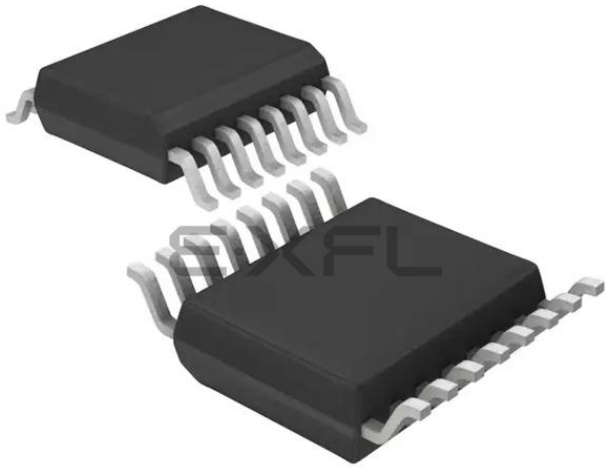

Welcome to E-XFL.COM

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	8MHz
Connectivity	SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	12
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	16-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908qc16cdte

13.3.2.2 Character Transmission . . . 126
13.3.2.3 Break Characters . . . 127
13.3.2.4 Idle Characters . . . 128
13.3.2.5 Inversion of Transmitted Output . . . 128
13.3.3 Receiver . . . 128
13.3.3.1 Character Length . . . 128
13.3.3.2 Character Reception . . . 128
13.3.3.3 Data Sampling . . . 130
13.3.3.4 Framing Errors . . . 131
13.3.3.5 Baud Rate Tolerance . . . 131
13.3.3.6 Receiver Wakeup . . . 133
13.4 Interrupts . . . 134
13.4.1 Transmitter Interrupts . . . 134
13.4.2 Receiver Interrupts . . . 134
13.4.3 Error Interrupts . . . 134
13.5 Low-Power Modes . . . 135
13.5.1 Wait Mode . . . 135
13.5.2 Stop Mode . . . 135
13.6 ESCI During Break Interrupts . . . 135
13.7 I/O Signals . . . 135
13.7.1 ESCI Transmit Data (TxD) . . . 135
13.7.2 ESCI Receive Data (RxD) . . . 136
13.8 Registers . . . 136
13.8.1 ESCI Control Register 1 . . . 136
13.8.2 ESCI Control Register 2 . . . 138
13.8.3 ESCI Control Register 3 . . . 140
13.8.4 ESCI Status Register 1 . . . 141
13.8.5 ESCI Status Register 2 . . . 143
13.8.6 ESCI Data Register . . . 144
13.8.7 ESCI Baud Rate Register . . . 144
13.8.8 ESCI Prescaler Register . . . 145
13.9 ESCI Arbiter . . . 149
13.9.1 ESCI Arbiter Control Register . . . 149
13.9.2 ESCI Arbiter Data Register . . . 150
13.9.3 Bit Time Measurement . . . 150
13.9.4 Arbitration Mode . . . 151

Chapter 14
System Integration Module (SIM)

14.1 Introduction . . . 153
14.2 $\overline{\text{RST}}$ and $\overline{\text{IRQ}}$ Pins Initialization . . . 153
14.3 SIM Bus Clock Control and Generation . . . 153
14.3.1 Bus Timing . . . 155
14.3.2 Clock Start-Up from POR . . . 155
14.3.3 Clocks in Stop Mode and Wait Mode . . . 155
14.4 Reset and System Initialization . . . 155
14.4.1 External Pin Reset . . . 155

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0242	TIM2 Counter Register Low (T2CNTL) See page 214.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0243	TIM2 Counter Modulo Register High (T2MODH) See page 215.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0244	TIM2 Counter Modulo Register Low (T2MODL) See page 215.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0245	TIM2 Channel 0 Status and Control Register (T2SC0) See page 215.	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0246	TIM2 Channel 0 Register High (T2CH0H) See page 218.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0247	TIM2 Channel 0 Register Low (T2CH0L) See page 218.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0248	TIM2 Channel 1 Status and Control Register (T2SC1) See page 215.	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0249	TIM2 Channel 1 Register High (T2CH1H) See page 218.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$024A	TIM2 Channel 1 Register Low (T2CH1L) See page 218.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$024B	Reserved									
$024C	Periodic Wakeup Status and Control Register (PWUSC) See page 119.	Read:	0	0	PWUON	PWUCLK-SEL	PWUF	0	PWUIE	SMODE
		Write:						PWUACK		
		Reset:	0	0	0	0	0	0	0	0
$024D	Periodic Wakeup Prescaler Register (PWUP) See page 120.	Read:	0	0	0	0	PS3	PS2	PS1	PS0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$024E	Periodic Wakeup Modulo Register (PWUMOD) See page 121.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 6 of 8)

