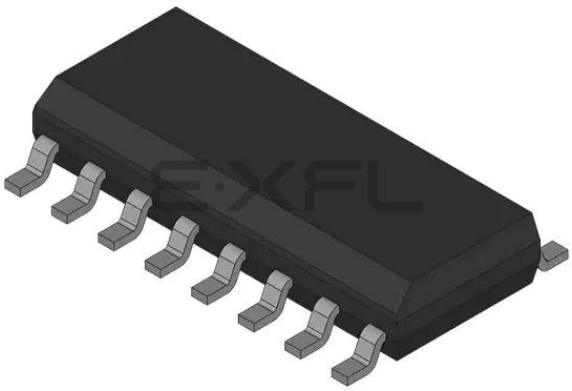

Welcome to E-XFL.COM

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	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	6KB (6K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	External
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/pro/item?MUrl=&PartUrl=mc908ql3cdwe

Chapter 9
Keyboard Interrupt Module (KBI)

Chapter 10
Low-Voltage Inhibit (LVI)

Chapter 11
Oscillator Module (OSC)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FFBF	Reserved									
$FFC0	Internal Oscillator Trim Value (Optional)	Read: Write:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
		Reset:	Resets to factory programmed value							
$FFC1	Reserved									
$FFFF	COP Control Register (COPCTL) See page 69.	Read:	LOW BYTE OF RESET VECTOR							
		Write:	WRITING CLEARS COP COUNTER (ANY VALUE)							
		Reset:	Unaffected by reset							

= Unimplemented R = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 7 of 7)

Table 2-1. Vector Addresses

Vector Priority	Vector	Address	Vector
Lowest	IF22–IF16	$FFD0,1–$FFDC,D	Unused vectors
	IF15	$FFDE,F	ADC conversion complete vector
	IF14	$FFE0,1	Keyboard vector
	IF13	$FFE2,3	Unused vector
	IF12	$FFE4,5	Unused vector
	IF11	$FFE6,7	Unused vector
	IF10	$FFE8,9	Unused vector
	IF9	$FFEA,B	SLIC vector
	IF8	$FFFC,D	Unused vector
	IF7	$FFEE,F	Unused vector
	IF6	$FFF0,1	Unused vector
	IF5	$FFF2,3	TIM overflow vector
	IF4	$FFF4,5	TIM channel 1 vector
	IF3	$FFF6,7	TIM channel 0 vector
	IF2	$FFF8,9	Unused vector
	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)
```

2.6 FLASH Memory (FLASH)

The FLASH memory is intended primarily for program storage. In-circuit programming allows the operating program to be loaded into the FLASH memory after final assembly of the application product. It is possible to program the entire array through the single-wire monitor mode interface. Because no special voltages are needed for FLASH erase and programming operations, in-application programming is also possible through other software-controlled communication paths.

This subsection describes the operation of the embedded FLASH memory. The FLASH memory can be read, programmed, and erased from the internal V_{DD} supply. The program and erase operations are enabled through the use of an internal charge pump.

The minimum size of FLASH memory that can be erased is 64 bytes; and the maximum size of FLASH memory that can be programmed in a program cycle is 32 bytes (a row). Program and erase operations are facilitated through control bits in the FLASH control register (FLCR). Details for these operations appear later in this section.

NOTE

An erased bit reads as a 1 and a programmed bit reads as a 0. A security feature prevents viewing of the FLASH contents.[1]

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.

2.6.1 FLASH Control Register

The FLASH control register (FLCR) controls FLASH program and erase operations.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 2-3. FLASH Control Register (FLCR)

HVEN — High Voltage Enable Bit

This read/write bit enables high voltage from the charge pump to the memory for either program or erase operation. It can only be set if either PGM =1 or ERASE =1 and the proper sequence for program or erase is followed.

1 = High voltage enabled to array and charge pump on
0 = High voltage disabled to array and charge pump off

MASS — Mass Erase Control Bit

This read/write bit configures the memory for mass erase operation.

1 = Mass erase operation selected
0 = Mass erase operation unselected

ERASE — Erase Control Bit

This read/write bit configures the memory for erase operation. ERASE is interlocked with the PGM bit such that both bits cannot be equal to 1 or set to 1 at the same time.

