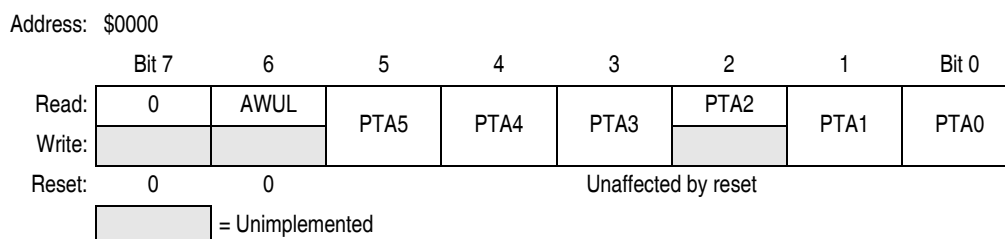
## 4.6 Input/Output Registers

The AWU shares registers with the keyboard interrupt (KBI) module and the port A I/O module. The following I/O registers control and monitor operation of the AWU:

- Port A data register (PTA)
- Keyboard interrupt status and control register (KBSCR)
- Keyboard interrupt enable register (KBIER)

### 4.6.1 Port A I/O Register

The port A data register (PTA) contains a data latch for the state of the AWU interrupt request, in addition to the data latches for port A.



**Figure 4-2. Port A Data Register (PTA)**

#### AWUL — Auto Wakeup Latch

This is a read-only bit which has the value of the auto wakeup interrupt request latch. The wakeup request signal is generated internally. There is no PTA6 port or any of the associated bits such as PTA6 data direction or pullup bits.

- 1 = Auto wakeup interrupt request is pending
- 0 = Auto wakeup interrupt request is not pending

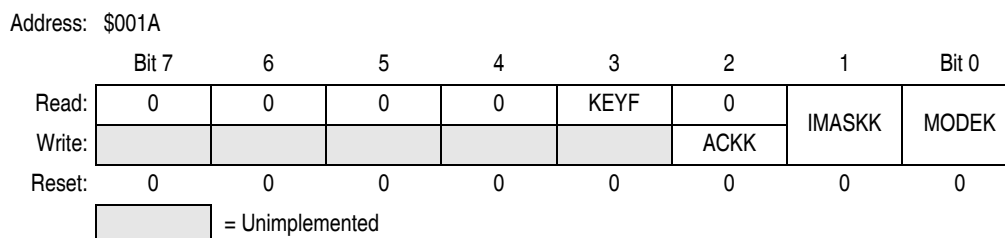
#### NOTE

*PTA5–PTA0 bits are not used in conjunction with the auto wakeup feature. To see a description of these bits, see [12.2.1 Port A Data Register](#).*

### 4.6.2 Keyboard Status and Control Register

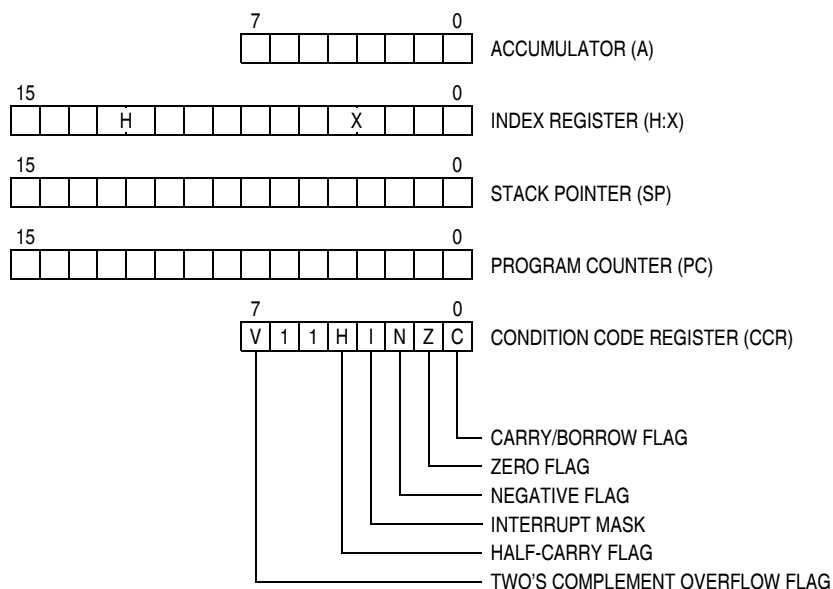
The keyboard status and control register (KBSCR):

- Flags keyboard/auto wakeup interrupt requests
- Acknowledges keyboard/auto wakeup interrupt requests
- Masks keyboard/auto wakeup interrupt requests



**Figure 4-3. Keyboard Status and Control Register (KBSCR)**

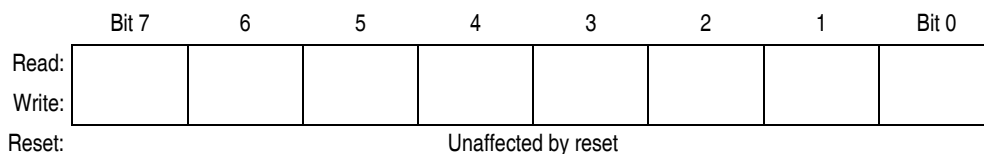
## Central Processor Unit (CPU)



