Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	-
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 6x10b
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/mc908qy2acdwe

Chapter 11
Oscillator (OSC) Module

Chapter 12
Input/Output Ports (PORTS)

Chapter 1
General Description

1.1 Introduction

The MC68HC908QY4A is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

Table 1-1. Summary of Device Variations

Device	FLASH Memory Size	ADC	Pin Count
MC68HC908QT1A	1536 bytes	—	8 pins
MC68HC908QT2A	1536 bytes	6 channel, 10 bit	8 pins
MC68HC908QT4A	4096 bytes	6 channel, 10 bit	8 pins
MC68HC908QY1A	1536 bytes	—	16 pins
MC68HC908QY2A	1536 bytes	6 channel, 10 bit	16 pins
MC68HC908QY4A	4096 bytes	6 channel, 10 bit	16 pins

1.2 Features

Features include:

- High-performance M68HC08 CPU core
- Fully upward-compatible object code with M68HC05 Family
- 5-V and 3-V operating voltages (V_{DD})
- 8-MHz internal bus operation at 5 V, 4-MHz at 3 V
- Trimmable internal oscillator
  - Software selectable 1 MHz, 2 MHz, or 3.2 MHz internal bus operation
  - 8-bit trim capability
  - ±25% untrimmed
  - Trimmable to approximately 0.4%(1)
- Software selectable crystal oscillator range, 32–100 kHz, 1–8 MHz and 8–32 MHz
- Software configurable input clock from either internal or external source
- Auto wakeup from STOP capability using dedicated internal 32-kHz RC or bus clock source
- On-chip in-application programmable FLASH memory
  - Internal program/erase voltage generation
  - Monitor ROM containing user callable program/erase routines
  - FLASH security(2)

1. See 16.11 Oscillator Characteristics for internal oscillator specifications
2. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.

Chapter 2
Memory

2.1 Introduction

The central processor unit (CPU08) can address 64 Kbytes of memory space. The memory map, shown in Figure 2-1.

2.2 Unimplemented Memory Locations

Executing code from an unimplemented location will cause an illegal address reset. In Figure 2-1, unimplemented locations are shaded.

2.3 Reserved Memory Locations

Accessing a reserved location can have unpredictable effects on MCU operation. In Figure 2-1, register locations are marked with the word Reserved or with the letter R.

2.4 Direct Page Registers

Figure 2-2 shows the memory mapped registers of the MC68HC908QYA/QTA Family. Registers with addresses between $0000 and $00FF are considered direct page registers and all instructions including those with direct page addressing modes can access them. Registers between $0100 and $FFFF require non-direct page addressing modes. See Chapter 7 Central Processor Unit (CPU) for more information on addressing modes.

Table 2-1. Vector Addresses

Vector Priority	Vector	Address	Vector
Lowest	IF22-IF16	$FFD0,1-$FFDC,D	Not used
	IF15	$FFDE,F	ADC conversion complete vector
	IF14	$FFE0,1	Keyboard vector
	IF13	—	Not used
	IF12	—	Not used
	IF11	—	Not used
	IF10	—	Not used
	IF9	—	Not used
	IF8	—	Not used
	IF7	—	Not used
	IF6	—	Not used
	IF5	$FFF2,3	TIM overflow vector
	IF4	$FFF4,5	TIM channel 1 vector
	IF3	$FFF6,7	TIM channel 0 vector
	IF2	—	Not used
	IF1	$FFFA,B	$\overline{\text{IRQ}}$ vector
	—	$FFFC,D	SWI vector
Highest	—	$FFFE,F	Reset vector