1 = Erase operation selected
0 = Erase operation unselected

PGM — Program Control Bit

This read/write bit configures the memory for program operation. PGM is interlocked with the ERASE bit such that both bits cannot be equal to 1 or set to 1 at the same time.

1 = Program operation selected
0 = Program operation unselected

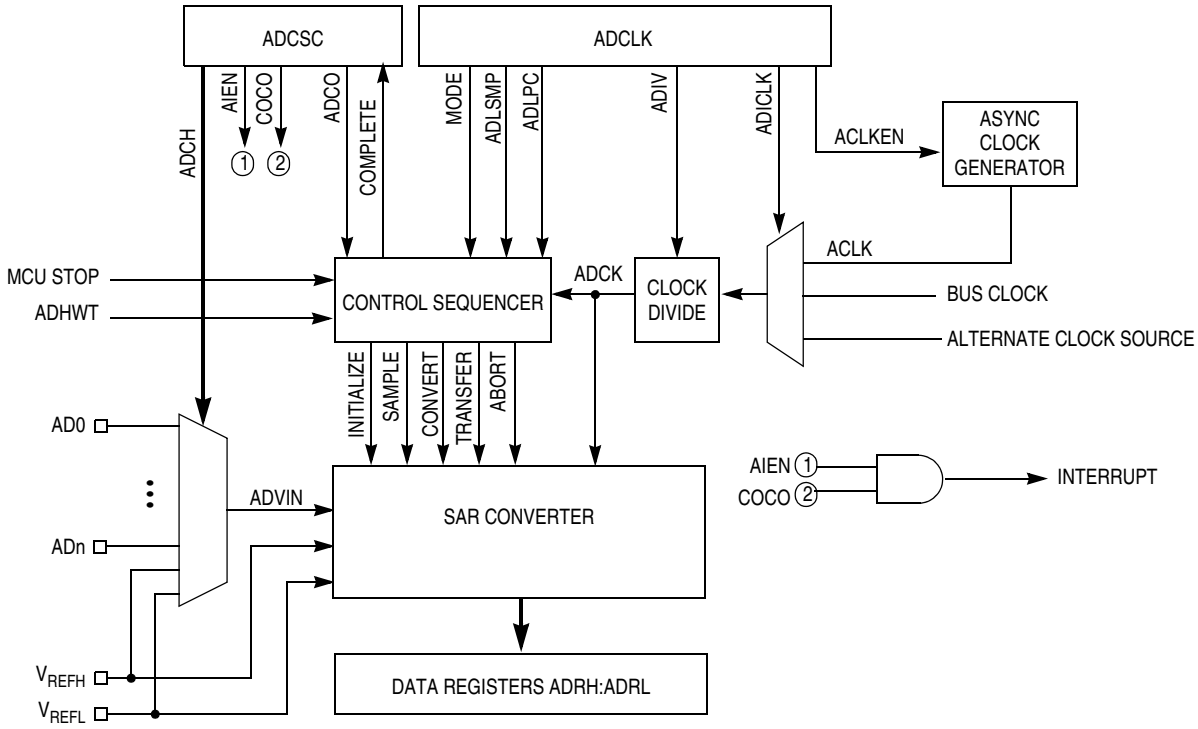

Figure 3-2. ADC10 Block Diagram

The ADC10 can perform an analog-to-digital conversion on one of the software selectable channels. The output of the input multiplexer (ADVIN) is converted by a successive approximation algorithm into a 10-bit digital result. When the conversion is completed, the result is placed in the data registers (ADRH and ADRL). In 8-bit mode, the result is rounded to 8 bits and placed in ADRL. The conversion complete flag is then set and an interrupt is generated if the interrupt has been enabled.

3.3.1 Clock Select and Divide Circuit

The clock select and divide circuit selects one of three clock sources and divides it by a configurable value to generate the input clock to the converter (ADCK). The clock can be selected from one of the following sources:

- The asynchronous clock source (ACLK) — This clock source is generated from a dedicated clock source which is enabled when the ADC10 is converting and the clock source is selected by setting the ACLKEN bit. When the ADLPC bit is clear, this clock operates from 1–2 MHz; when ADLPC is set it operates at 0.5–1 MHz. This clock is not disabled in STOP and allows conversions in stop mode for lower noise operation.