**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.



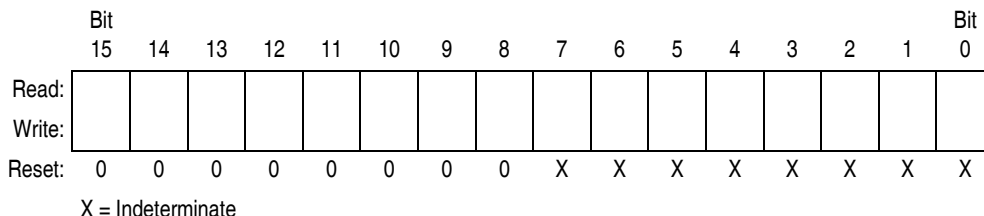
**Figure 7-2. Accumulator (A)**

### 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.



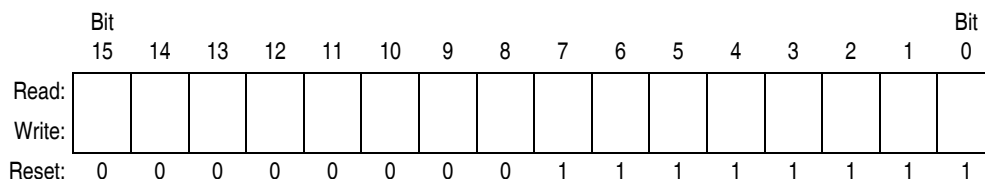
**Figure 7-3. Index Register (H:X)**



### 7.3.3 Stack Pointer

The stack pointer is a 16-bit register that contains the address of the next location on the stack. During a reset, the stack pointer is preset to \$00FF. The reset stack pointer (RSP) instruction sets the least significant byte to \$FF and does not affect the most significant byte. The stack pointer decrements as data is pushed onto the stack and increments as data is pulled from the stack.

In the stack pointer 8-bit offset and 16-bit offset addressing modes, the stack pointer can function as an index register to access data on the stack. The CPU uses the contents of the stack pointer to determine the conditional address of the operand.



**Figure 7-4. Stack Pointer (SP)**

**NOTE**

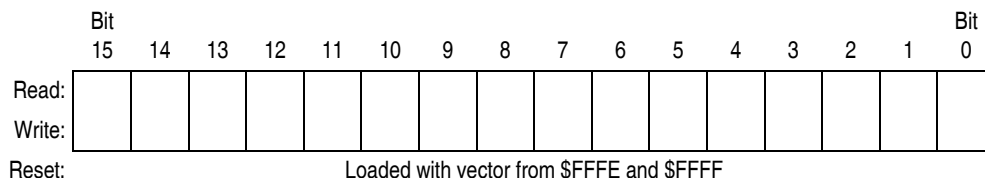
*The location of the stack is arbitrary and may be relocated anywhere in random-access memory (RAM). Moving the SP out of page 0 (\$0000 to \$00FF) frees direct address (page 0) space. For correct operation, the stack pointer must point only to RAM locations.*

### 7.3.4 Program Counter

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched.

Normally, the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

During reset, the program counter is loaded with the reset vector address located at \$FFFE and \$FFFF. The vector address is the address of the first instruction to be executed after exiting the reset state.



