



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	4KB (4K 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/mc908qy4acdwer

Revision History

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Revision History

Date	Revision Level	Description	Page Number(s)
December, 2005	N/A	Initial release	N/A
August, 2006	1	Added 1.7 Unused Pin Termination .	20
		Figure 4-1. Auto Wakeup Interrupt Request Generation Logic — Corrected clock source.	51
		4.3 Functional Description — Clarified operation.	52
		4.5.1 Wait Mode — Corrected operation details.	53
		4.6.4 Configuration Register 2 — Corrected clock source.	55
		4.6.5 Configuration Register 1 — Added SSREC bit description.	55
		5.2 Functional Description — Corrected clock source.	58
		12.1 Introduction — Replaced note.	103
		13.7.2 Stop Mode — Corrected clock source.	121
		16.12 Supply Current Characteristics — Updated maximum values for SI_{DD} at both 5 V and 3 V.	165
		A.2.3 Improved Auto Wakeup Module (AWU) — Corrected clock source.	194
April, 2007	2	Chapter 3 Analog-to-Digital Converter (ADC10) Module — Renamed ADCSC register to ADSCR to be consistent with development tools.	37
		Figure 15-18. Monitor Mode Entry Timing — Changed CGMXCLK to BUSCLKX4	154
		16.12 Supply Current Characteristics — Added note 6 below table	165
		Chapter 17 Ordering Information and Mechanical Specifications — Updated chapter to include:	
		Table 17-1. Consumer and Industrial Device Numbering System Table 17-2. Automotive Device Numbering System 17.3 Orderable Part Numbering System 17.3.1 Consumer and Industrial Orderable Part Numbering System 17.3.2 Automotive Orderable Part Number System	171 171 172 172 172
March, 2010	3	Clarify internal oscillator trim register information.	27, 30, 31, 34, 95, 101



List of Chapters

Chapter 11

Oscillator (OSC) Module

11.1	Introduction	93
11.2	Features	93
11.3	Functional Description	93
11.3.1	Internal Signal Definitions	94
11.3.1.1	Oscillator Enable Signal (SIMOSCEN)	94
11.3.1.2	XTAL Oscillator Clock (XTALCLK)	95
11.3.1.3	RC Oscillator Clock (RCCLK)	95
11.3.1.4	Internal Oscillator Clock (INTCLK)	95
11.3.1.5	Bus Clock Times 4 (BUSCLKX4)	95
11.3.1.6	Bus Clock Times 2 (BUSCLKX2)	95
11.3.2	Internal Oscillator	95
11.3.2.1	Internal Oscillator Trimming	95
11.3.2.2	Internal to External Clock Switching	96
11.3.2.3	External to Internal Clock Switching	96
11.3.3	External Oscillator	96
11.3.4	XTAL Oscillator	96
11.3.5	RC Oscillator	98
11.4	Interrupts	98
11.5	Low-Power Modes	98
11.5.1	Wait Mode	98
11.5.2	Stop Mode	98
11.6	OSC During Break Interrupts	99
11.7	I/O Signals	99
11.7.1	Oscillator Input Pin (OSC1)	99
11.7.2	Oscillator Output Pin (OSC2)	99
11.8	Registers	100
11.8.1	Oscillator Status and Control Register	100
11.8.2	Oscillator Trim Register (OSCTRIM)	101

Chapter 12

Input/Output Ports (PORTS)

12.1	Introduction	103
12.2	Unused Pin Termination	103
12.3	Port A	103
12.3.1	Port A Data Register	104
12.3.2	Data Direction Register A	104
12.3.3	Port A Input Pullup Enable Register	105
12.3.4	Port A Summary Table	106
12.4	Port B	106
12.4.1	Port B Data Register	106
12.4.2	Data Direction Register B	107
12.4.3	Port B Input Pullup Enable Register	108
12.4.4	Port B Summary Table	108