2.5 Random-Access Memory (RAM)

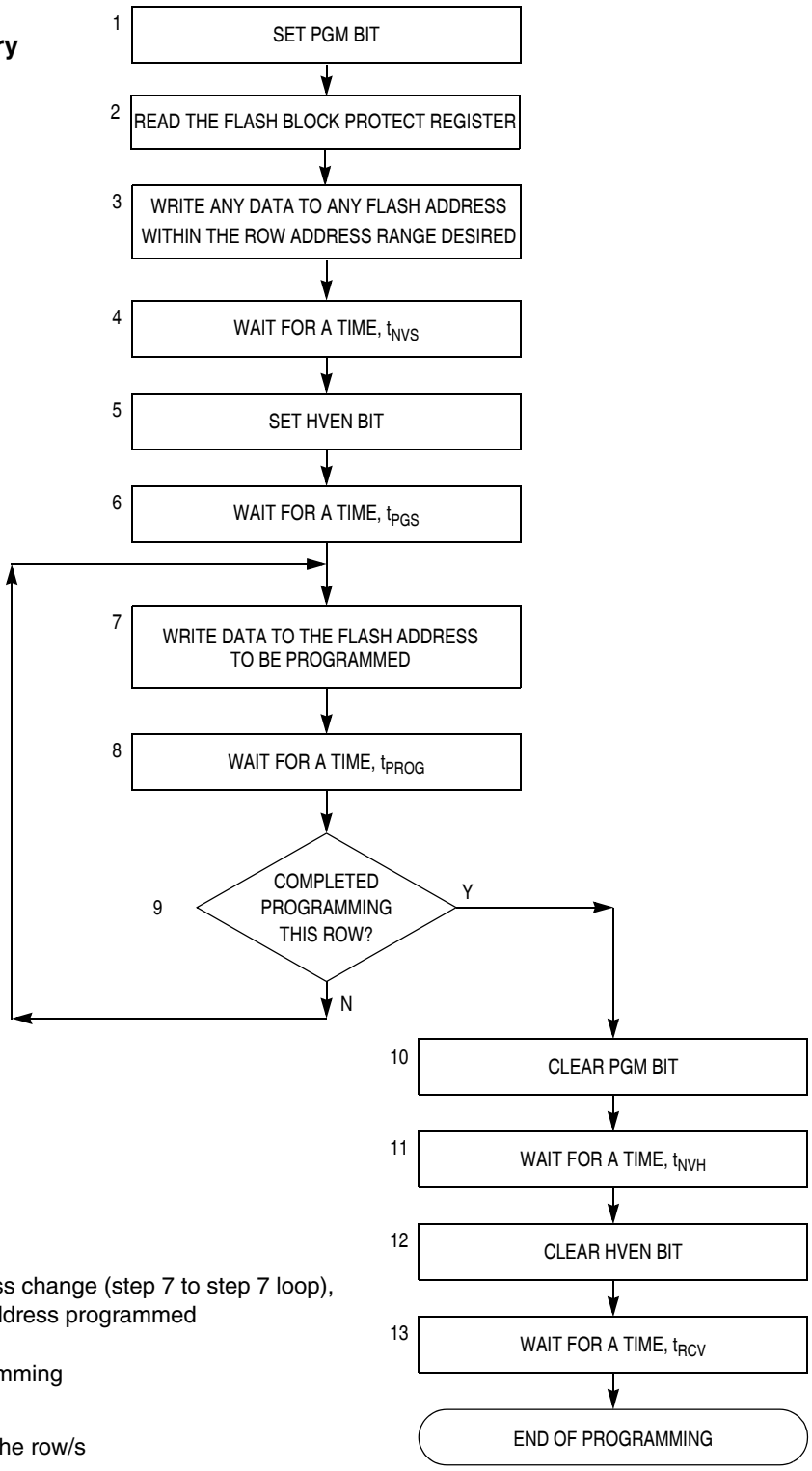
This MCU includes static RAM. The locations in RAM below $0100 can be accessed using the more efficient direct addressing mode, and any single bit in this area can be accessed with the bit manipulation instructions (BCLR, BSET, BRCLR, and BRSET). Locating the most frequently accessed program variables in this area of RAM is preferred.

The RAM retains data when the MCU is in low-power wait or stop mode. At power-on, the contents of RAM are uninitialized. RAM data is unaffected by any reset provided that the supply voltage does not drop below the minimum value for RAM retention.

For compatibility with older M68HC05 MCUs, the HC08 resets the stack pointer to $00FF. In the devices that have RAM above $00FF, it is usually best to reinitialize the stack pointer to the top of the RAM so the direct page RAM can be used for frequently accessed RAM variables and bit-addressable program variables. Include the following 2-instruction sequence in your reset initialization routine (where RamLast is equated to the highest address of the RAM).

```
LDHX    #RamLast+1    ;point one past RAM
TXS                   ;SP<-(H:X-1)
```

Algorithm for Programming a Row (32 Bytes) of FLASH Memory



NOTES:

The time between each FLASH address change (step 7 to step 7 loop), or the time between the last FLASH address programmed to clearing PGM bit (step 7 to step 10) must not exceed the maximum programming time, t_{PROG} max.

This row program algorithm assumes the row/s to be programmed are initially erased.

Figure 2-4. FLASH Programming Flowchart

clocks are too fast, then the clock must be divided to the appropriate frequency. This divider is specified by the ADIV[1:0] bits and can be divide-by 1, 2, 4, or 8.

3.3.2 Input Select and Pin Control

Only one analog input may be used for conversion at any given time. The channel select bits in ADSCR are used to select the input signal for conversion.

3.3.3 Conversion Control

Conversions can be performed in either 10-bit mode or 8-bit mode as determined by the MODE bits. Conversions can be initiated by either a software or hardware trigger. In addition, the ADC10 module can be configured for low power operation, long sample time, and continuous conversion.

3.3.3.1 Initiating Conversions

A conversion is initiated:

- Following a write to ADSCR (with ADCH bits not all 1s) if software triggered operation is selected.
- Following a hardware trigger event if hardware triggered operation is selected.
- Following the transfer of the result to the data registers when continuous conversion is enabled.

If continuous conversions are enabled a new conversion is automatically initiated after the completion of the current conversion. In software triggered operation, continuous conversions begin after ADSCR is written and continue until aborted. In hardware triggered operation, continuous conversions begin after a hardware trigger event and continue until aborted.

3.3.3.2 Completing Conversions

A conversion is completed when the result of the conversion is transferred into the data result registers, ADRH and ADRL. This is indicated by the setting of the COCO bit. An interrupt is generated if AIEN is high at the time that COCO is set.

A blocking mechanism prevents a new result from overwriting previous data in ADRH and ADRL if the previous data is in the process of being read while in 10-bit mode (ADRH has been read but ADRL has not). In this case the data transfer is blocked, COCO is not set, and the new result is lost. When a data transfer is blocked, another conversion is initiated regardless of the state of ADCO (single or continuous conversions enabled). If single conversions are enabled, this could result in several discarded conversions and excess power consumption. To avoid this issue, the data registers must not be read after initiating a single conversion until the conversion completes.