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

Do not program any byte in the FLASH more than once after a successful erase operation. Reprogramming bits to a byte which is already programmed is not allowed without first erasing the page in which the byte resides or mass erasing the entire FLASH memory. Programming without first erasing may disturb data stored in the FLASH.

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}.
5. Set the HVEN bit.
6. Wait for a time, t_{PGS}.
7. Write data to the FLASH address being programmed(1).
8. Wait for time, t_{PROG}.
9. Repeat step 7 and 8 until all desired bytes within the row are programmed.
10. Clear the PGM bit (1).
11. Wait for time, t_{NVH}.
12. Clear the HVEN bit.
13. After time, t_{RCV}, the memory can be accessed in read mode again.

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_{PROG} maximum, see 19.17 Memory Characteristics.

1. 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.

clocks are too fast, then the clock must be divided to the appropriate frequency. This divider is specified by ADIV[1:0] 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 COCO. An interrupt request is generated if AIEN is set 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.

Upon reset or when a conversion is otherwise aborted, the ADC10 module will enter a low power, inactive state. In this state, all internal clocks and references are disabled. This state is entered asynchronously and immediately upon aborting of a conversion.

3.3.3.4 Total Conversion Time

The total conversion time depends on many factors such as sample time, bus frequency, whether ACLKEN is set, and synchronization time. The total conversion time is summarized in Table 3-1.

Table 3-1. Total Conversion Time versus Control Conditions

Conversion Mode	ACLKEN	Maximum Conversion Time
8-Bit Mode (short sample — ADLSMP = 0):		
Single or 1st continuous	0	18 ADCK + 3 bus clock
Single or 1st continuous	1	18 ADCK + 3 bus clock + 5 µs
Subsequent continuous ($f_{Bus} \geq f_{ADCK}$)	X	16 ADCK
8-Bit Mode (long sample — ADLSMP = 1):		
Single or 1st continuous	0	38 ADCK + 3 bus clock
Single or 1st continuous	1	38 ADCK + 3 bus clock + 5 µs
Subsequent continuous ($f_{Bus} \geq f_{ADCK}$)	X	36 ADCK
10-Bit Mode (short sample — ADLSMP = 0):		
Single or 1st continuous	0	21 ADCK + 3 bus clock
Single or 1st continuous	1	21 ADCK + 3 bus clock + 5 µs
Subsequent continuous ($f_{Bus} \geq f_{ADCK}$)	X	19 ADCK
10-Bit Mode (long sample — ADLSMP = 1):		
Single or 1st continuous	0	41 ADCK + 3 bus clock
Single or 1st continuous	1	41 ADCK + 3 bus clock + 5 µs
Subsequent continuous ($f_{Bus} \geq f_{ADCK}$)	X	39 ADCK

The maximum total conversion time for a single conversion or the first conversion in continuous conversion mode is determined by the clock source chosen and the divide ratio selected. The clock source is selectable by ADICLK and ACLKEN, and the divide ratio is specified by ADIV. For example, if the alternative clock source is 16 MHz and is selected as the input clock source, the input clock divide-by-8 ratio is selected and the bus frequency is 4 MHz, then the conversion time for a single 10-bit conversion is:

$$\text{Conversion time} = \frac{\text{21 ADCK cycles}}{\text{16 MHz/8}} + \frac{\text{3 bus cycles}}{\text{4 MHz}} = 11.25\ \mu s$$

Number of bus cycles = 11.25 µs x 4 MHz = 45 cycles

NOTE

The ADCK frequency must be between f_{ADCK} minimum and f_{ADCK} maximum to meet A/D specifications.

3.8.2 ADC10 Result High Register (ADRH)

This register holds the MSBs of the result and is updated each time a conversion completes. All other bits read as 0s. Reading ADRH prevents the ADC10 from transferring subsequent conversion results into the result registers until ADRL is read. If ADRL is not read until the after next conversion is completed, then the intermediate conversion result will be lost. In 8-bit mode, this register contains no interlocking with ADRL.