14.3.3	Output Compare	126
14.3.3.1	Unbuffered Output Compare	127
14.3.3.2	Buffered Output Compare	128
14.3.4	Pulse Width Modulation (PWM)	128
14.3.4.1	Unbuffered PWM Signal Generation	129
14.3.4.2	Buffered PWM Signal Generation	129
14.3.4.3	PWM Initialization	130
14.4	Interrupts	131
14.5	Low-Power Modes	131
14.5.1	Wait Mode	131
14.5.2	Stop Mode	131
14.6	TIM During Break Interrupts	131
14.7	I/O Signals	132
14.7.1	TIM Channel I/O Pins (TCH1:TCH0)	132
14.7.2	TIM Clock Pin (TCLK)	132
14.8	Registers	132
14.8.1	TIM Status and Control Register	132
14.8.2	TIM Counter Registers	134
14.8.3	TIM Counter Modulo Registers	134
14.8.4	TIM Channel Status and Control Registers	135
14.8.5	TIM Channel Registers	137

Chapter 15 Development Support

15.1	Introduction	139
15.2	Break Module (BRK)	139
15.2.1	Functional Description	139
15.2.1.1	Flag Protection During Break Interrupts	141
15.2.1.2	TIM During Break Interrupts	141
15.2.1.3	COP During Break Interrupts	141
15.2.2	Break Module Registers	141
15.2.2.1	Break Status and Control Register	142
15.2.2.2	Break Address Registers	142
15.2.2.3	Break Auxiliary Register	143
15.2.2.4	Break Status Register	143
15.2.2.5	Break Flag Control Register	143
15.2.3	Low-Power Modes	144
15.3	Monitor Module (MON)	144
15.3.1	Functional Description	144
15.3.1.1	Normal Monitor Mode	148
15.3.1.2	Forced Monitor Mode	148
15.3.1.3	Monitor Vectors	149
15.3.1.4	Data Format	150
15.3.1.5	Break Signal	150
15.3.1.6	Baud Rate	150
15.3.1.7	Commands	150
15.3.2	Security	154

3.3.4.4 Code Width and Quantization Error

The ADC10 quantizes the ideal straight-line transfer function into 1024 steps (in 10-bit mode). Each step ideally has the same height (1 code) and width. The width is defined as the delta between the transition points from one code to the next. The ideal code width for an N bit converter (in this case N can be 8 or 10), defined as 1LSB, is:

$$1\text{LSB} = (V_{\text{REFH}} - V_{\text{REFL}}) / 2^N$$

Because of this quantization, there is an inherent quantization error. Because the converter performs a conversion and then rounds to 8 or 10 bits, the code will transition when the voltage is at the midpoint between the points where the straight line transfer function is exactly represented by the actual transfer function. Therefore, the quantization error will be $\pm 1/2\text{LSB}$ in 8- or 10-bit mode. As a consequence, however, the code width of the first (\$000) conversion is only 1/2LSB and the code width of the last (\$FF or \$3FF) is 1.5LSB.

3.3.4.5 Linearity Errors

The ADC10 may also exhibit non-linearity of several forms. Every effort has been made to reduce these errors but the user should be aware of them because they affect overall accuracy. These errors are:

- Zero-Scale Error (E_{ZS}) (sometimes called offset) — This error is defined as the difference between the actual code width of the first conversion and the ideal code width (1/2LSB). Note, if the first conversion is \$001, then the difference between the actual \$001 code width and its ideal (1LSB) is used.
- Full-Scale Error (E_{FS}) — This error is defined as the difference between the actual code width of the last conversion and the ideal code width (1.5LSB). Note, if the last conversion is \$3FE, then the difference between the actual \$3FE code width and its ideal (1LSB) is used.
- Differential Non-Linearity (DNL) — This error is defined as the worst-case difference between the actual code width and the ideal code width for all conversions.
- Integral Non-Linearity (INL) — This error is defined as the highest-value the (absolute value of the) running sum of DNL achieves. More simply, this is the worst-case difference of the actual transition voltage to a given code and its corresponding ideal transition voltage, for all codes.
- Total Unadjusted Error (TUE) — This error is defined as the difference between the actual transfer function and the ideal straight-line transfer function, and therefore includes all forms of error.

3.3.4.6 Code Jitter, Non-Monotonicity and Missing Codes

Analog-to-digital converters are susceptible to three special forms of error. These are code jitter, non-monotonicity, and missing codes.

- Code jitter is when, at certain points, a given input voltage converts to one of two values when sampled repeatedly. Ideally, when the input voltage is infinitesimally smaller than the transition voltage, the converter yields the lower code (and vice-versa). However, even very small amounts of system noise can cause the converter to be indeterminate (between two codes) for a range of input voltages around the transition voltage. This range is normally around $\pm 1/2\text{LSB}$ but will increase with noise.
- Non-monotonicity is defined as when, except for code jitter, the converter converts to a lower code for a higher input voltage. Non-monotonicity is present if the apparent code jitter covers three codes (when the converter's output is indeterminate between three values for a given input voltage) or is greater than 1LSB.
- Missing codes are those which are never converted for any input value. In 8-bit or 10-bit mode, the ADC10 is guaranteed to be monotonic and to have no missing codes.