- Alternate Clock Source — This clock source is equal to the external oscillator clock or a four times the bus clock. The alternate clock source is MCU specific, see 3.1 Introduction to determine source and availability of this clock source option. This clock is selected when ADICLK and ACLKEN are both low.
- The bus clock — This clock source is equal to the bus frequency. This clock is selected when ADICLK is high and ACLKEN is low.

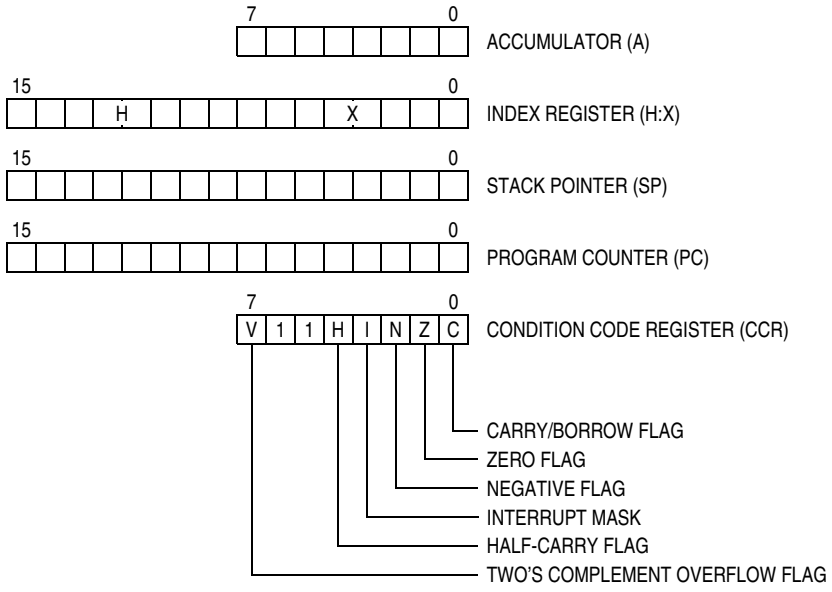

Figure 7-1. CPU Registers

7.3.1 Accumulator

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and the results of arithmetic/logic operations.

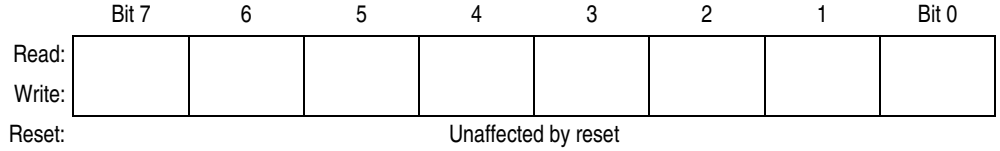

Figure 7-2. Accumulator (A)

7.3.2 Index Register

The 16-bit index register allows indexed addressing of a 64-Kbyte memory space. H is the upper byte of the index register, and X is the lower byte. H:X is the concatenated 16-bit index register.

In the indexed addressing modes, the CPU uses the contents of the index register to determine the conditional address of the operand.

The index register can serve also as a temporary data storage location.

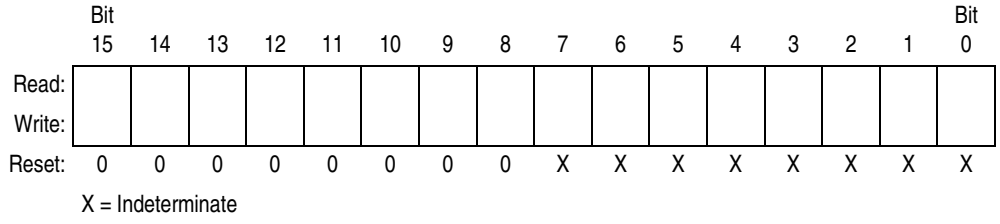

Figure 7-3. Index Register (H:X)

Chapter 8
External Interrupt (IRQ)

8.1 Introduction

The IRQ (external interrupt) module provides a maskable interrupt input.