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

= Unimplemented

Figure 3-4. ADC10 Data Register High (ADRH), 8-Bit Mode

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	AD9	AD8
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 3-5. ADC10 Data Register High (ADRH), 10-Bit Mode

3.8.3 ADC10 Result Low Register (ADRL)

This register holds the LSBs of the result. This register is updated each time a conversion completes. Reading ADRH prevents the ADC10 from transferring subsequent conversion results into the result registers until ADRL is read. If ADRL is not read until the after next conversion is completed, then the intermediate conversion result will be lost. In 8-bit mode, there is no interlocking with ADRH.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 3-6. ADC10 Data Register Low (ADRL)

3.8.4 ADC10 Clock Register (ADCLK)

This register selects the clock frequency for the ADC10 and the modes of operation.

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	ADLPC	ADIV1	ADIV0	ADICLK	MODE1	MODE0	ADLSMP	ACLKEN
Reset:	0	0	0	0	0	0	0	0

Figure 3-7. ADC10 Clock Register (ADCLK)

IRQPUD — $\overline{\text{IRQ}}$ Pin Pullup Control Bit

1 = Internal pullup is disconnected
0 = Internal pullup is connected between $\overline{\text{IRQ}}$ pin and V_{DD}

IRQEN — $\overline{\text{IRQ}}$ Pin Function Selection Bit

1 = Interrupt request function active in pin
0 = Interrupt request function inactive in pin

TIM2POS — TIM2 Position Bit

TIM2POS is used to reposition the timer channels for TIM2 to a different pair of pins. This allows the user to free up one of the communication ports based on application needs.

1 = TIM2 timer channel pins share PTB4 and PTB5
0 = TIM2 timer channel pins share PTB1 and PTB2

ESCIBDSRC — ESCI Baud Rate Clock Source Bit

ESCIBDSRC controls the clock source used for the ESCI. The setting of the bit affects the frequency at which the ESCI operates.

1 = Internal data bus clock used as clock source for ESCI
0 = BUSCLKX4 used as clock source for ESCI

OSCENINSTOP— Oscillator Enable in Stop Mode Bit

OSCENINSTOP, when set, will allow the clock source to continue to generate clocks in stop mode. This function can be used to keep the periodic wakeup running while the rest of the microcontroller stops. When clear, the clock source is disabled when the microcontroller enters stop mode.

1 = Oscillator enabled to operate during stop mode
0 = Oscillator disabled during stop mode

RSTEN — $\overline{\text{RST}}$ Pin Function Selection

1 = Reset function active in pin
0 = Reset function inactive in pin

NOTE

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

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVITRIP	SSREC	STOP	COPD
Reset:	0	0	0	0	U	0	0	0
POR:	0	0	0	0	0	0	0	0

U = Unaffected

Figure 4-2. Configuration Register 1 (CONFIG1)

COPRS — COP Reset Period Selection Bit

1 = COP reset short cycle = 8176 × BUSCLKX4
0 = COP reset long cycle = 262,128 × BUSCLKX4

8.3.1 Keyboard Operation

Writing to the KBIEx bits in the keyboard interrupt enable register (KBIER) independently enables or disables each KBI pin. The polarity of the keyboard interrupt is controlled using the KBIPx bits in the keyboard interrupt polarity register (KBIPR). Edge-only or edge and level sensitivity is controlled using the MODEK bit in the keyboard status and control register (KBISCR).

Enabling a keyboard interrupt pin also enables its internal pullup or pulldown device based on the polarity enabled. On falling edge or low level detection, a pullup device is configured. On rising edge or high level detection, a pulldown device is configured.

The keyboard interrupt latch is set when one or more enabled keyboard interrupt inputs are asserted.

- If the keyboard interrupt sensitivity is edge-only, for KBIPx = 0, a falling (for KBIPx = 1, a rising) edge on a keyboard interrupt input does not latch an interrupt request if another enabled keyboard pin is already asserted. To prevent losing an interrupt request on one input because another input remains asserted, software can disable the latter input while it is asserted.
- If the keyboard interrupt is edge and level sensitive, an interrupt request is present as long as any enabled keyboard interrupt input is asserted.

8.3.1.1 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.