Keyboard Interrupt Module (KBI)

9.8.2 Keyboard Interrupt Enable Register (KBIER)

KBIER enables or disables each keyboard interrupt pin.

	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 — Keyboard Interrupt Enable Bits

Each of these read/write bits enables the corresponding keyboard interrupt pin to latch KBI interrupt requests.

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 \(AWU\)](#).

9.8.3 Keyboard Interrupt Polarity Register (KBIPR)

KBIPR determines the polarity of the enabled keyboard interrupt pin and enables the appropriate pullup or pulldown device.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	KBIP5	KBIP4	KBIP3	KBIP2	KBIP1	KBIP0
Write:								
Reset:	0	0	0	0	0	0	0	0

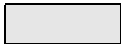
 = Unimplemented

Figure 9-5. Keyboard Interrupt Polarity Register (KBIPR)

KBIP5–KBIP0 — Keyboard Interrupt Polarity Bits

Each of these read/write bits enables the polarity of the keyboard interrupt detection.

1 = Keyboard polarity is high level and/or rising edge

0 = Keyboard polarity is low level and/or falling edge

10.4 LVI Interrupts

The LVI module does not generate interrupt requests.

10.5 Low-Power Modes

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

10.5.1 Wait Mode

If enabled, the LVI module remains active in wait mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of wait mode.

10.5.2 Stop Mode

If the LVIPWRD bit in the configuration register is cleared and the LVISTOP bit in the configuration register is set, the LVI module remains active. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of stop mode.

10.6 Registers

The LVI status register (LVISR) contains a status bit that is useful when the LVI is enabled and LVI reset is disabled.

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

 = Unimplemented
 R = Reserved

Figure 10-2. LVI Status Register (LVISR)

LVIOUT — LVI Output Bit

This read-only flag becomes set when the V_{DD} voltage falls below the V_{TRIPF} trip voltage and is cleared when V_{DD} voltage rises above V_{TRIPR} . (See [Table 10-1](#)).

Table 10-1. LVIOUT Bit Indication

V_{DD}	LVIOUT
$V_{DD} > V_{TRIPR}$	0
$V_{DD} < V_{TRIPF}$	1
$V_{TRIPF} < V_{DD} < V_{TRIPR}$	Previous value

The oscillator configuration uses five components:

- Crystal, X_1
- Fixed capacitor, C_1
- Tuning capacitor, C_2 (can also be a fixed capacitor)
- Feedback resistor, R_B
- Series resistor, R_S (optional)

NOTE

The series resistor (R_S) is included in the diagram to follow strict Pierce oscillator guidelines and may not be required for all ranges of operation, especially with high frequency crystals. Refer to the oscillator characteristics table in the Electricals section for more information.