3.3.3.3 Aborting Conversions

Any conversion in progress will be aborted when:

- A write to ADSCR occurs (the current conversion will be aborted and a new conversion will be initiated, if ADCH are not all 1s).
- A write to ADCLK occurs.
- The MCU is reset.
- The MCU enters stop mode with ACLK not enabled.

When a conversion is aborted, the contents of the data registers, ADRH and ADRL, are not altered but continue to be the values transferred after the completion of the last successful conversion. In the case that the conversion was aborted by a reset, ADRH and ADRL return to their reset states.

LVIPWRD — LVI Power Disable Bit

LVIPWRD disables the LVI module.

1 = LVI module power disabled
0 = LVI module power enabled

LVITRIP — LVI Trip Point Selection Bit

LVITRIP selects the voltage operating mode of the LVI module. The voltage mode selected for the LVI should match the operating V_{DD} for the LVI's voltage trip points for each of the modes.

1 = LVI operates for a 5-V protection
0 = LVI operates for a 3-V protection

NOTE

The LVITRIP bit is cleared by a power-on reset (POR) only. Other resets will leave this bit unaffected.

SSREC — Short Stop Recovery Bit

SSREC enables the CPU to exit stop mode with a delay of 32 BUSCLKX4 cycles instead of a 4096 BUSCLKX4 cycle delay.

1 = Stop mode recovery after 32 BUSCLKX4 cycles
0 = Stop mode recovery after 4096 BUSCLKX4 cycles

NOTE

Exiting stop mode by an LVI reset will result in the long stop recovery.

When using the LVI during normal operation but disabling during stop mode, the LVI will have an enable time of t_{EN}. The system stabilization time for power-on reset and long stop recovery (both 4096 BUSCLKX4 cycles) gives a delay longer than the LVI enable time for these startup scenarios. There is no period where the MCU is not protected from a low-power condition. However, when using the short stop recovery configuration option, the 32 BUSCLKX4 delay must be greater than the LVI's turn on time to avoid a period in startup where the LVI is not protecting the MCU.

STOP — STOP Instruction Enable Bit

STOP enables the STOP instruction.

1 = STOP instruction enabled
0 = STOP instruction treated as illegal opcode

COPD — COP Disable Bit

COPD disables the COP module.

1 = COP module disabled
0 = COP module enabled

Chapter 7
Central Processor Unit (CPU)

7.1 Introduction

The M68HC08 CPU (central processor unit) is an enhanced and fully object-code-compatible version of the M68HC05 CPU. The *CPU08 Reference Manual* (document order number CPU08RM/AD) contains a description of the CPU instruction set, addressing modes, and architecture.