**Figure 7-5. Program Counter (PC)**

Table 7-1. Instruction Set Summary (Sheet 3 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles	
			V	H	I	N	Z					C
CLI	Clear Interrupt Mask	$I \leftarrow 0$	-	-	0	-	-	-	INH	9A		2
CLR <i>opr</i> CLRA CLR $X$ CLR $H$ CLR <i>opr,X</i> CLR $,X$ CLR <i>opr,SP</i>	Clear	$M \leftarrow \$00$ $A \leftarrow \$00$ $X \leftarrow \$00$ $H \leftarrow \$00$ $M \leftarrow \$00$ $M \leftarrow \$00$ $M \leftarrow \$00$	0	-	-	0	1	-	DIR INH INH INH IX1 IX SP1	3F 4F 5F 8C 6F 7F 9E6F	dd  ff ff	3 1 1 1 3 2 4
CMP # <i>opr</i> CMP <i>opr</i> CMP <i>opr</i> CMP <i>opr,X</i> CMP <i>opr,X</i> CMP $,X$ CMP <i>opr,SP</i> CMP <i>opr,SP</i>	Compare A with M	$(A) - (M)$	†	-	-	†	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	A1 B1 C1 D1 E1 F1 9EE1 9ED1	ii dd hh ll ee ff ff ff ff ff	2 3 4 4 3 2 4 5
COM <i>opr</i> COMA COM $X$ COM <i>opr,X</i> COM $,X$ COM <i>opr,SP</i>	Complement (One's Complement)	$M \leftarrow (\overline{M}) = \$FF - (M)$ $A \leftarrow (\overline{A}) = \$FF - (M)$ $X \leftarrow (\overline{X}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$	0	-	-	†	†	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd  ff ff ff	4 1 1 4 3 5
CPHX # <i>opr</i> CPHX <i>opr</i>	Compare H:X with M	$(H:X) - (M:M + 1)$	†	-	-	†	†	†	IMM DIR	65 75	ii ii+1 dd	3 4
CPX # <i>opr</i> CPX <i>opr</i> CPX <i>opr</i> CPX $,X$ CPX <i>opr,X</i> CPX <i>opr,X</i> CPX <i>opr,SP</i> CPX <i>opr,SP</i>	Compare X with M	$(X) - (M)$	†	-	-	†	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	A3 B3 C3 D3 E3 F3 9EE3 9ED3	ii dd hh ll ee ff ff ff ff	2 3 4 4 3 2 4 5
DAA	Decimal Adjust A	$(A)_{10}$	U	-	-	†	†	†	INH	72		2
DBNZ <i>opr,rel</i> DBNZ $A,rel$ DBNZ $X,rel$ DBNZ <i>opr,X,rel</i> DBNZ $X,rel$ DBNZ <i>opr,SP,rel</i>	Decrement and Branch if Not Zero	$A \leftarrow (A) - 1$ or $M \leftarrow (M) - 1$ or $X \leftarrow (X) - 1$ $PC \leftarrow (PC) + 3 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 2 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 2 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 3 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 2 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 4 + rel ? (result) \neq 0$	-	-	-	-	-	-	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	5 3 3 5 4 6
DEC <i>opr</i> DECA DEC $X$ DEC <i>opr,X</i> DEC $,X$ DEC <i>opr,SP</i>	Decrement	$M \leftarrow (M) - 1$ $A \leftarrow (A) - 1$ $X \leftarrow (X) - 1$ $M \leftarrow (M) - 1$ $M \leftarrow (M) - 1$ $M \leftarrow (M) - 1$	†	-	-	†	†	-	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd  ff ff	4 1 1 4 3 5
DIV	Divide	$A \leftarrow (H:A)/(X)$ $H \leftarrow \text{Remainder}$	-	-	-	-	†	†	INH	52		7
EOR # <i>opr</i> EOR <i>opr</i> EOR <i>opr</i> EOR <i>opr,X</i> EOR <i>opr,X</i> EOR $,X$ EOR <i>opr,SP</i> EOR <i>opr,SP</i>	Exclusive OR M with A	$A \leftarrow (A \oplus M)$	0	-	-	†	†	-	IMM DIR EXT IX2 IX1 IX SP1 SP2	A8 B8 C8 D8 E8 F8 9EE8 9ED8	ii dd hh ll ee ff ff ff ff	2 3 4 4 3 2 4 5
INC <i>opr</i> INCA INC $X$ INC <i>opr,X</i> INC $,X$ INC <i>opr,SP</i>	Increment	$M \leftarrow (M) + 1$ $A \leftarrow (A) + 1$ $X \leftarrow (X) + 1$ $M \leftarrow (M) + 1$ $M \leftarrow (M) + 1$ $M \leftarrow (M) + 1$	†	-	-	†	†	-	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd  ff ff	4 1 1 4 3 5

## 8.4 Interrupts

The following IRQ source can generate interrupt requests:

- Interrupt flag (IRQF) — The IRQF bit is set when the  $\overline{IRQ}$  pin is asserted based on the IRQ mode. The IRQ interrupt mask bit, IMASK, is used to enable or disable IRQ interrupt requests.

## 8.5 Low-Power Modes

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

### 8.5.1 Wait Mode

The IRQ module remains active in wait mode. Clearing IMASK in INTSCR enables IRQ interrupt requests to bring the MCU out of wait mode.

### 8.5.2 Stop Mode

The IRQ module remains active in stop mode. Clearing IMASK in INTSCR enables IRQ interrupt requests to bring the MCU out of stop mode.

## 8.6 IRQ Module During Break Interrupts

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state. See [Chapter 13 System Integration Module \(SIM\)](#).

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

To protect status bits during the break state, write a 0 to BCFE. With BCFE cleared (its default state), software can read and write registers during the break state without affecting status bits. Some status bits have a two-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is cleared. After the break, doing the second step clears the status bit.

## 8.7 I/O Signals

The IRQ module shares its pin with the keyboard interrupt, input/output ports, and timer interface modules.

### NOTE

*When the  $\overline{IRQ}$  function is enabled in the CONFIG2 register, the BIH and BIL instructions can be used to read the logic level on the  $\overline{IRQ}$  pin. If the  $\overline{IRQ}$  function is disabled, these instructions will behave as if the  $\overline{IRQ}$  pin is a logic 1, regardless of the actual level on the pin. Conversely, when the  $\overline{IRQ}$  function is enabled, bit 2 of the port A data register will always read a 0.*


*When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine. An internal pullup resistor to  $V_{DD}$  is connected to the  $\overline{IRQ}$  pin; this can be disabled by setting the IRQPUD bit in the CONFIG2 register (\$001E).*

### 13.8.1 SIM Reset Status Register

The SRSR register contains flags that show the source of the last reset. The status register will automatically clear after reading SRSR. A power-on reset sets the POR bit and clears all other bits in the register. All other reset sources set the individual flag bits but do not clear the register. More than one reset source can be flagged at any time depending on the conditions at the time of the internal or external reset. For example, the POR and LVI bit can both be set if the power supply has a slow rise time.