IRQ functionality is enabled by setting configuration register 2 (CONFIG2) IRQEN bit accordingly. A zero disables the IRQ function and $\overline{\text{IRQ}}$ will assume the other shared functionalities. A one enables the IRQ function. See Chapter 5 Configuration Register (CONFIG) for more information on enabling the $\overline{\text{IRQ}}$ pin.

The $\overline{\text{IRQ}}$ pin shares its pin with general-purpose input/output (I/O) port pins. See Figure 8-1 for port location of this shared pin.

8.2 Features

Features of the IRQ module include:

- A dedicated external interrupt pin $\overline{\text{IRQ}}$
- IRQ interrupt control bits
- Programmable edge-only or edge and level interrupt sensitivity
- Automatic interrupt acknowledge
- Internal pullup device

8.3 Functional Description

A low level applied to the external interrupt request ($\overline{\text{IRQ}}$) pin can latch a CPU interrupt request. Figure 8-2 shows the structure of the IRQ module.

Interrupt signals on the $\overline{\text{IRQ}}$ pin are latched into the IRQ latch. The IRQ latch remains set until one of the following actions occurs:

- IRQ vector fetch. An IRQ vector fetch automatically generates an interrupt acknowledge signal that clears the latch that caused the vector fetch.
- Software clear. Software can clear the IRQ latch by writing a 1 to the ACK bit in the interrupt status and control register (INTSCR).
- Reset. A reset automatically clears the IRQ latch.

The external $\overline{\text{IRQ}}$ pin is falling edge sensitive out of reset and is software-configurable to be either falling edge or falling edge and low level sensitive. The MODE bit in INTSCR controls the triggering sensitivity of the $\overline{\text{IRQ}}$ pin.

The LVI module contains a bandgap reference circuit and comparator. When the LVITRIP bit is cleared, the default state at power-on reset, V_{TRIPF} is configured for the lower V_{DD} operating range. The actual trip points are specified in 17.5 5-V DC Electrical Characteristics and 17.8 3.3-V DC Electrical Characteristics.

Because the default LVI trip point after power-on reset is configured for low voltage operation, a system requiring high voltage LVI operation must set the LVITRIP bit during system initialization. V_{DD} must be above the LVI trip rising voltage, V_{TRIPR}, for the high voltage operating range or the MCU will immediately go into LVI reset.

After an LVI reset occurs, the MCU remains in reset until V_{DD} rises above V_{TRIPR}. See Chapter 13 System Integration Module (SIM) for the reset recovery sequence.

The output of the comparator controls the state of the LVIOUT flag in the LVI status register (LVISR) and can be used for polling LVI operation when the LVI reset is disabled.

The LVI is enabled out of reset. The following bits located in the configuration register can alter the default conditions.

- Setting the LVI power disable bit, LVIPWRD, disables the LVI.
- Setting the LVI reset disable bit, LVIRSTD, prevents the LVI module from generating a reset.
- Setting the LVI enable in stop mode bit, LVISTOP, enables the LVI to operate in stop mode.
- Setting the LVI trip point bit, LVITRIP, configures the trip point voltage (V_{TRIPF}) for the higher V_{DD} operating range.

10.3.1 Polled LVI Operation

In applications that can operate at V_{DD} levels below the V_{TRIPF} level, software can monitor V_{DD} by polling the LVIOUT bit. In the configuration register, LVIPWRD must be cleared to enable the LVI module, and LVIRSTD must be set to disable LVI resets.

10.3.2 Forced Reset Operation

In applications that require V_{DD} to remain above the V_{TRIPF} level, enabling LVI resets allows the LVI module to reset the MCU when V_{DD} falls below the V_{TRIPF} level. In the configuration register, LVIPWRD and LVIRSTD must be cleared to enable the LVI module and to enable LVI resets.

10.3.3 LVI Hysteresis

The LVI has hysteresis to maintain a stable operating condition. After the LVI has triggered (by having V_{DD} fall below V_{TRIPF}), the MCU will remain in reset until V_{DD} rises above the rising trip point voltage, V_{TRIPR}. This prevents a condition in which the MCU is continually entering and exiting reset if V_{DD} is approximately equal to V_{TRIPF}. V_{TRIPR} is greater than V_{TRIPF} by the typical hysteresis voltage, V_{HYS}.