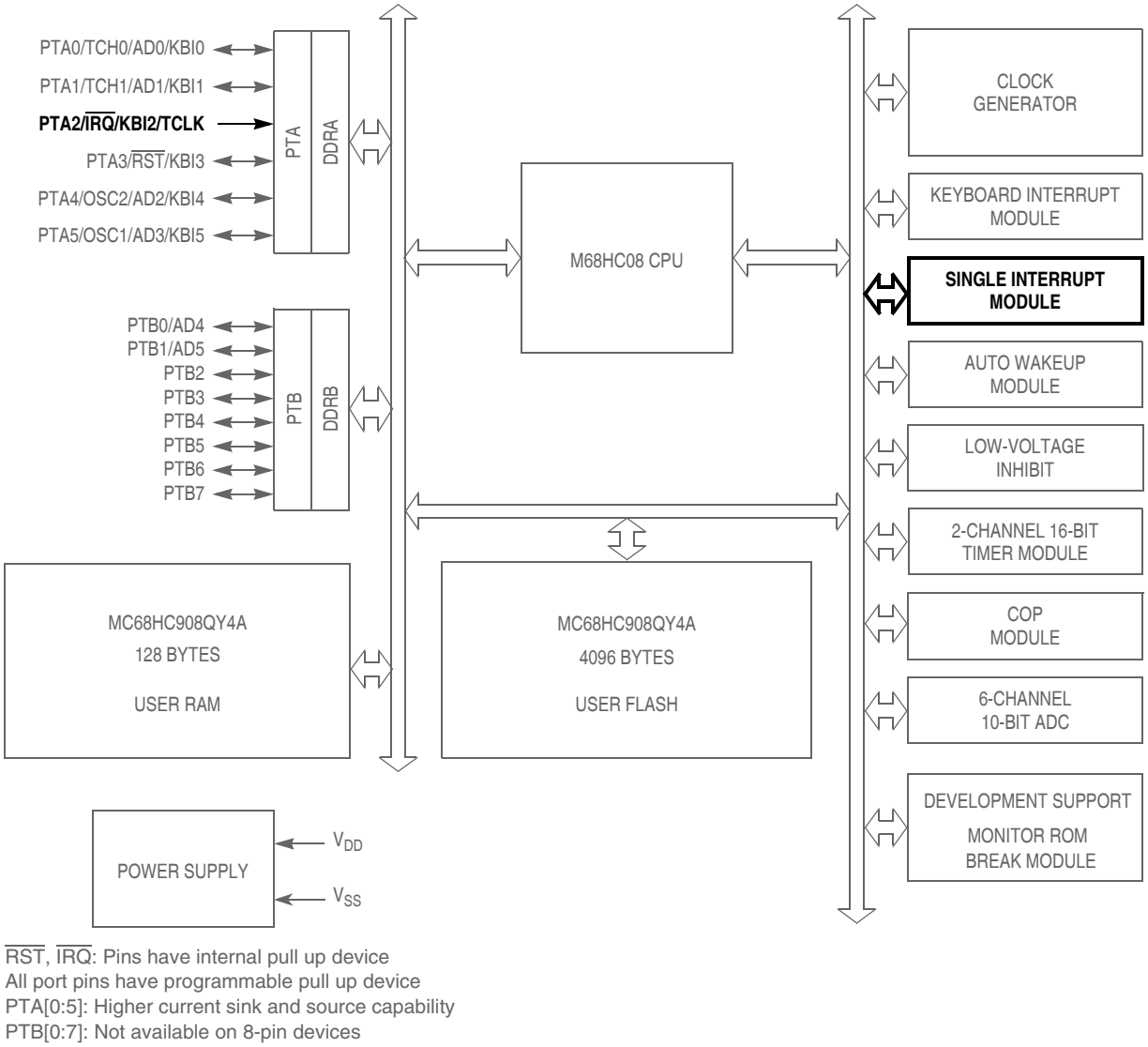
7.2 Features

Features of the CPU include:

- Object code fully upward-compatible with M68HC05 Family
- 16-bit stack pointer with stack manipulation instructions
- 16-bit index register with x-register manipulation instructions
- 8-MHz CPU internal bus frequency
- 64-Kbyte program/data memory space
- 16 addressing modes
- Memory-to-memory data moves without using accumulator
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- Enhanced binary-coded decimal (BCD) data handling
- Modular architecture with expandable internal bus definition for extension of addressing range beyond 64 Kbytes
- Low-power stop and wait modes

7.3 CPU Registers

Figure 7-1 shows the five CPU registers. CPU registers are not part of the memory map.



RST, IRQ: Pins have internal pull up device
All port pins have programmable pull up device
PTA[0:5]: Higher current sink and source capability
PTB[0:7]: Not available on 8-pin devices

Figure 8-1. Block Diagram Highlighting IRQ Block and Pin

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.

8.4 Interrupts

The following IRQ source can generate interrupt requests:

- Interrupt flag (IRQF) — The IRQF bit is set when the $\overline{\text{IRQ}}$ 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 BFCR in the SIM section of this data sheet.

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

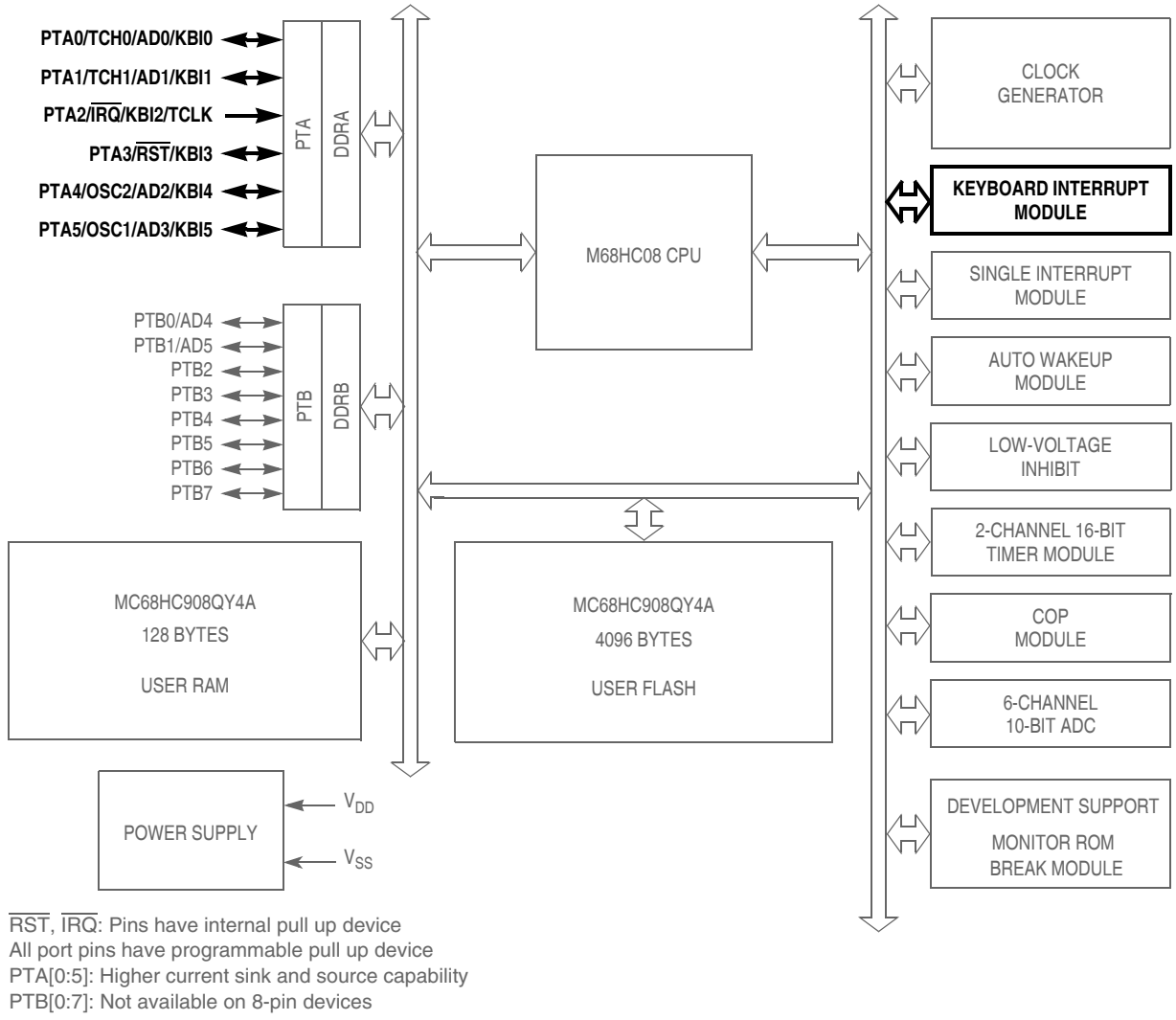
To protect status bits during the break state, write a 0 to BCFE. With BCFE cleared (its default state), software can read and write registers during the break state without affecting status bits. Some status bits have a two-step read/write clearing procedure. If software does the first step on such 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 does not share its pin with any module on this MCU.