Address: \$FE01

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
Write:								
POR:	1	0	0	0	0	0	0	0

 = Unimplemented

**Figure 13-19. SIM Reset Status Register (SRSR)**

#### **POR — Power-On Reset Bit**

- 1 = Last reset caused by POR circuit
- 0 = Read of SRSR

#### **PIN — External Reset Bit**

- 1 = Last reset caused by external reset pin ( $\overline{RST}$ )
- 0 = POR or read of SRSR

#### **COP — Computer Operating Properly Reset Bit**

- 1 = Last reset caused by COP counter
- 0 = POR or read of SRSR

#### **ILOP — Illegal Opcode Reset Bit**

- 1 = Last reset caused by an illegal opcode
- 0 = POR or read of SRSR

#### **ILAD — Illegal Address Reset Bit (illegal attempt to fetch an opcode from an unimplemented address)**

- 1 = Last reset caused by an opcode fetch from an illegal address
- 0 = POR or read of SRSR

#### **MODRST — Monitor Mode Entry Module Reset Bit**

- 1 = Last reset caused by monitor mode entry when vector locations \$FFFE and \$FFFF are \$FF after POR while  $\overline{IRQ} \neq V_{TST}$
- 0 = POR or read of SRSR

#### **LVI — Low Voltage Inhibit Reset Bit**

- 1 = Last reset caused by LVI circuit
- 0 = POR or read of SRSR


## 14.9.1 TIM Status and Control Register

The TIM status and control register (TSC) does the following:

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

Address: \$0020

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

 = Unimplemented

**Figure 14-4. TIM Status and Control Register (TSC)**

### TOF — TIM Overflow Flag Bit

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

1 = TIM counter has reached modulo value

0 = TIM counter has not reached modulo value

### TOIE — TIM Overflow Interrupt Enable Bit

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

1 = TIM overflow interrupts enabled

0 = TIM overflow interrupts disabled

### TSTOP — TIM Stop Bit

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

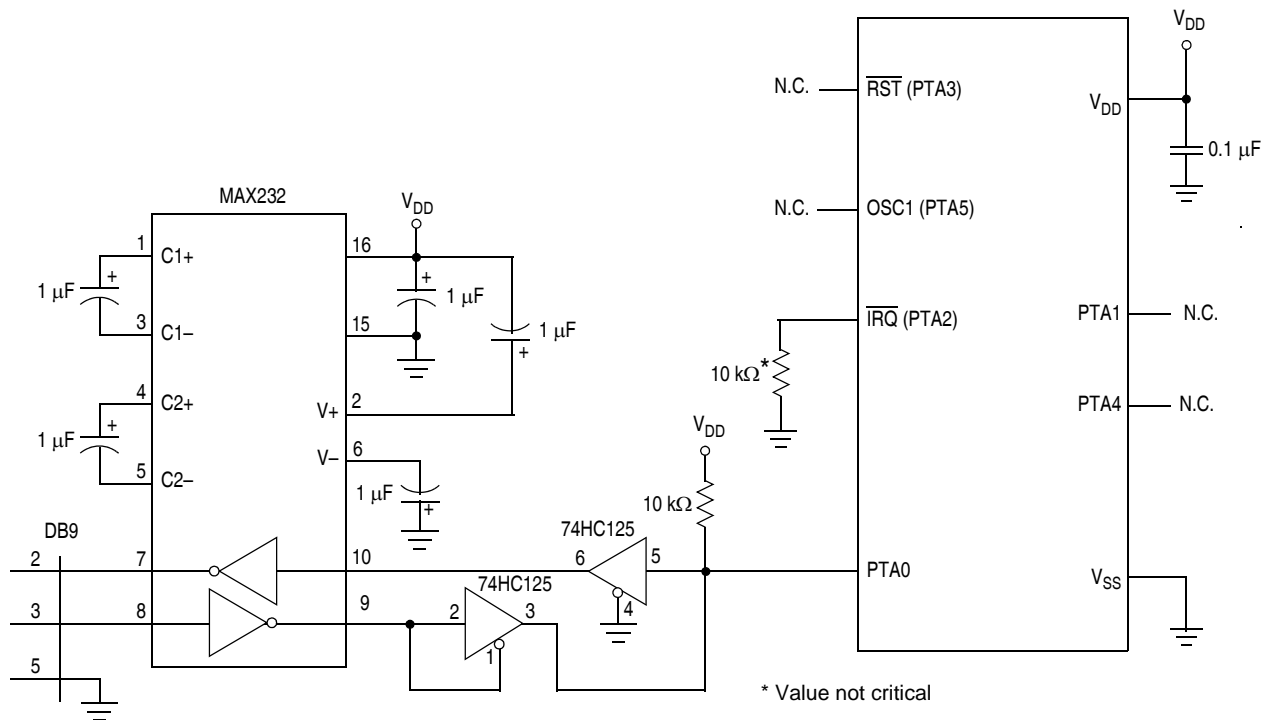
1 = TIM counter stopped

0 = TIM counter active

#### **NOTE**

*Do not set the TSTOP bit before entering wait mode if the TIM is required to exit wait mode. When the TSTOP bit is set and the timer is configured for input capture operation, input captures are inhibited until the TSTOP bit is cleared.*

*When using TSTOP to stop the timer counter, see if any timer flags are set. If a timer flag is set, it must be cleared by clearing TSTOP, then clearing the flag, then setting TSTOP again.*



**Figure 15-12. Monitor Mode Circuit (Internal Clock, No High Voltage)**

Simple monitor commands can access any memory address. In monitor mode, the MCU can execute code downloaded into RAM by a host computer while most MCU pins retain normal operating mode functions. All communication between the host computer and the MCU is through the PTA0 pin. A level-shifting and multiplexing interface is required between PTA0 and the host computer. PTA0 is used in a wired-OR configuration and requires a pullup resistor.