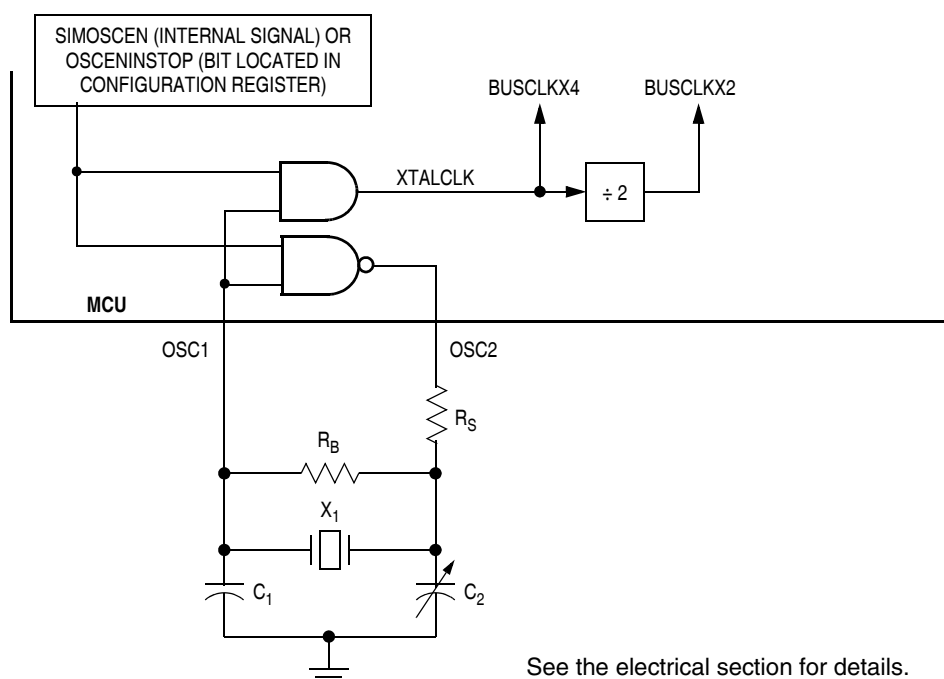


Figure 11-2. XTAL Oscillator External Connections

11.6 OSC During Break Interrupts

There are no status flags associated with the OSC module.

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.

11.7 I/O Signals

The OSC shares its pins with general-purpose input/output (I/O) port pins. See [Figure 11-1](#) for port location of these shared pins.

11.7.1 Oscillator Input Pin (OSC1)

The OSC1 pin is an input to the crystal oscillator amplifier, an input to the RC oscillator circuit, or an input from an external clock source.

When the OSC is configured for internal oscillator, the OSC1 pin can be used as a general-purpose input/output (I/O) port pin or other alternative pin function.

11.7.2 Oscillator Output Pin (OSC2)

For the XTAL oscillator option, the OSC2 pin is the output of the crystal oscillator amplifier.

When the OSC is configured for internal oscillator, external clock, or RC, the OSC2 pin can be used as a general-purpose I/O port pin or other alternative pin function. When the oscillator is configured for internal or RC, the OSC2 pin can be used to output BUSCLKX4.

Table 11-1. OSC2 Pin Function

Option	OSC2 Pin Function
XTAL oscillator	Inverting OSC1
External clock	General-purpose I/O or alternative pin function
Internal oscillator or RC oscillator	Controlled by OSC2EN bit OSC2EN = 0: General-purpose I/O or alternative pin function OSC2EN = 1: BUSCLKX4 output



Chapter 13

System Integration Module (SIM)

13.1 Introduction

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

The SIM is responsible for:

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

Table 13-1. Signal Name Conventions

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

13.2 \overline{RST} and \overline{IRQ} Pins Initialization

\overline{RST} and \overline{IRQ} pins come out of reset as PTA3 and PTA2 respectively. \overline{RST} and \overline{IRQ} functions can be activated by programing CONFIG2 accordingly. Refer to [Chapter 5 Configuration Register \(CONFIG\)](#).

In wait mode, the CPU clocks are inactive. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

Wait mode can also be exited by a reset (or break in emulation mode). A break interrupt during wait mode sets the SIM break stop/wait bit, SBSW, in the break status register (BSR). If the COP disable bit, COPD, in the configuration register is 0, then the computer operating properly module (COP) is enabled and remains active in wait mode.

Figure 13-15 and Figure 13-16 show the timing for wait recovery.