8.7.1 IRQ Input Pins ($\overline{\text{IRQ}}$)

The $\overline{\text{IRQ}}$ pin provides a maskable external interrupt source. The $\overline{\text{IRQ}}$ pin contains an internal pullup device.



RST, IRQ: Pins have internal pull up device
All port pins have programmable pull up device
PTA[0:5]: Higher current sink and source capability
PTB[0:7]: Not available on 8-pin devices

Figure 9-1. Block Diagram Highlighting KBI Block and Pins

9.3.1.1 MODEK = 1

If the MODEK bit is set, the keyboard interrupt inputs are both edge and level sensitive. The KBIPx bit will determine whether a edge sensitive pin detects rising or falling edges and on level sensitive pins whether the pin detects low or high levels. With MODEK set, both of the following actions must occur to clear a keyboard interrupt request:

- Return of all enabled keyboard interrupt inputs to a deasserted level. As long as any enabled keyboard interrupt pin is asserted, the keyboard interrupt remains active.
- Vector fetch or software clear. A KBI vector fetch generates an interrupt acknowledge signal to clear the KBI latch. Software generates the interrupt acknowledge signal by writing a 1 to ACKK in KBSCR. The ACKK bit is useful in applications that poll the keyboard interrupt inputs and require software to clear the KBI latch. Writing to ACKK prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACKK does not affect subsequent transitions on the keyboard interrupt inputs. An edge detect that occurs after writing to ACKK latches another interrupt request. If the keyboard interrupt mask bit, IMASKK, is clear, the CPU loads the program counter with the KBI vector address.

9.7 I/O Signals

The KBI module can share its pins with the general-purpose I/O pins. See Figure 9-1 for the port pins that are shared.

9.7.1 KBI Input Pins (KBIx:KBI0)

Each KBI pin is independently programmable as an external interrupt source. KBI pin polarity can be controlled independently. Each KBI pin when enabled will automatically configure the appropriate pullup/pulldown device based on polarity.

9.8 Registers

The following registers control and monitor operation of the KBI module:

- KBSCR (keyboard interrupt status and control register)
- KBIER (keyboard interrupt enable register)
- KBIPR (keyboard interrupt polarity register)

9.8.1 Keyboard Status and Control Register (KBSCR)

Features of the KBSCR:

- Flags keyboard interrupt requests
- Acknowledges keyboard interrupt requests
- Masks keyboard interrupt requests
- Controls keyboard interrupt triggering sensitivity

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
Write:						ACKK		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 9-3. Keyboard Status and Control Register (KBSCR)

Bits 7–4 — Not used

KEYF — Keyboard Flag Bit

This read-only bit is set when a keyboard interrupt is pending.

1 = Keyboard interrupt pending
0 = No keyboard interrupt pending

ACKK — Keyboard Acknowledge Bit

Writing a 1 to this write-only bit clears the KBI request. ACKK always reads 0.

IMASKK— Keyboard Interrupt Mask Bit

Writing a 1 to this read/write bit prevents the output of the KBI latch from generating interrupt requests.

1 = Keyboard interrupt requests disabled
0 = Keyboard interrupt requests enabled

MODEK — Keyboard Triggering Sensitivity Bit

This read/write bit controls the triggering sensitivity of the keyboard interrupt pins.

1 = Keyboard interrupt requests on edge and level
0 = Keyboard interrupt requests on edge only

Setting MS0B causes the contents of TSC1 to be ignored by the TIM and reverts TCH1 to general-purpose I/O.

1 = Buffered output compare/PWM operation enabled
0 = Buffered output compare/PWM operation disabled

MSxA — Mode Select Bit A

When ELSxB:A ≠ 00, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation. See Table 14-2.

1 = Unbuffered output compare/PWM operation
0 = Input capture operation

When ELSxB:A = 00, this read/write bit selects the initial output level of the TCHx pin (see Table 14-2).

1 = Initial output level low
0 = Initial output level high

NOTE

Before changing a channel function by writing to the MSxB or MSxA bit, set the TSTOP and TRST bits in the TIM status and control register (TSC).

ELSxB and ELSxA — Edge/Level Select Bits

When channel x is an input capture channel, these read/write bits control the active edge-sensing logic on channel x.