The monitor code has been updated from previous versions of the monitor code to allow enabling the internal oscillator to generate the internal clock. This addition, which is enabled when  $\overline{\text{IRQ}}$  is held low out of reset, is intended to support serial communication/programming at 9600 baud in monitor mode by using the internal oscillator, and the internal oscillator user trim value  $\text{OSCTRIM}$  (FLASH location \$FFC0, if programmed) to generate the desired internal frequency (3.2 MHz). Since this feature is enabled only when  $\overline{\text{IRQ}}$  is held low out of reset, it cannot be used when the reset vector is programmed (i.e., the value is not \$FFFF) because entry into monitor mode in this case requires  $V_{\text{TST}}$  on  $\overline{\text{IRQ}}$ . The  $\overline{\text{IRQ}}$  pin must remain low during this monitor session in order to maintain communication.

Table 15-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a power-on reset (POR) and will allow communication at 9600 baud provided one of the following sets of conditions is met:

- If \$FFFE and \$FFFF do not contain \$FF (programmed state):
  - The external clock is 9.8304 MHz
  - $\overline{\text{IRQ}} = V_{\text{TST}}$
- If \$FFFE and \$FFFF contain \$FF (erased state):
  - The external clock is 9.8304 MHz
  - $\overline{\text{IRQ}} = V_{\text{DD}}$  (this can be implemented through the internal  $\overline{\text{IRQ}}$  pullup)
- If \$FFFE and \$FFFF contain \$FF (erased state):
  - $\overline{\text{IRQ}} = V_{\text{SS}}$  (internal oscillator is selected, no external clock required)

If monitor mode was entered with  $V_{TST}$  on  $\overline{IRQ}$ , then the COP is disabled as long as  $V_{TST}$  is applied to  $\overline{IRQ}$ .

### 15.3.1.2 Forced Monitor Mode

If entering monitor mode without high voltage on  $\overline{IRQ}$ , then startup port pin requirements and conditions, (PTA1/PTA4) are not in effect. This is to reduce circuit requirements when performing in-circuit programming.

#### NOTE

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

If monitor mode was entered as a result of the reset vector being blank, the COP is always disabled regardless of the state of  $\overline{IRQ}$ .

If the voltage applied to the  $\overline{IRQ}$  is less than  $V_{TST}$ , the MCU will come out of reset in user mode. Internal circuitry monitors the reset vector fetches and will assert an internal reset if it detects that the reset vectors are erased (\$FF). When the MCU comes out of reset, it is forced into monitor mode without requiring high voltage on the  $\overline{IRQ}$  pin. Once out of reset, the monitor code is initially executing with the internal clock at its default frequency.

If  $\overline{IRQ}$  is held high, all pins will default to regular input port functions except for PTA0 and PTA5 which will operate as a serial communication port and OSC1 input respectively (refer to [Figure 15-10](#)). That will allow the clock to be driven from an external source through OSC1 pin.

If  $\overline{IRQ}$  is held low, all pins will default to regular input port function except for PTA0 which will operate as serial communication port. Refer to [Figure 15-11](#).

Regardless of the state of the  $\overline{IRQ}$  pin, it will not function as a port input pin in monitor mode. Bit 2 of the Port A data register will always read 0. The BIH and BIL instructions will behave as if the  $\overline{IRQ}$  pin is enabled, regardless of the settings in the configuration register. See [Chapter 5 Configuration Register \(CONFIG\)](#).

The COP module is disabled in forced monitor mode. Any reset other than a power-on reset (POR) will automatically force the MCU to come back to the forced monitor mode.

### 15.3.1.3 Monitor Vectors

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

#### NOTE

*Exiting monitor mode after it has been initiated by having a blank reset vector requires a power-on reset (POR). Pulling  $\overline{RST}$  (when  $\overline{RST}$  pin available) low will not exit monitor mode in this situation.*

[Table 15-2](#) summarizes the differences between user mode and monitor mode regarding vectors.

### 15.3.2 Security

A security feature discourages unauthorized reading of FLASH locations while in monitor mode. The host can bypass the security feature at monitor mode entry by sending eight security bytes that match the bytes at locations \$FFF6–\$FFFD. Locations \$FFF6–\$FFFD contain user-defined data.

**NOTE**

*Do not leave locations \$FFF6–\$FFFD blank. For security reasons, program locations \$FFF6–\$FFFD even if they are not used for vectors.*

During monitor mode entry, the MCU waits after the power-on reset for the host to send the eight security bytes on pin PTA0. If the received bytes match those at locations \$FFF6–\$FFFD, the host bypasses the security feature and can read all FLASH locations and execute code from FLASH. Security remains bypassed until a power-on reset occurs. If the reset was not a power-on reset, security remains bypassed and security code entry is not required. See [Figure 15-18](#).