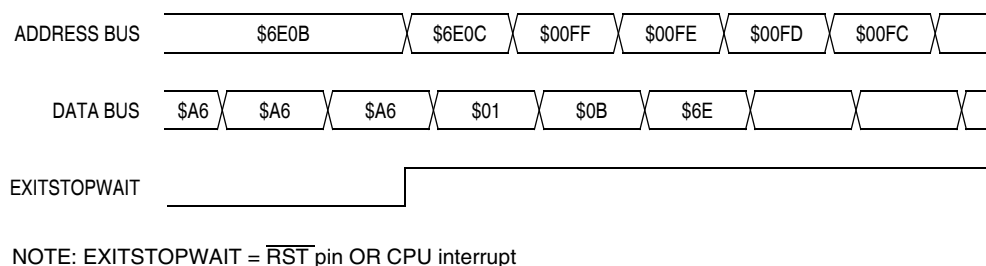


Figure 13-15. Wait Recovery from Interrupt

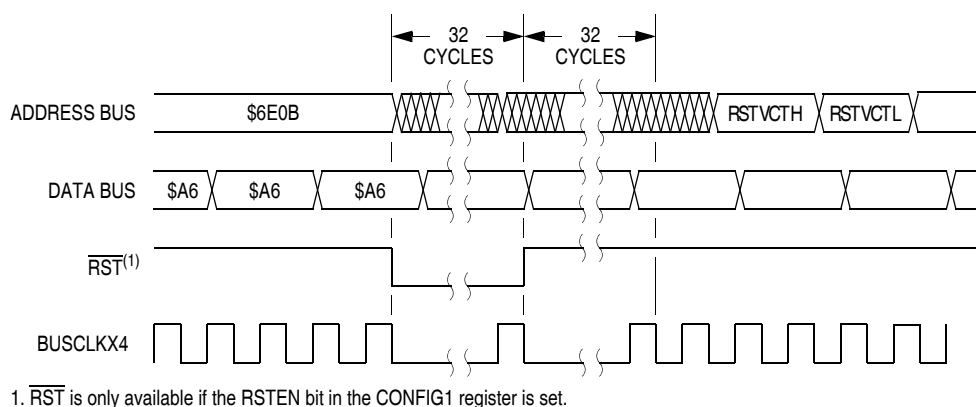


Figure 13-16. Wait Recovery from Internal Reset

13.7.2 Stop Mode

In stop mode, the SIM counter is reset and the system clocks are disabled. An interrupt request from a module can cause an exit from stop mode. Stacking for interrupts begins after the selected stop recovery time has elapsed. Reset or break also causes an exit from stop mode.

The SIM disables the oscillator signals (BUSCLKX2 and BUSCLKX4) in stop mode, stopping the CPU and peripherals. If OSCENINSTOP is set, BUSCLKX2 will remain running in STOP and can be used to run the AWU. Stop recovery time is selectable using the SSREC bit in the configuration register 1 (CONFIG1). If SSREC is set, stop recovery is reduced from the normal delay of 4096 BUSCLKX4 cycles down to 32. This is ideal for the internal oscillator, RC oscillator, and external oscillator options which do not require long start-up times from stop mode.

NOTE

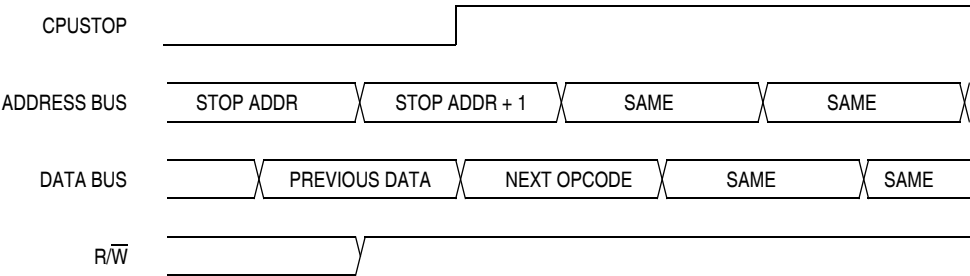
External crystal applications should use the full stop recovery time by clearing the SSREC bit.

System Integration Module (SIM)

The SIM counter is held in reset from the execution of the STOP instruction until the beginning of stop recovery. It is then used to time the recovery period. [Figure 13-17](#) shows stop mode entry timing and [Figure 13-18](#) shows the stop mode recovery time from interrupt or break

NOTE

To minimize stop current, all pins configured as inputs should be driven to a logic 1 or logic 0.



NOTE: Previous data can be operand data or the STOP opcode, depending on the last instruction.

Figure 13-17. Stop Mode Entry Timing

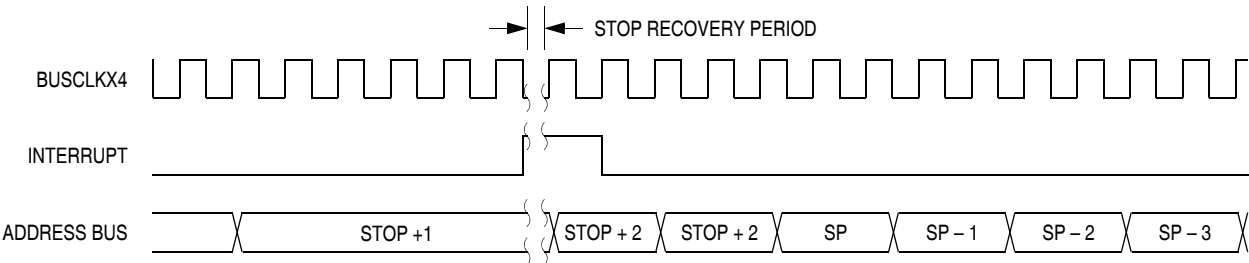


Figure 13-18. Stop Mode Recovery from Interrupt

13.8 SIM Registers

The SIM has two memory mapped registers.