The KBI vector fetch or software clear and the return of all enabled keyboard interrupt pins to a deasserted level may occur in any order.

Reset clears the keyboard interrupt request and the MODEK bit, clearing the interrupt request even if a keyboard interrupt input stays asserted.

8.3.1.2 MODEK = 0

If the MODEK bit is clear, the keyboard interrupt inputs are edge sensitive. The KBIPx bit will determine whether an edge sensitive pin detects rising or falling edges. A KBI vector fetch or software clear immediately clears the KBI latch.

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.

10.7 I/O Signals

The OSC shares its pins with general-purpose input/output (I/O) port pins. See Figure 10-1 for port location of these shared pins.

10.7.1 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.

10.7.2 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 10-1. 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

10.8 Registers

The oscillator module contains two registers:

- Oscillator status and control register (OSCSC)
- Oscillator trim register (OSCTRIM)

10.8.1 Oscillator Status and Control Register

The oscillator status and control register (OSCSC) contains the bits for switching between internal and external clock sources. If the application uses an external crystal, bits in this register are used to select the crystal oscillator amplifier necessary for the desired crystal. While running off the internal clock source, the user can use bits in this register to select the internal clock source frequency.

PTY — Parity Bit

This read/write bit determines whether the ESCI generates and checks for odd parity or even parity (see Table 13-4).

1 = Odd parity
0 = Even parity

NOTE

Changing the PTY bit in the middle of a transmission or reception can generate a parity error.

13.8.2 ESCI Control Register 2

ESCI control register 2 (SCC2):

- Enables these interrupt requests:
  - SCTE bit to generate transmitter interrupt requests
  - TC bit to generate transmitter interrupt requests
  - SCRF bit to generate receiver interrupt requests
  - IDLE bit to generate receiver interrupt requests
- Enables the transmitter
- Enables the receiver
- Enables ESCI wakeup
- Transmits ESCI break characters

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0

Figure 13-10. ESCI Control Register 2 (SCC2)

SCTIE — ESCI Transmit Interrupt Enable Bit

This read/write bit enables the SCTE bit to generate ESCI transmitter interrupt requests. Setting the SCTIE bit in SCC2 enables the SCTE bit to generate interrupt requests.

1 = SCTE enabled to generate interrupt
0 = SCTE not enabled to generate interrupt

TCIE — Transmission Complete Interrupt Enable Bit

This read/write bit enables the TC bit to generate ESCI transmitter interrupt requests.

1 = TC enabled to generate interrupt requests
0 = TC not enabled to generate interrupt requests

SCRIE — ESCI Receive Interrupt Enable Bit

This read/write bit enables the SCRF bit to generate ESCI receiver interrupt requests. Setting the SCRIE bit in SCC2 enables the SCRF bit to generate interrupt requests.

1 = SCRF enabled to generate interrupt
0 = SCRF not enabled to generate interrupt

14.4.2.1 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 14-6.