10.3.4 LVI Trip Selection

LVITRIP in the configuration register selects the LVI protection range. The default setting out of reset is for the low voltage range. Because LVITRIP is in a write-once configuration register, the protection range cannot be changed after initialization.

NOTE

The MCU is guaranteed to operate at a minimum supply voltage. The trip point (V_{TRIPF}) may be lower than this. See Chapter 17 Electrical Specifications for the actual trip point voltages.

Chapter 11
Oscillator Module (OSC)

11.1 Introduction

The oscillator (OSC) module is used to provide a stable clock source for the MCU system and bus.

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. The OSC2EN bit is located in the port A pull enable register (PTAPUEN) on this MCU. See Chapter 12 Input/Output Ports (PORTS) for information on PTAPUEN register.

11.2 Features

The bus clock frequency is one fourth of any of these clock source options:

1. Internal oscillator: An internally generated, fixed frequency clock, trimmable to ± 0.4%. There are four choices for the internal oscillator, 25.6 MHz, 12.8 MHz, 8 MHz or 4 MHz. The 12.8-MHz internal oscillator is the default option out of reset.
2. External oscillator: An external clock that can be driven directly into OSC1.
3. External RC: A built-in oscillator module (RC oscillator) that requires an external R connection only. The capacitor is internal to the chip.
4. External crystal: A built-in XTAL oscillator that requires an external crystal or ceramic-resonator. There are three crystal frequency ranges supported, 8–32 MHz, 1–8 MHz, and 32–100 kHz.

11.3 Functional Description

The oscillator contains these major subsystems:

- Internal oscillator circuit
- Internal or external clock switch control
- External clock circuit
- External crystal circuit
- External RC clock circuit

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

Interrupts are latched, and arbitration is performed in the SIM at the start of interrupt processing. The arbitration result is a constant that the CPU uses to determine which vector to fetch. Once an interrupt is latched by the SIM, no other interrupt can take precedence, regardless of priority, until the latched interrupt is serviced (or the I bit is cleared).

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. Figure 13-8 shows interrupt entry timing. Figure 13-9 shows interrupt recovery timing.

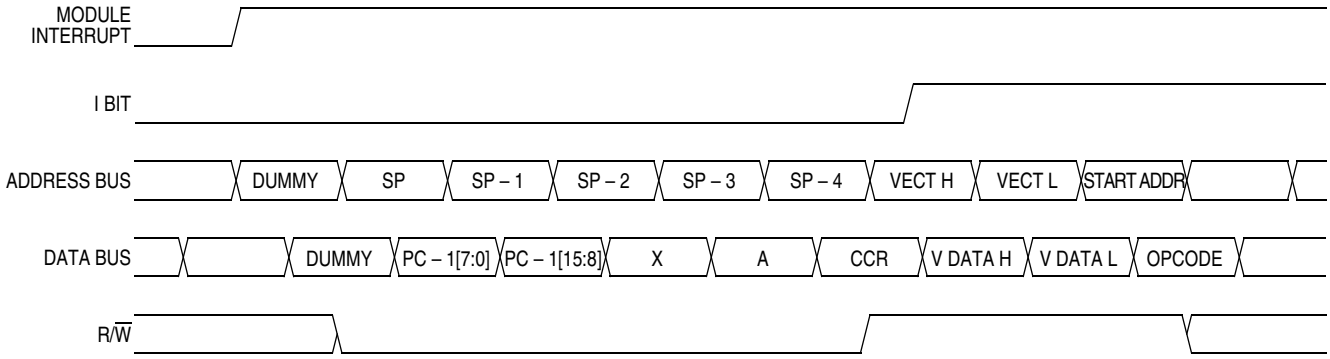

Figure 13-8. Interrupt Entry

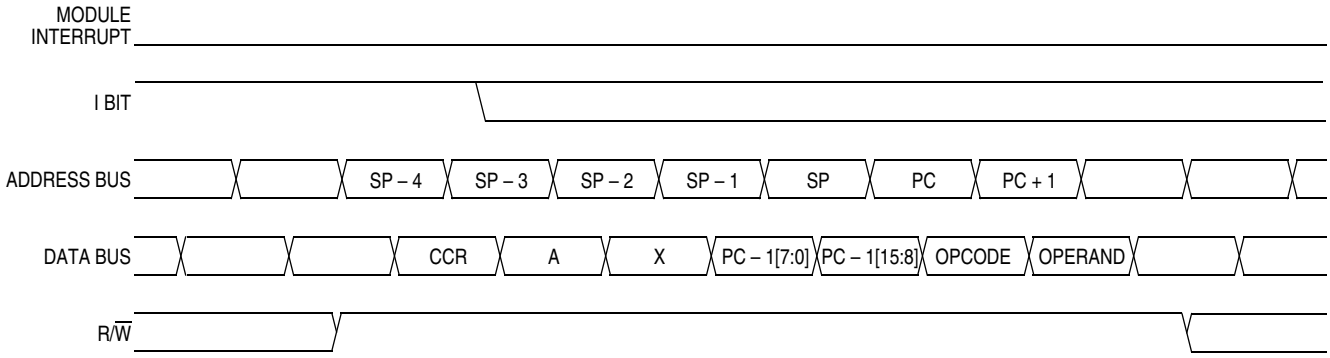

Figure 13-9. Interrupt Recovery

14.7 I/O Signals

The SLIC module can share its two pins with the general-purpose I/O pins. See Figure 14-1 for the port pins that are shared.

14.7.1 SLCTX — SLIC Transmit Pin

The SLCTX pin serves as the serial output of the SLIC module.

14.7.2 SLCRX — SLIC Receive Pin

The SLCRX pin serves as the serial input of the SLIC module. This input feeds directly into the digital receive filter block which filters out noise glitches from the incoming data stream.