When channel x is an output compare channel, ELSxB and ELSxA control the channel x output behavior when an output compare occurs.

When ELSxB and ELSxA are both clear, channel x is not connected to an I/O port, and pin TCHx is available as a general-purpose I/O pin. Table 14-2 shows how ELSxB and ELSxA work.

Table 14-2. Mode, Edge, and Level Selection

MSxB	MSxA	ELSxB	ELSxA	Mode	Configuration
X	0	0	0	Output preset	Pin under port control; initial output level high
X	1	0	0		Pin under port control; initial output level low
0	0	0	1	Input capture	Capture on rising edge only
0	0	1	0		Capture on falling edge only
0	0	1	1		Capture on rising or falling edge
0	1	0	0	Output compare or PWM	Software compare only
0	1	0	1		Toggle output on compare
0	1	1	0		Clear output on compare
0	1	1	1		Set output on compare
1	X	0	1	Buffered output compare or buffered PWM	Toggle output on compare
1	X	1	0		Clear output on compare
1	X	1	1		Set output on compare

NOTE

After initially enabling a TIM channel register for input capture operation and selecting the edge sensitivity, clear CHxF to ignore any erroneous edge detection flags.

TOVx — Toggle-On-Overflow Bit

When channel x is an output compare channel, this read/write bit controls the behavior of the channel x output when the counter overflows. When channel x is an input capture channel, TOVx has no effect.

1 = Channel x pin toggles on TIM counter overflow.

0 = Channel x pin does not toggle on TIM counter overflow.

NOTE

When TOVx is set, a counter overflow takes precedence over a channel x output compare if both occur at the same time.

CHxMAX — Channel x Maximum Duty Cycle Bit

When the TOVx bit is at 1, setting the CHxMAX bit forces the duty cycle of buffered and unbuffered PWM signals to 100%. As Figure 14-11 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.

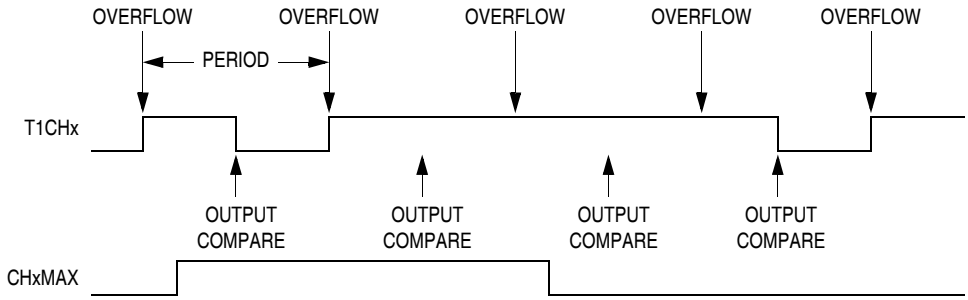


Figure 14-11. CHxMAX Latency

14.8.5 TIM Channel Registers

These read/write registers contain the captured 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.

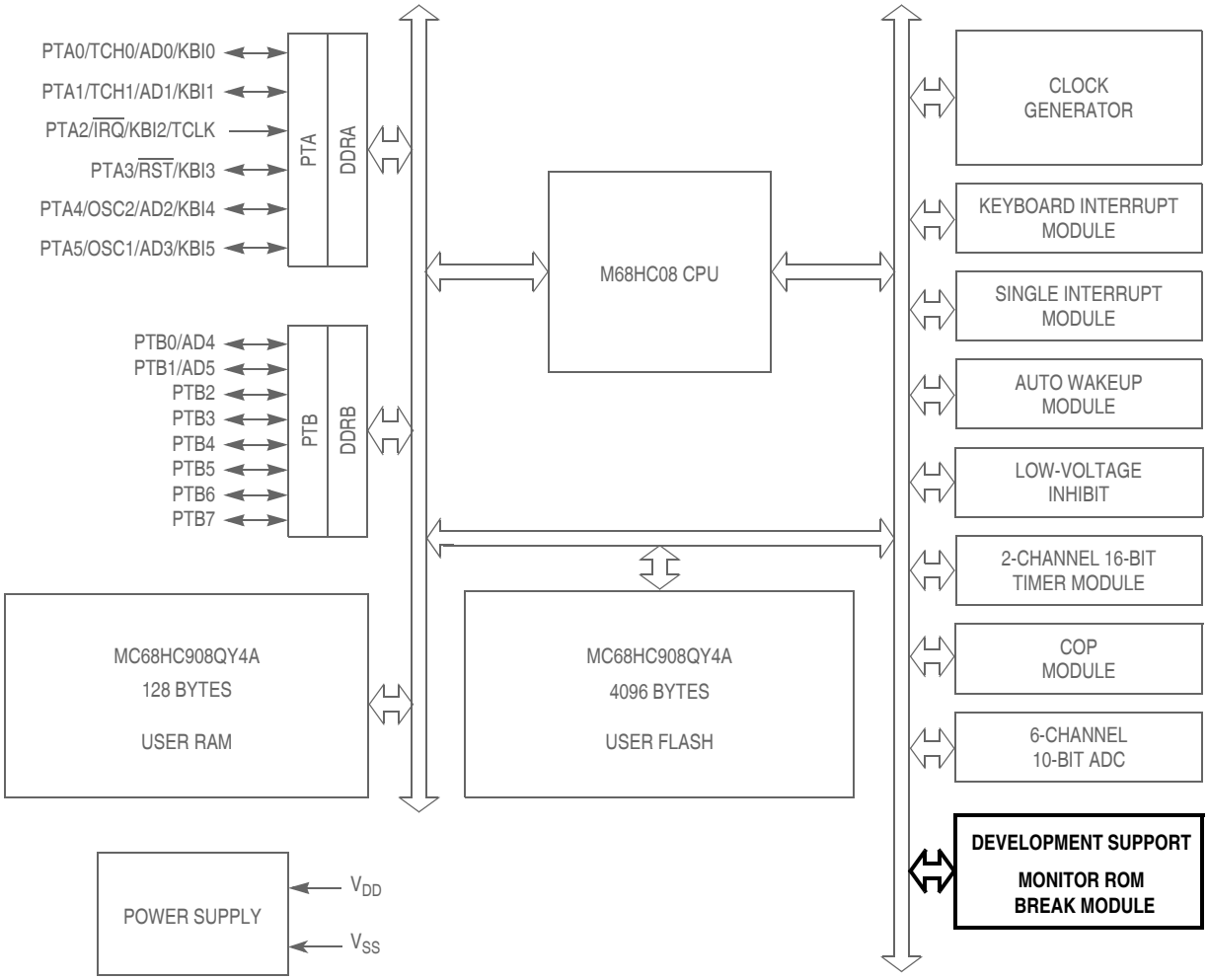
In output compare mode (MSxB:MSxA ≠ 0:0), writing to the high byte of the TIM channel x registers (TCHxH) inhibits output compares until the low byte (TCHxL) is written.