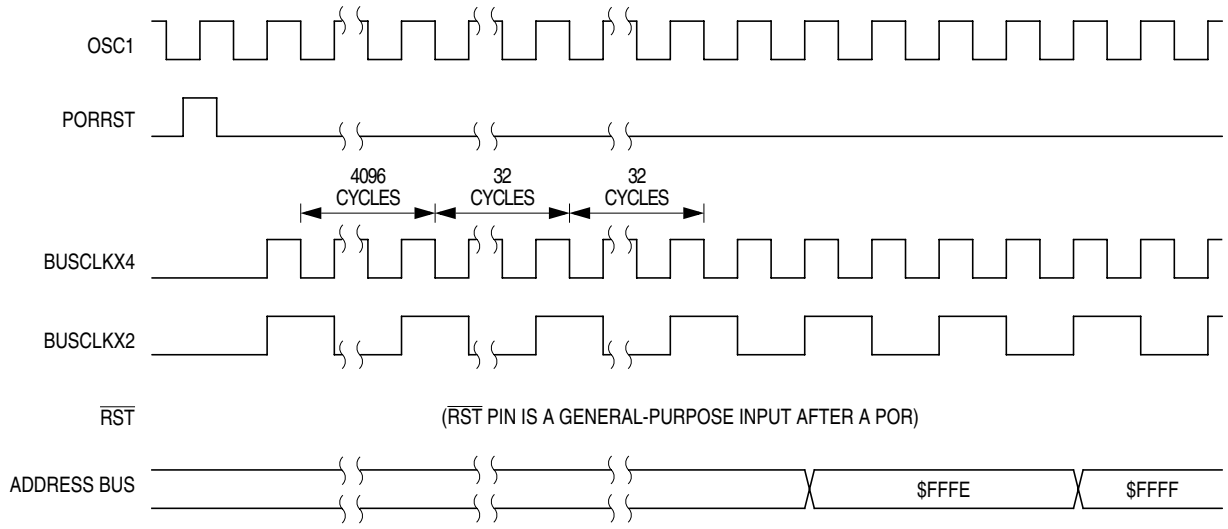

Figure 14-6. POR Recovery

14.4.2.2 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 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).

15.3.5 Resetting the SPI

Any system reset completely resets the SPI. Partial resets occur whenever the SPI enable bit (SPE) is 0. Whenever SPE is 0, the following occurs:

- The SPTE flag is set.
- Any transmission currently in progress is aborted.
- The shift register is cleared.
- The SPI state counter is cleared, making it ready for a new complete transmission.
- All the SPI pins revert back to being general-purpose I/O.

These items are reset only by a system reset:

- All control bits in the SPCR register
- All control bits in the SPSCR register (MODFEN, ERRIE, SPR1, and SPR0)
- The status flags SPRF, OVRF, and MODF

By not resetting the control bits when SPE is low, the user can clear SPE between transmissions without having to set all control bits again when SPE is set high for the next transmission.

By not resetting the SPRF, OVRF, and MODF flags, the user can still service these interrupts after the SPI has been disabled. The user can disable the SPI by writing 0 to the SPE bit. The SPI can also be disabled by a mode fault occurring in an SPI that was configured as a master with the MODFEN bit set.

15.3.6 Error Conditions

The following flags signal SPI error conditions:

- Overflow (OVRF) — Failing to read the SPI data register before the next full byte enters the shift register sets the OVRF bit. The new byte does not transfer to the receive data register, and the unread byte still can be read. OVRF is in the SPI status and control register.
- Mode fault error (MODF) — The MODF bit indicates that the voltage on the slave select pin ($\overline{SS}$) is inconsistent with the mode of the SPI. MODF is in the SPI status and control register.

15.3.6.1 Overflow Error

The overflow flag (OVRF) becomes set if the receive data register still has unread data from a previous transmission when the capture strobe of bit 1 of the next transmission occurs. The bit 1 capture strobe occurs in the middle of SPSCK cycle 7 (see Figure 15-4 and Figure 15-6.) If an overflow occurs, all data received after the overflow and before the OVRF bit is cleared does not transfer to the receive data register and does not set the SPI receiver full bit (SPRF). The unread data that transferred to the receive data register before the overflow occurred can still be read. Therefore, an overflow error always indicates the loss of data. Clear the overflow flag by reading the SPI status and control register and then reading the SPI data register.

OVRF generates a receiver/error interrupt request if the error interrupt enable bit (ERRIE) is also set. The SPRF, MODF, and OVRF interrupts share the same interrupt vector (see Figure 15-11.) It is not possible to enable MODF or OVRF individually to generate a receiver/error interrupt request. However, leaving MODFEN low prevents MODF from being set.

If the SPRF interrupt is enabled and the OVRF interrupt is not, watch for an overflow condition. Figure 15-9 shows how it is possible to miss an overflow. The first part of Figure 15-9 shows how it is possible to read the SPSCR and SPDR to clear the SPRF without problems. However, as illustrated by

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
Reset:	0	0	1	0	1	0	0	0
	R	= Reserved						

Figure 15-13. SPI Control Register (SPCR)

SPRIE — SPI Receiver Interrupt Enable Bit

This read/write bit enables interrupt requests generated by the SPRF bit. The SPRF bit is set when a byte transfers from the shift register to the receive data register.

1 = SPRF interrupt requests enabled
0 = SPRF interrupt requests disabled

SPMSTR — SPI Master Bit

This read/write bit selects master mode operation or slave mode operation.

1 = Master mode
0 = Slave mode

CPOL — Clock Polarity Bit

This read/write bit determines the logic state of the SPSCK pin between transmissions. (See Figure 15-4 and Figure 15-6.) To transmit data between SPI modules, the SPI modules must have identical CPOL values.