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.

	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)

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. Writing a 1 to TOF has no effect.

- 1 = Counter has reached modulo value
- 0 = 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.

- 1 = TIM overflow interrupts enabled
- 0 = TIM overflow interrupts disabled

TSTOP — TIM Stop Bit

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

- 1 = Counter stopped
- 0 = Counter active

NOTE

Do not set the TSTOP bit before entering wait mode if the TIM is required to exit wait mode. Also, 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.

TRST — TIM Reset Bit

Setting this write-only bit resets the counter and the TIM prescaler. Setting TRST has no effect on any other timer registers. Counting resumes from \$0000. TRST is cleared automatically after the counter is reset and always reads as 0.

- 1 = Prescaler and counter cleared
- 0 = No effect

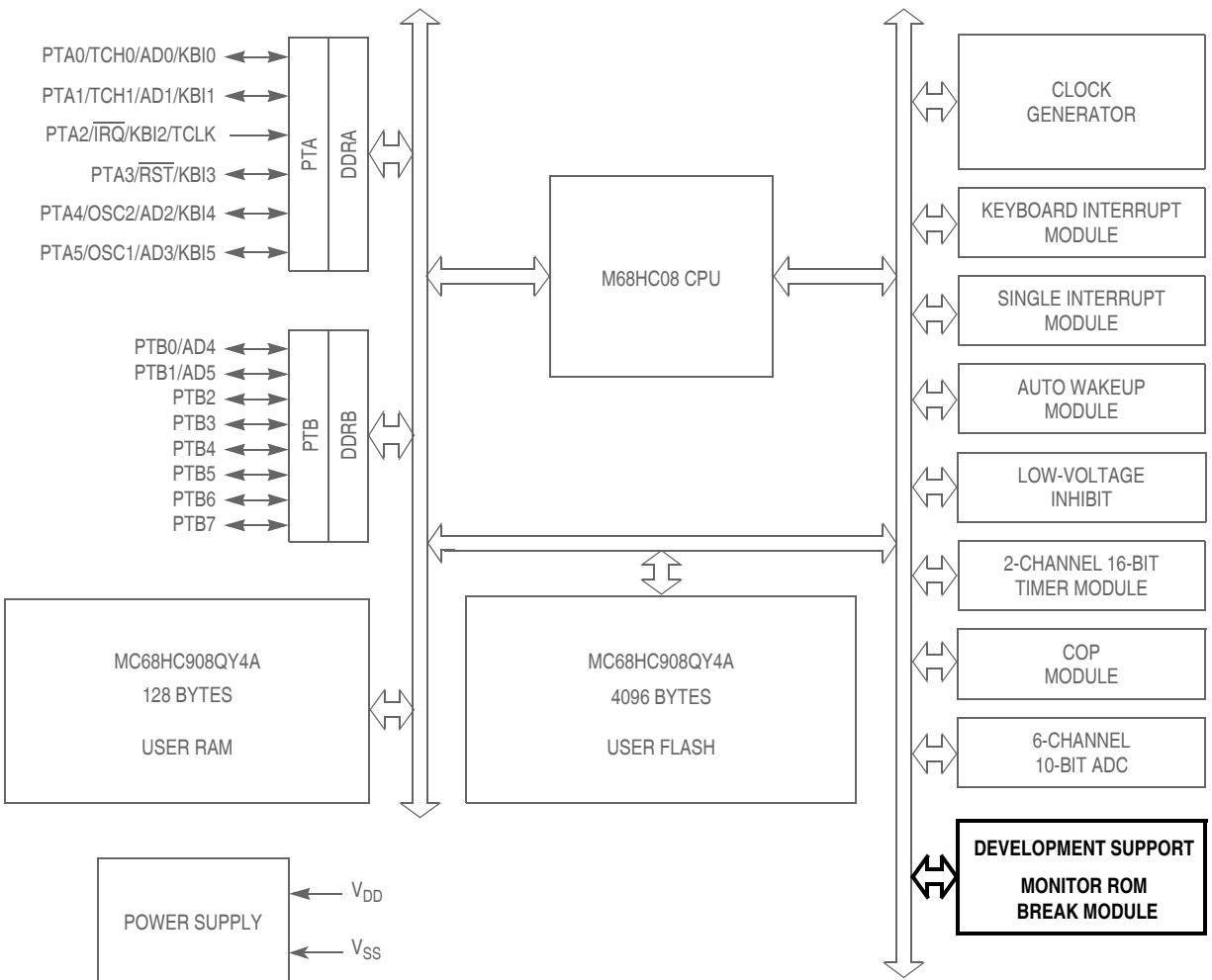
NOTE

Setting the TSTOP and TRST bits simultaneously stops the counter at a value of \$0000. **PS[2:0] — Prescaler Select Bits**

These read/write bits select one of the seven prescaler outputs as the input to the counter as [Table 14-1](#) shows.

Table 14-1. Prescaler Selection

PS2	PS1	PS0	TIM Clock Source
0	0	0	Internal bus clock ÷ 1
0	0	1	Internal bus clock ÷ 2
0	1	0	Internal bus clock ÷ 4
0	1	1	Internal bus clock ÷ 8
1	0	0	Internal bus clock ÷ 16
1	0	1	Internal bus clock ÷ 32
1	1	0	Internal bus clock ÷ 64
1	1	1	TCLK (if available)



$\overline{\text{RST}}$, $\overline{\text{IRQ}}$: 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