14.8 Registers

14.8.1 SLIC Control Register 1

SLIC control register 1 (SLCC1) contains bits used to control various basic features of the SLIC module, including features used for initialization and at runtime.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	INITREQ	0	WAKETX	TXABRT	IMSG	SLCIE
Write:								
Reset:	0	0	1	0	0	0	0	0

= Unimplemented

Figure 14-4. SLIC Control Register 1 (SLCC1)

INITREQ —Initialization Request

Requesting initialization mode by setting this bit will place the SLIC module into its initialized state immediately. If transmission or reception of data is in progress, the transaction will be terminated immediately upon entry into initialization mode (signified by INITACK being set to 1). To return to normal SLIC operation after the SLIC has been initialized (the INITACK is high), the INITREQ must be cleared by software.

1 = Request for SLIC to be put into reset state immediately
0 = Normal operation

WAKETX — Transmit Wakeup Symbol

This bit allows the user to transmit a wakeup symbol on the LIN bus. When set, this sends a wakeup symbol, as defined in the LIN specification a single time, then resets to 0. This bit will read 1 while the wakeup symbol is being transmitted on the bus. This bit will be automatically cleared when the wakeup symbol is complete.

1 = Send wakeup symbol on LIN bus
0 = Normal operation

TXABRT — Transmit Abort Message

1 = Transmitter aborts current transmission at next byte boundary; TXABRT resets to 0 after the transmission is successfully aborted
0 = Normal operation

checksum on the first data byte. Using CHKMOD in this way allows the SLIC to receive messages with either method of data consistency check and change on a frame-by-frame basis. If a system uses both types of data consistency checking methods, the software must simply change the setting of this bit based on the identifier of each message. If the network only uses one type of check, CHKMOD can be set as a constant value in the user's code. If CHKMOD is not written on each frame, care must be taken not to accidentally modify the bit when writing the data length and TXGO bits. This is especially true if using C code without carefully inspecting the output of the compiler and assembler.

The control bits and data length code are contained in one register, allowing the user to maximize the efficiency of the identifier processing by writing a single byte value to indicate the nature of the message frame. This allows very efficient identifier processing code, which is important in a command frame, as the master node can be sending data immediately following the identifier byte which might be as little as one byte in length. The SLIC module uses a separate internal storage area for the incoming data bytes, so there is no danger of losing incoming data, but the user should spend as little time as possible within the ISR to ensure that the application or other ISRs are able to use the majority of the CPU bandwidth.