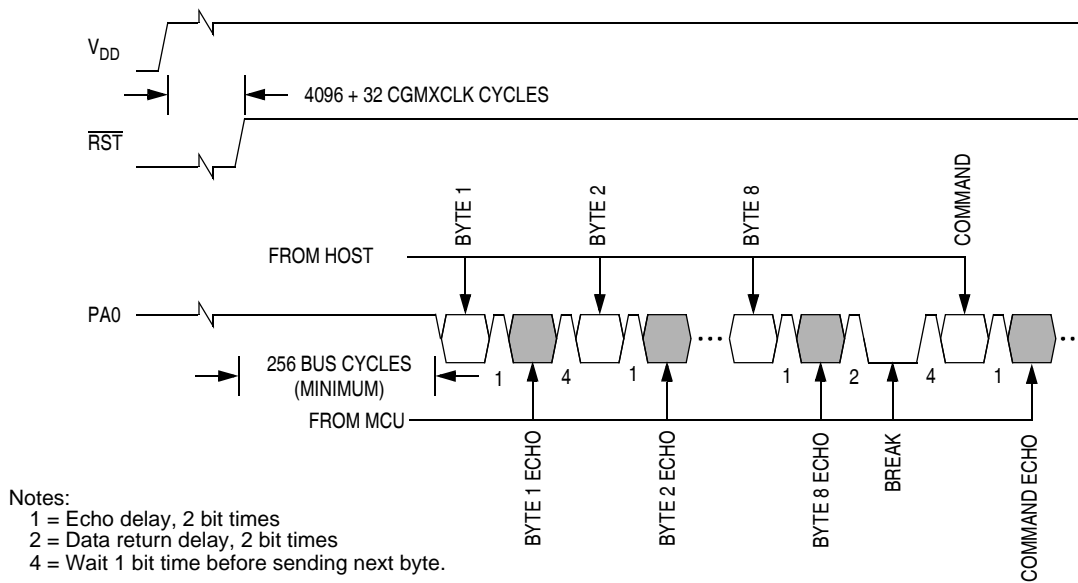
Upon power-on reset, if the received bytes of the security code do not match the data at locations \$FFF6–\$FFFD, the host fails to bypass the security feature. The MCU remains in monitor mode, but reading a FLASH location returns an invalid value and trying to execute code from FLASH causes an illegal address reset. After receiving the eight security bytes from the host, the MCU transmits a break character, signifying that it is ready to receive a command.

**NOTE**

*The MCU does not transmit a break character until after the host sends the eight security bytes.*

To determine whether the security code entered is correct, check to see if bit 6 of RAM address \$80 is set. If it is, then the correct security code has been entered and FLASH can be accessed.

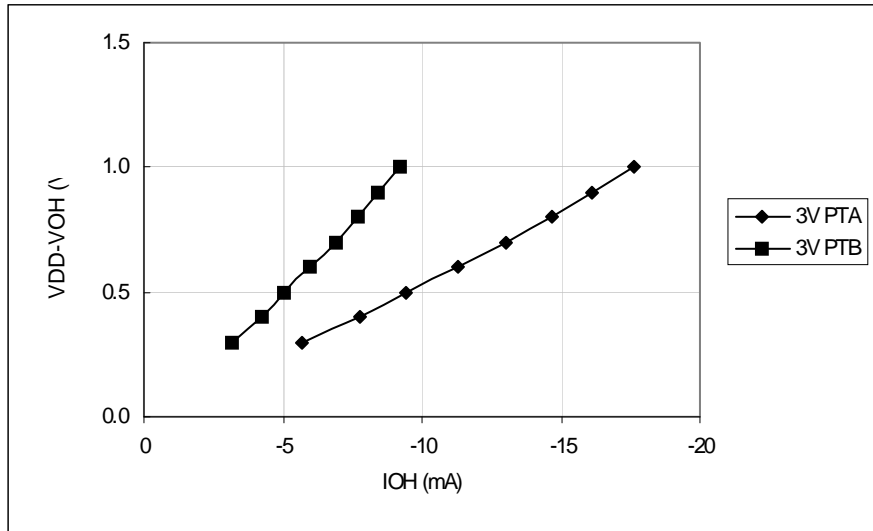
If the security sequence fails, the device should be reset by a power-on reset and brought up in monitor mode to attempt another entry. After failing the security sequence, the FLASH module can also be mass erased by executing an erase routine that was downloaded into internal RAM. The mass erase operation clears the security code locations so that all eight security bytes become \$FF (blank).



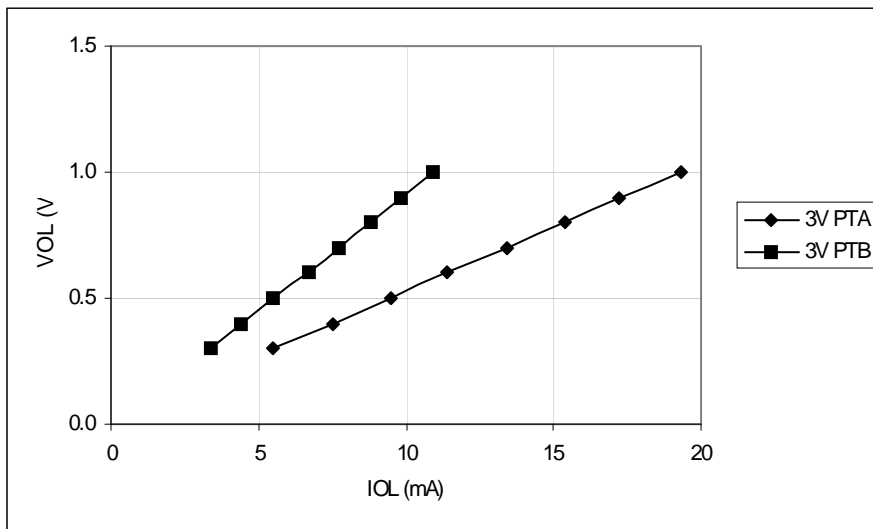
**Figure 15-18. Monitor Mode Entry Timing**