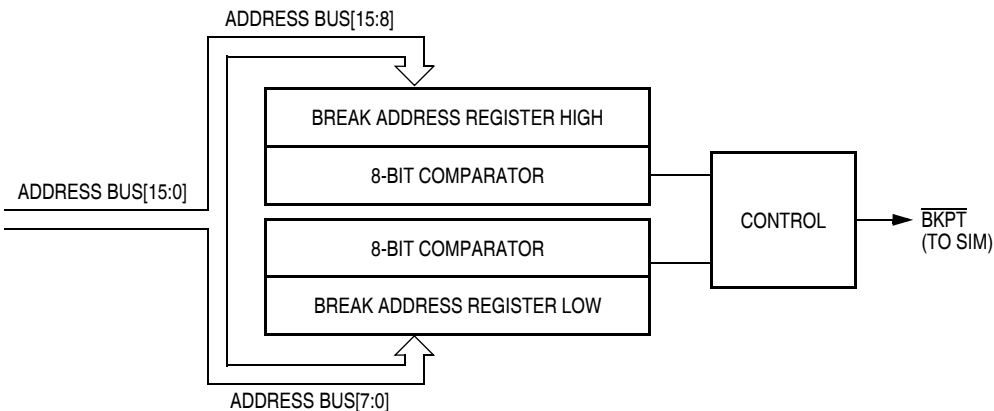


Figure 15-2. Break Module Block Diagram

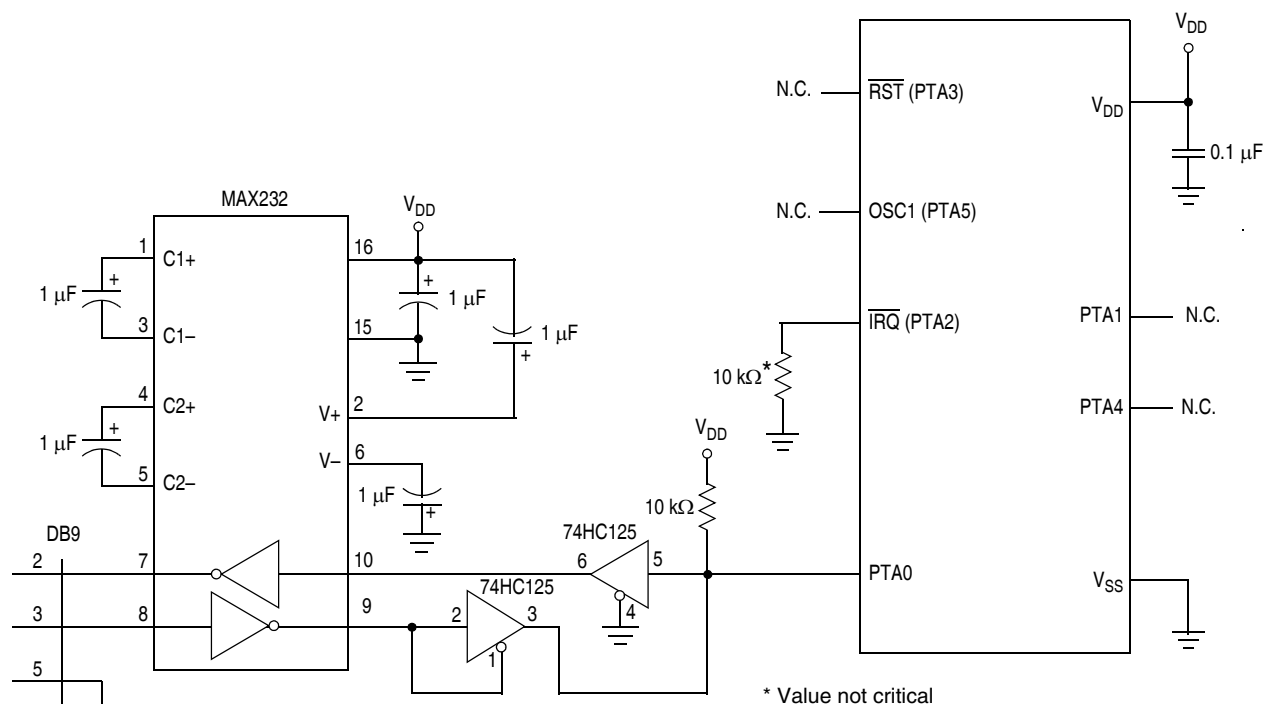


Figure 15-12. Monitor Mode Circuit (Internal Clock, No High Voltage)

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 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_{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)

The rising edge of the internal $\overline{\text{RST}}$ signal latches the monitor mode. Once monitor mode is latched, the values on PTA1 and PTA4 pins can be changed.

Once out of reset, the MCU waits for the host to send eight security bytes (see [15.3.2 Security](#)). After the security bytes, the MCU sends a break signal (10 consecutive 0s) to the host, indicating that it is ready to receive a command.





A.3 Conversion Considerations

Enhancements lead to slight differences in operation from QYx Classic to the QYxA. There are a few points that should be considered in the conversion process.

- The Monitor ROM changed from 2 K to 1 K in size. This has led to the limitation that programming across page boundaries is no longer supported by the on-chip program range routine. Also, in very rare cases, ROM code improvements could cause customers to have to modify a few instructions in their application code. For example, when performing a mass erase, a valid address is required instead of an unspecified address.
- The QYxA contains new modules like the 10-bit ADC and OSC. In rare cases, new modules could cause customers to have to modify a few instructions in their application code. For example, if ADC code was written so that entire registers are configured without respect to reserve bits, then the ADC code will need to be revised to work correctly on the QYxA.