The identifier must be processed in a maximum of 2 byte times on the LIN bus to ensure that the ISR completes before the checksum would be received for the shortest possible message. This should be easily achievable, as the only operations required are to read SLCID and look up the checksum method, data length, and command/request state of that identifier, then write that value to the SLCDLC. This can be easily streamlined in code with a lookup table of identifiers and corresponding SLCDLC bytes.

NOTE

Once the ID is decoded for a message header and a length code written to SLCDLC, the SLIC is expecting that number of bytes to be received. If the SLIC module doesn't receive the number of bytes indicated in the SLCDLC register, it will continue to look for data bytes. If another message header begins, a byte framing error will trigger on the break symbol of that second message. The second message will still properly generate an ID received interrupt, but the byte framing error prior to this is an indication to the application that the previous message was not properly handled and should be discarded.

14.9.8.2 Extended Command Message Frames

Handling of extended frames is very similar to handling of standard frames, providing that the length is less than or equal to 64 bytes. Because the SLIC module can only receive 8 bytes at a time, the receive buffer must be emptied periodically for long message frames. This is not standard LIN operation, and is likely only to be used for downloading calibration data or reprogramming FLASH devices in a factory or service facility, so the added steps required for processing are not as critical to performance. During these types of operations, the application code is likely very limited in scope and special adjustments can be made to compensate for added message processing time.

For extended command frames, the data length is still written one time at the time the identifier is decoded, along with the TXGO and CHKMOD bits. When this is done, a software counter must also be initialized to keep track of how many bytes are expected to be received in the message frame. The ISR completes, clearing the SLCF flag, and resumes application execution. The SLIC will generate an interrupt, if unmasked, after 8 bytes are received or an error is detected. At this interrupt, the SLCSV will indicate an error condition (in case of byte framing error, idle bus) or that the receive buffer is full. If the data is successfully received, the user must then empty the buffer by reading SLCD7-SLCD0 and then subtract 8 from the software byte count. When this software counter reaches 8 or fewer, the remaining

15.7.1 TIM Channel I/O Pins (TCH1:TCH0)

Each channel I/O pin is programmable independently as an input capture pin or an output compare pin. TCH0 can be configured as buffered output compare or buffered PWM pin.

15.7.2 TIM Clock Pin (TCLK)

TCLK is an external clock input that can be the clock source for the counter instead of the prescaled internal bus clock. Select the TCLK input by writing 1s to the three prescaler select bits, PS[2:0]. The minimum TCLK pulse width is specified in the Timer Interface Module Characteristics table in the Electricals section. The maximum TCLK frequency is the least of 4 MHz or bus frequency ÷ 2.

15.8 Registers

The following registers control and monitor operation of the TIM:

- TIM status and control register (TSC)
- TIM control registers (TCNTH:TCNTL)
- TIM counter modulo registers (TMODH:TMODL)
- TIM channel status and control registers (TSC0 and TSC1)
- TIM channel registers (TCH0H:TCH0L and TCH1H:TCH1L)

15.8.1 TIM Status and Control Register

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

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

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

= Unimplemented

Figure 15-4. TIM Status and Control Register (TSC)

TOF — TIM Overflow Flag Bit

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

1 = Counter has reached modulo value
0 = Counter has not reached modulo value

16.2.2.3 Break Auxiliary Register

The break auxiliary register (BRKAR) contains a bit that enables software to disable the COP while the MCU is in a state of break interrupt with monitor mode.

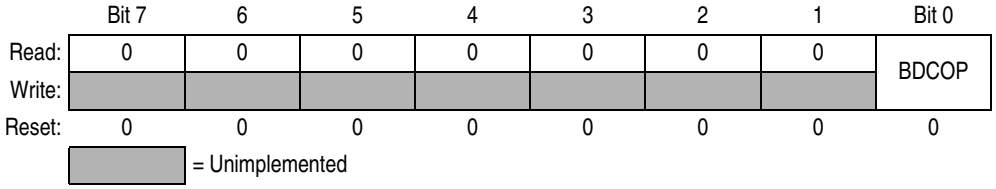

Figure 16-6. Break Auxiliary Register (BRKAR)

BDCOP — Break Disable COP Bit

This read/write bit disables the COP during a break interrupt. Reset clears the BDCOP bit.