## 16.10 Typical 3.0-V Output Drive Characteristics



**Figure 16-5. Typical 3-Volt Output High Voltage versus Output High Current (25°C)**



**Figure 16-6. Typical 3-Volt Output Low Voltage versus Output Low Current (25°C)**

## 16.13 Supply Current Characteristics

Characteristic <sup>(1)</sup>	Voltage	Bus Frequency (MHz)	Symbol	Typ <sup>(2)</sup>	Max	Unit
Run Mode $V_{DD}$ supply current <sup>(3)</sup>	5.0 3.0	3.2 3.2	$R_{IDD}$	6.0 2.5	7.0 3.2	mA
Wait Mode $V_{DD}$ supply current <sup>(4)</sup>	5.0 3.0	3.2 3.2	$W_{IDD}$	1.0 0.67	1.5 1.0	mA mA
Stop Mode $V_{DD}$ supply current <sup>(5)</sup> -40 to 85°C -40 to 105°C -40 to 125°C 25°C with auto wakeup enabled Incremental current with LVI enabled at 25°C	5.0		$S_{IDD}$	0.04 — — 7 125	1.0 2.0 5.0 — —	μA
Stop Mode $V_{DD}$ supply current <sup>(5)</sup> -40 to 85°C -40 to 105°C -40 to 125°C 25°C with auto wakeup enabled Incremental current with LVI enabled at 25°C	3.0		$S_{IDD}$	0.02 — — 5 100	0.5 1.0 4.0 — —	μA

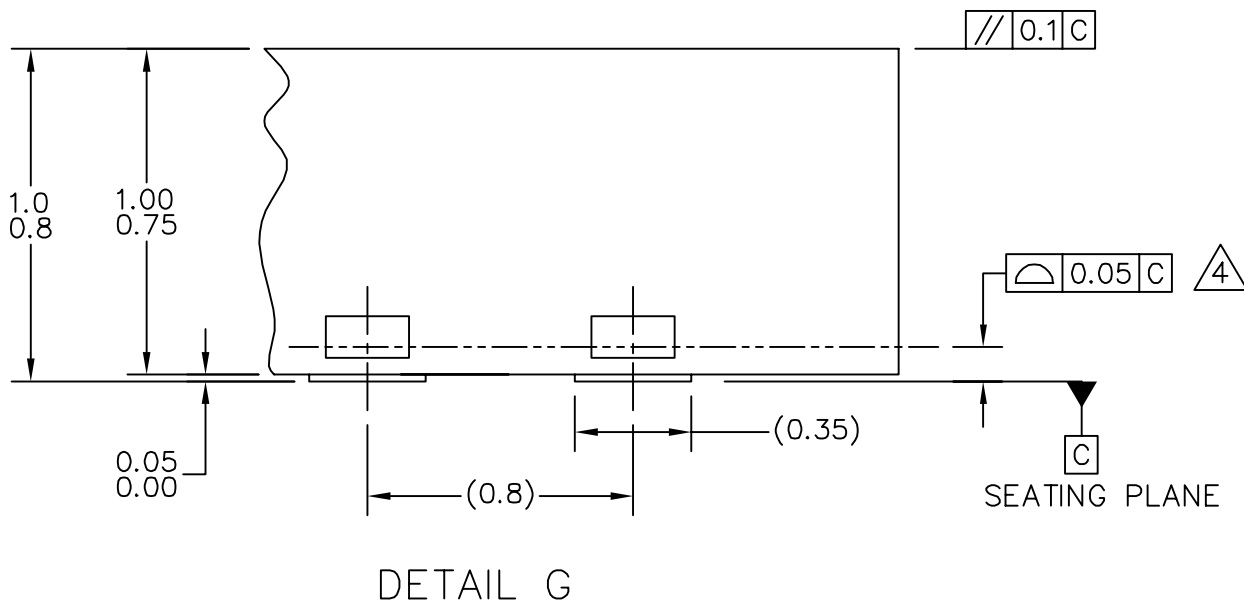
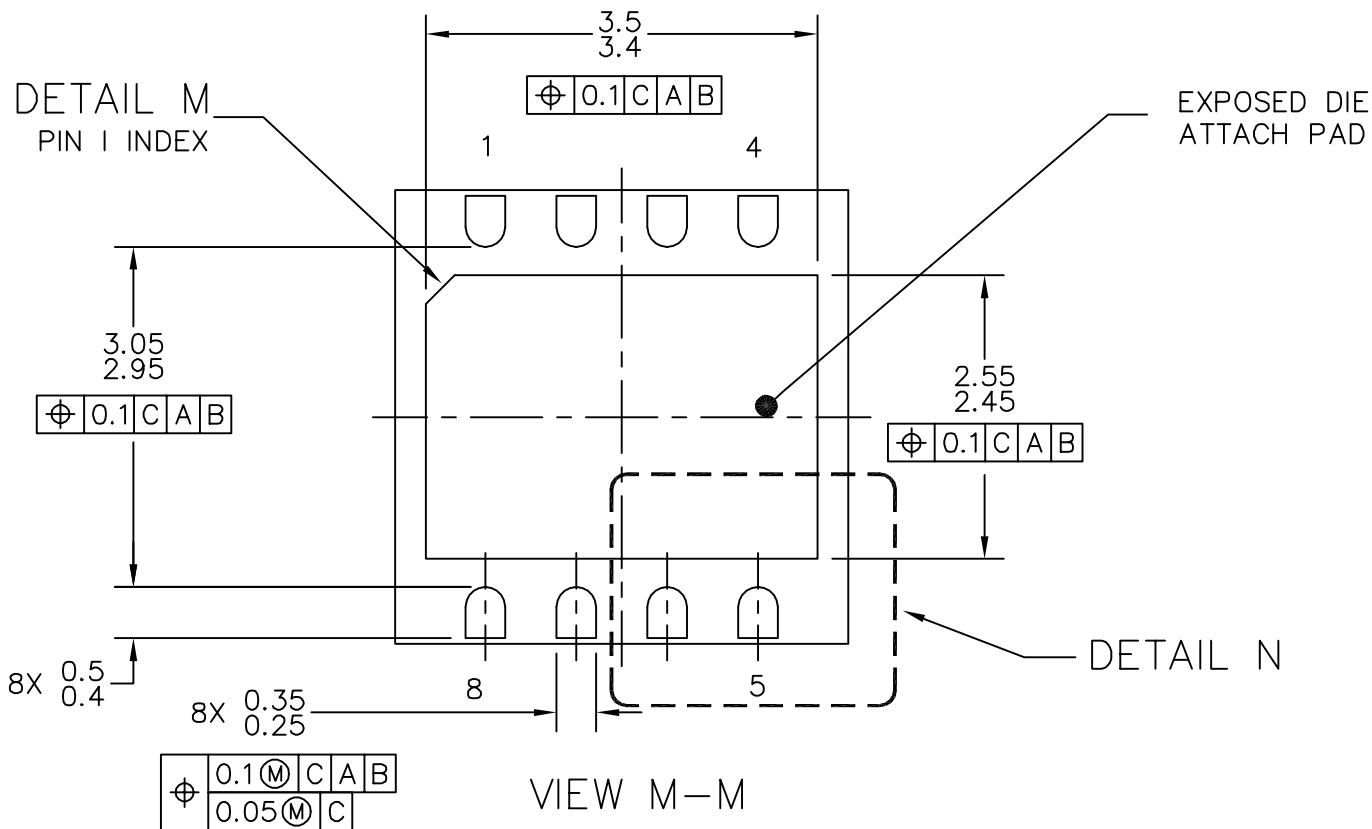
- $V_{SS} = 0$  Vdc,  $T_A = T_L$  to  $T_H$ , unless otherwise noted.
- Typical values reflect average measurements at 25°C only.
- Run (operating)  $I_{DD}$  measured using trimmed internal oscillator, ADC off, all other modules enabled. All pins configured as inputs and tied to 0.2 V from rail.
- Wait  $I_{DD}$  measured using trimmed internal oscillator, ADC off, all other modules enabled. All pins configured as inputs and tied to 0.2 V from rail.
- Stop  $I_{DD}$  measured with all pins tied to 0.2 V or less from rail. No dc loads. On the 8-pin versions, port B is configured as inputs with pullups enabled.