CPHA — Clock Phase Bit

This read/write bit controls the timing relationship between the serial clock and SPI data. (See Figure 15-4 and Figure 15-6.) To transmit data between SPI modules, the SPI modules must have identical CPHA values. When CPHA = 0, the $\overline{SS}$ pin of the slave SPI module must be high between bytes. (See Figure 15-12.)

SPWOM — SPI Wired-OR Mode Bit

This read/write bit configures pins SPSCK, MOSI, and MISO so that these pins become open-drain outputs.

1 = Wired-OR SPSCK, MOSI, and MISO pins
0 = Normal push-pull SPSCK, MOSI, and MISO pins

SPE — SPI Enable

This read/write bit enables the SPI module. Clearing SPE causes a partial reset of the SPI. (See 15.3.5 Resetting the SPI.)

1 = SPI module enabled
0 = SPI module disabled

SPTIE— SPI Transmit Interrupt Enable

This read/write bit enables interrupt requests generated by the SPTE bit. SPTE is set when a byte transfers from the transmit data register to the shift register.

1 = SPTE interrupt requests enabled
0 = SPTE interrupt requests disabled

16.3.1 TIM1 Counter Prescaler

The TIM1 clock source is one of the seven prescaler outputs or the external clock input pin, T1CLK if available. The prescaler generates seven clock rates from the internal bus clock. The prescaler select bits, PS[2:0], in the TIM1 status and control register (T1SC) select the clock source.

16.3.2 Input Capture

With the input capture function, the TIM1 can capture the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the TIM1 latches the contents of the counter into the TIM1 channel registers, T1CHxH:T1CHxL. The polarity of the active edge is programmable. Input captures can be enabled to generate interrupt requests.

16.3.3 Output Compare

With the output compare function, the TIM1 can generate a periodic pulse with a programmable polarity, duration, and frequency. When the counter reaches the value in the registers of an output compare channel, the TIM1 can set, clear, or toggle the channel pin. Output compares can be enabled to generate interrupt requests.

16.3.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in 16.3.3 Output Compare. The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIM1 channel registers.

An unsynchronized write to the TIM1 channel registers to change an output compare value could cause incorrect operation for up to two counter overflow periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that counter overflow period. Also, using a TIM1 overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIM1 may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the output compare value on channel x:

- When changing to a smaller value, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current output compare pulse. The interrupt routine has until the end of the counter overflow period to write the new value.
- When changing to a larger output compare value, enable TIM1 overflow interrupts and write the new value in the TIM1 overflow interrupt routine. The TIM1 overflow interrupt occurs at the end of the current counter overflow period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same counter overflow period.

16.3.3.2 Buffered Output Compare

Channels 0 and 1 can be linked to form a buffered output compare channel whose output appears on the T1CH0 pin. The TIM1 channel registers of the linked pair alternately control the output.

Setting the MS0B bit in TIM1 channel 0 status and control register (T1SC0) links channel 0 and channel 1. The output compare value in the TIM1 channel 0 registers initially controls the output on the T1CH0 pin. Writing to the TIM1 channel 1 registers enables the TIM1 channel 1 registers to synchronously control the

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 17-1 shows.

Table 17-1. Prescaler Selection

PS2	PS1	PS0	TIM2 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	T2CLK (if available)

17.8.2 TIM2 Counter Registers

The two read-only TIM2 counter registers contain the high and low bytes of the value in the counter. Reading the high byte (T2CNTH) latches the contents of the low byte (T2CNTL) into a buffer. Subsequent reads of T2CNTH do not affect the latched T2CNTL value until T2CNTL is read. Reset clears the TIM2 counter registers. Setting the TIM2 reset bit (TRST) also clears the TIM2 counter registers.

NOTE

If you read T2CNTH during a break interrupt, be sure to unlatch T2CNTL by reading T2CNTL before exiting the break interrupt. Otherwise, T2CNTL retains the value latched during the break.