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

Figure 14-12. TIM Channel x Register High (TCHxH)

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

Figure 14-13. TIM Channel Register Low (TCHxL)



RST, IRQ: Pins have internal pull up device
All port pins have programmable pull up device
PTA[0:5]: Higher current sink and source capability
PTB[0:7]: Not available on 8-pin devices

Figure 15-1. Block Diagram Highlighting BRK and MON Blocks

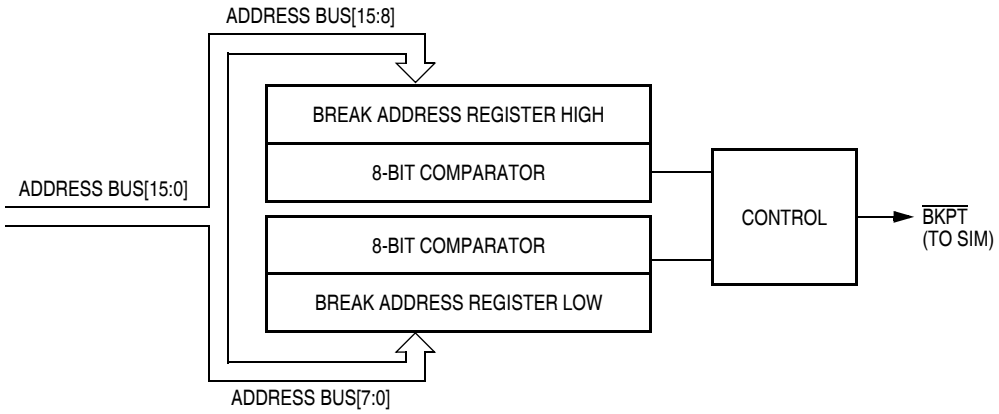


Figure 15-2. Break Module Block Diagram

Chapter 16
Electrical Specifications

16.1 Introduction

This section contains electrical and timing specifications.

16.2 Absolute Maximum Ratings

Maximum ratings are the extreme limits to which the microcontroller unit (MCU) can be exposed without permanently damaging it.

NOTE

This device is not guaranteed to operate properly at the maximum ratings. Refer to 16.5 5-V DC Electrical Characteristics and 16.8 3-V DC Electrical Characteristics for guaranteed operating conditions.

Characteristic[1]	Symbol	Value	Unit
Supply voltage	V_{DD}	−0.3 to +6.0	V
Input voltage	V_{IN}	V_{SS} −0.3 to V_{DD} +0.3	V
Mode entry voltage, $\overline{IRQ}$ pin	V_{TST}	V_{SS} −0.3 to +9.1	V
Maximum current per pin excluding PTA0–PTA5, V_{DD}, and V_{SS}	I	±15	mA
Maximum current for pins PTA0–PTA5	I_{PTA0}—I_{PTA5}	±25	mA
Storage temperature	T_{STG}	−55 to +150	°C
Maximum current out of V_{SS}	I_{MVSS}	100	mA
Maximum current into V_{DD}	I_{MVDD}	100	mA

1. Voltages references to V_{SS}.

NOTE

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{IN} and V_{OUT} be constrained to the range $V_{SS} \le (V_{IN}$ or $V_{OUT}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD}.)