- The Reference Clock for ADC conversions has changed from the bus clock to the system clock (Bus Clock * 4). A change to the divide register may be necessary to set the reference clock to a specified value.

A.4 Code Changes Checklist

Below is a checklist that should be reviewed in the conversion process. This checklist will point out all the issues that should be addressed as your code is ported.

1. Does the original software use Auxiliary ROM routines (for example, Getbyte, Putbyte, delnus)?
If so, the software will have to be changed to handle new Auxiliary ROM routines, addresses of these routines have changed in QYxA. Code will have to be changed to use the proper addresses.
2. Does the software use FLASH as EEPROM?
If so, there are several possible issues for the page erase and mass erase routine. Software will have to be checked to ensure that proper procedure is used and the CTRLBYT is set with a MOV instruction not a BSET. Also, on-chip FLASH programming routines can no longer program across row boundaries
3. Does the code use the auto wake up timer and does the application depend on the typical auto wake time out?
Since the timeout has been improved for QYxA it may be necessary to modify software to compensate for the change in timeout.
4. Bits changed in the OSCSC, CONFIG2, and ADC registers?
Any code that writes to these registers should be reviewed to ensure that the writes are not affecting the changed bits
5. Does the code use external OSC, crystal, or RC?
If so, since the OSCOPT bits have changed locations code will have to be updated to update these bits in their proper locations.
6. Does the code use the ADC?
If so, because on QYxA the ADC clock is driven from 4XBUSCLK instead of BUSCLK changes to the ADC clock divider bits may be needed to maintain proper operation.