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

Figure 17-5. TIM2 Counter High Register (T2CNTH)

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

= Unimplemented

Figure 17-6. TIM2 Counter Low Register (T2CNTL)

The average chip-junction temperature (T_J) in °C can be obtained from:

$$T_J = T_A + (P_D \times \theta_{JA}) \quad \text{Eqn. 19-1}$$

where:

T_A = Ambient temperature, °C
θ_{JA} = Package thermal resistance, junction-to-ambient, °C/W
$P_D = P_{int} + P_{I/O}$
$P_{int} = I_{DD} \times V_{DD}$, Watts — chip internal power
$P_{I/O}$ = Power dissipation on input and output pins — user determined

For most applications, $P_{I/O} << P_{int}$ and can be neglected. An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad \text{Eqn. 19-2}$$

Solving Equation 19-1 and Equation 19-2 for K gives:

$$K = P_D \times (T_A + 273°C) + \theta_{JA} \times (P_D)^2 \quad \text{Eqn. 19-3}$$

where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving Equation 19-1 and Equation 19-2 iteratively for any value of T_A.

19.5 5-V DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage I_{Load} = –2.0 mA, all I/O pins I_{Load} = –10.0 mA, all I/O pins I_{Load} = –15.0 mA, PTA0, PTA1, PTA3–PTA5 only	V_{OH}	 V_{DD}–0.4 V_{DD}–1.5 V_{DD}–0.8	 — — —	 — — —	V
Maximum combined I_{OH} (all I/O pins)	I_{OHT}	—	—	50	mA
Output low voltage I_{Load} = 1.6 mA, all I/O pins I_{Load} = 10.0 mA, all I/O pins I_{Load} = 15.0 mA, PTA0, PTA1, PTA3–PTA5 only	V_{OL}	 — — —	 — — —	 0.4 1.5 0.8	V
Maximum combined I_{OL} (all I/O pins)	I_{OHL}	—	—	50	mA
Input high voltage PTA0–PTA5, PTB0–PTB7, PTC3–PTC0, PTD7–PTD0	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input low voltage PTA0–PTA5, PTB0–PTB7, PTC3–PTC0, PTD7–PTD0	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
Input hysteresis	V_{HYS}	0.06 x V_{DD}	—	—	V

— Continued on next page

19.10 3.3-V Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency	f_{OP} (f_{Bus})	—	4	MHz
Internal clock period ($1/f_{OP}$)	t_{CYC}	250	—	ns
$\overline{RST}$ input pulse width low[2]	t_{RL}	200	—	ns
$\overline{IRQ}$ interrupt pulse width low (edge-triggered)[2]	t_{ILIH}	200	—	ns
$\overline{IRQ}$ interrupt pulse period[2]	t_{ILIL}	Note[3]	—	t_{CYC}

1. V_{DD} = 3.0 to 3.6 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H; timing shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Values are based on characterization results, not tested in production.
3. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1 t_{CYC}.

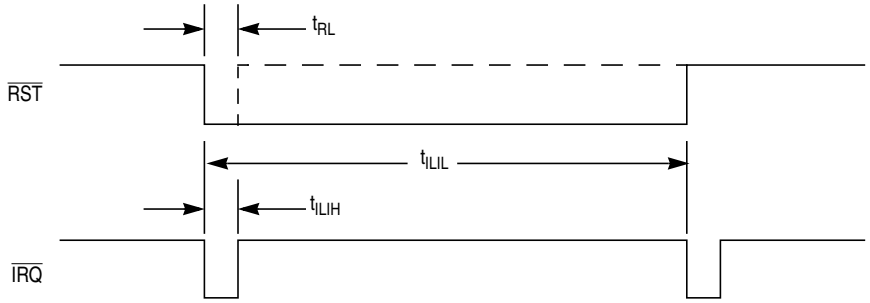

Figure 19-6. $\overline{RST}$ and $\overline{IRQ}$ Timing

19.11 Oscillator Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Internal oscillator frequency[1] ICFS1:ICFS0 = 00 ICFS1:ICFS0 = 01 ICFS1:ICFS0 = 10 ICFS1:ICFS0 = 11 (not allowed if V_{DD} < 4.5 V)	f_{INTCLK}	 — — — —	 4 8 12.8 25.6	 — — — —	MHz
Trim accuracy[2][3]	Δ_{TRIM_ACC}	—	± 0.4	—	%
Deviation from trimmed Internal oscillator[3][4] 4, 8, 12.8, 25.6MHz, V_{DD} ± 10%, 0 to 70°C 4, 8, 12.8, 25.6MHz, V_{DD} ± 10%, –40 to 125°C	Δ_{INT_TRIM}	 — —	 ± 2 —	 — ± 5	%
External RC oscillator frequency, RCCLK[1] [3] $V_{DD} \geq 4.5$ V $V_{DD} < 4.5$ V	f_{RCCLK}	 2 2	 — —	 12 8.4	MHz
External clock reference frequency [1] [5] [6] $V_{DD} \geq 4.5$ V $V_{DD} \geq 3.0$ V	$f_{OSCXCLK}$	 dc dc	 — —	 32 16	MHz