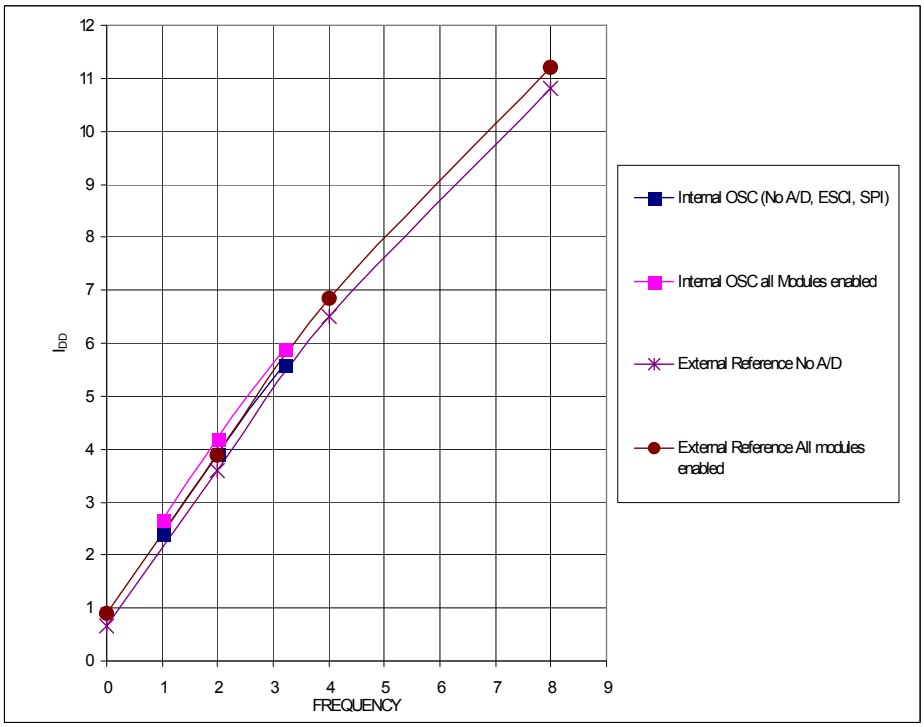


Figure 16-9. Typical 5-Volt Run Current versus Bus Frequency (25•C)

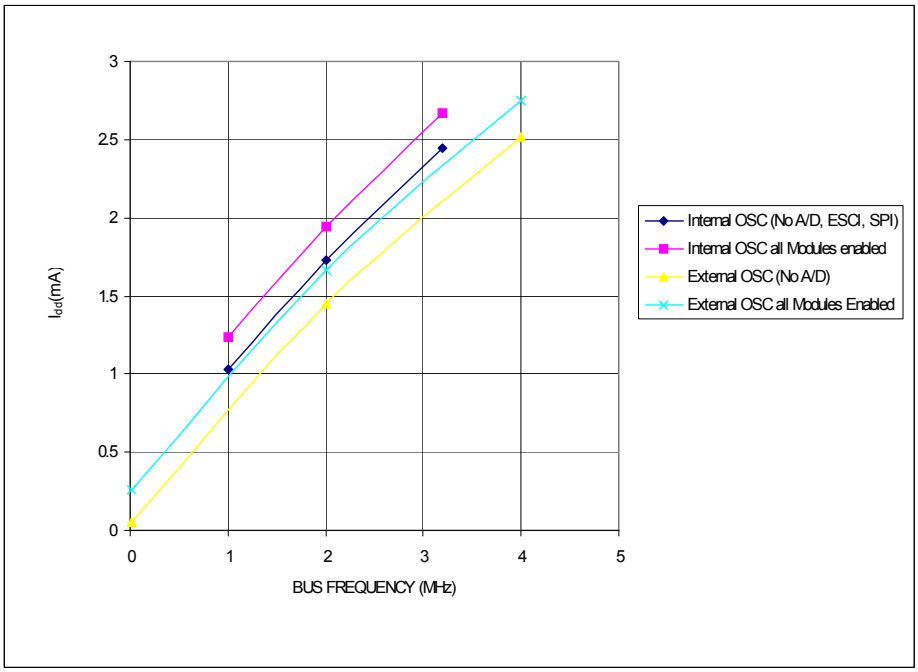


Figure 16-10. Typical 3-Volt Run Current versus Bus Frequency (25•C)

Case 968 page 2 of 3

A.2.2 Enhanced Oscillator Module (OSC)

The QYxA contains a much enhanced oscillator module that allows more options than the QYx Classic.

- The ICFS bits in the Oscillator Status and Control Register (OSCSC) allow the Internal Oscillator to be configured for 1-, 2-, or 3.2-MHz operation. Also, the ECFS bits in the same register allow a low, medium, or high crystal frequency range to be selected for the source of the system clock. With this option you can choose to use a 32-kHz (low range) or a 16-MHz (high range) crystal.
- Another improvement to the Oscillator Module design is that you can switch between internal oscillator and external oscillator options at any time. For example, if you wanted the low power advantage of running from a 32-kHz crystal but still needed some processing power to perform math calculations you could switch back and forth between internal and external clock. The same is true for switching between 1-, 2-, and 3.2-MHz internal oscillator options.

A.2.2.1 Registers Affected

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	OSCOPT1	OSCOPT0	ICFS1	ICFS0	ECFS1	ECFS0	ECGON	ECGST
Write:								
Reset:	0	0	1	0	0	0	0	0

= Unimplemented

Figure A-4. Oscillator Status and Control Register (OSCSC)

The OSCOPT bits are no longer in the CONFIG2 register and now reside in the OSCSC register. Also, the ICFSx and ECFSx bits now reside in this register.

The IFS bits are used to select different Internal Oscillator speeds.

The ECFS bits are used to select the range of crystal that should be used to provide the reference clock.