## 16.16 Memory Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
RAM data retention voltage	$V_{RDR}$	1.3	—	—	V
FLASH program bus clock frequency	—	1	—	—	MHz
FLASH read bus clock frequency	$f_{Read}^{(1)}$	0	—	8 M	Hz
FLASH page erase time <1 k cycles >1 k cycles	$t_{Erase}$	0.9 3.6	1 4	1.1 5.5	ms
FLASH mass erase time	$t_{MErase}$	4	—	—	ms
FLASH PGM/ERASE to HVEN setup time	$t_{NVS}$	10	—	—	$\mu$ s
FLASH high-voltage hold time	$t_{NVH}$	5	—	—	$\mu$ s
FLASH high-voltage hold time (mass erase)	$t_{NVHL}$	100	—	—	$\mu$ s
FLASH program hold time	$t_{PGS}$	5	—	—	$\mu$ s
FLASH program time	$t_{PROG}$	30	—	40	$\mu$ s
FLASH return to read time	$t_{RCV}^{(2)}$	1	—	—	$\mu$ s
FLASH cumulative program HV period	$t_{HV}^{(3)}$	—	—	4	ms
FLASH endurance <sup>(4)</sup>	—	10 k	100 k	—	Cycles
FLASH data retention time <sup>(5)</sup>	—	15	100	—	Years

- $f_{Read}$  is defined as the frequency range for which the FLASH memory can be read.
- $t_{RCV}$  is defined as the time it needs before the FLASH can be read after turning off the high voltage charge pump, by clearing HVEN to 0.
- $t_{HV}$  is defined as the cumulative high voltage programming time to the same row before next erase.  
 $t_{HV}$  must satisfy this condition:  $t_{NVS} + t_{NVH} + t_{PGS} + (t_{PROG} \times 32) \leq t_{HV}$  maximum.
- Typical endurance was evaluated for this product family. For additional information on how Freescale defines *Typical Endurance*, please refer to Engineering Bulletin EB619.
- Typical data retention values are based on intrinsic capability of the technology measured at high temperature and de-rated to 25°C using the Arrhenius equation. For additional information on how Freescale defines *Typical Data Retention*, please refer to Engineering Bulletin EB618.





TITLE: THERMALLY ENHANCED DUAL  
 FLAT NO LEAD PACKAGE (DFN)  
 8 TERMINAL, 0.8 PITCH (4 X 4 X 1)

CASE NUMBER: 1452-01

STANDARD: NON-JEDEC

PACKAGE CODE: 6165

SHEET: 2 OF 5