— Continued on next page

Chapter 20
Ordering Information and Mechanical Specifications

20.1 Introduction

This section contains order numbers for the MC68HC908QC16, MC68HC908QC8, and MC68HC908QC4. See Table 20-1 and Figure 20-1.

20.2 MC Order Numbers

Table 20-1. MC Order Numbers

	Temp. Range	16 TSSOP	16 SOIC	20 TSSOP	20 SOIC	28 TSSOP	28 SOIC
Automotive	C = –40°C to 85°C	S908QC16CDTE(R)		S908QC16CDSE		S908QC16CDRE	
		S908QC8CDTE		S908QC8CDSE		S908QC8CDRE	
	V = –40°C to 105°C			S908QC16VDSE			
				S908QC8VDSE			
	M = –40°C to 125°C	S908QC16MDTE		S908QC16MDSE(R)		S908QC16MDRE	
		S908QC8MDTE		S908QC8MDSE(R)		S908QC8 MDRE	
				S908QC4MDSE(R)		S908AC4MDRE	
Consumer and Industrial	C = –40°C to 85°C	MC908QC16CDTE	MC908QC16CDXE	MC908QC16CDSE	MC908QC16CDYE	MC908QC16CDRE	MC908QC16CDZE
		MC908QC8CDTE	MC908QC8CDXE	MC908QC8CDSE	MC908QC8CDYE	MC908QC8CDRE	MC908QC8CDZE
						MC908QC4CDRE	
	V = –40°C to 105°C			MC908QC16VDSE		MC908QC16VDRE	
				MC908QC8VDSE		MC908QC8VDRE	

Temperature designators:
C = –40°C to +85°C
V = –40°C to +105°C
M = –40°C to +125°C

Package designators:
DX = 16-pin SOIC
DY = 20-pin SOIC
DZ = 28-pin SOIC
DT = 16-pin TSSOP
DS = 20-pin TSSOP
DR = 28-pin TSSOP

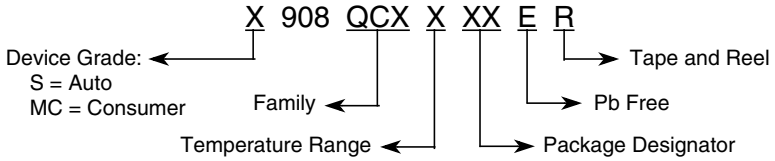

Figure 20-1. Device Numbering System

20.3 Package Dimensions

Refer to the following pages for detailed package dimensions.

NOTES:

1. DIMENSIONS ARE IN MILLIMETERS.

2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M–1994.

3. DATUMS A AND B TO BE DETERMINED AT THE PLANE WHERE THE BOTTOM OF THE LEADS EXIT THE PLASTIC BODY.

4. THIS DIMENSION DOES NOT INCLUDE MOLD FLASH, PROTRUSION OR GATE BURRS. MOLD FLASH, PROTRUSION OR GATE BURRS SHALL NOT EXCEED 0.15 MM PER SIDE. THIS DIMENSION IS DETERMINED AT THE PLANE WHERE THE BOTTOM OF THE LEADS EXIT THE PLASTIC BODY.

5. THIS DIMENSION DOES NOT INCLUDE INTER–LEAD FLASH OR PROTRUSIONS. INTER–LEAD FLASH AND PROTRUSIONS SHALL NOT EXCEED 0.25 MM PER SIDE. THIS DIMENSION IS DETERMINED AT THE PLANE WHERE THE BOTTOM OF THE LEADS EXIT THE PLASTIC BODY.

6. THIS DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED 0.62 mm.

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE:	DOCUMENT NO: 98ASB42343B	REV: J
20LD SOIC W/B, 1.27 PITCH, CASE OUTLINE	CASE NUMBER: 751D–07	23 MAR 2005
	STANDARD: JEDEC MS–013AC	