1 = COP disabled during break interrupt
0 = COP enabled during break interrupt

16.2.2.4 Break Status Register

The break status register (BSR) contains a flag to indicate that a break caused an exit from wait mode. This register is only used in emulation mode.

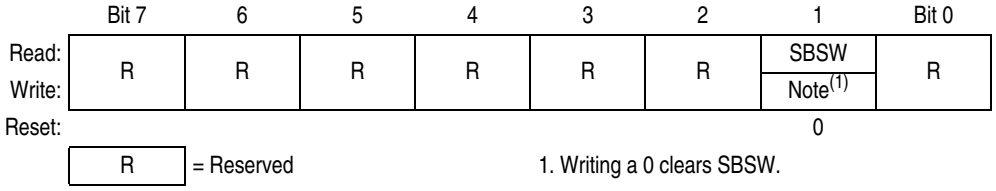

Figure 16-7. Break Status Register (BSR)

SBSW — SIM Break Stop/Wait

SBSW can be read within the break state SWI routine. The user can modify the return address on the stack by subtracting one from it.

1 = Wait mode was exited by break interrupt
0 = Wait mode was not exited by break interrupt

16.2.2.5 Break Flag Control Register

The break control register (BFCR) contains a bit that enables software to clear status bits while the MCU is in a break state.

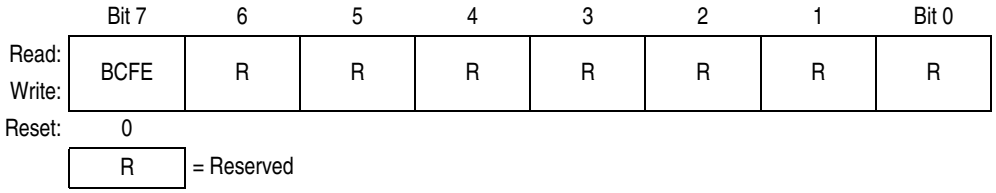

Figure 16-8. Break Flag Control Register (BFCR)

BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

1 = Status bits clearable during break
0 = Status bits not clearable during break

16.3.1.4 Data Format

Communication with the monitor ROM is in standard non-return-to-zero (NRZ) mark/space data format. Transmit and receive baud rates must be identical.

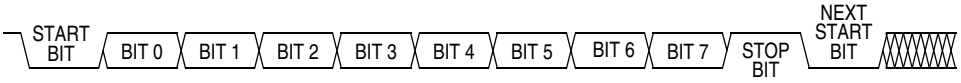

Figure 16-13. Monitor Data Format

16.3.1.5 Break Signal

A start bit (0) followed by nine 0 bits is a break signal. When the monitor receives a break signal, it drives the PTA0 pin high for the duration of approximately two bits and then echoes back the break signal.

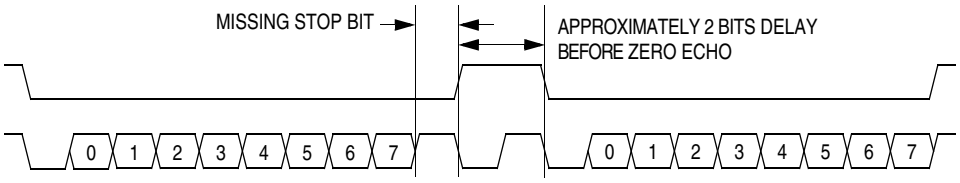

Figure 16-14. Break Transaction

16.3.1.6 Baud Rate

The monitor communication baud rate is controlled by the frequency of the external or internal oscillator and the state of the appropriate pins as shown in Table 16-1.

Table 16-1 also lists the bus frequencies to achieve standard baud rates. The effective baud rate is the bus frequency divided by 256 when using an external oscillator. When using the internal oscillator in forced monitor mode, the effective baud rate is the bus frequency divided by 335.

16.3.1.7 Commands

The monitor ROM firmware uses these commands:

- READ (read memory)
- WRITE (write memory)
- IREAD (indexed read)
- IWRITE (indexed write)
- READSP (read stack pointer)
- RUN (run user program)

The monitor ROM firmware echoes each received byte back to the PTA0 pin for error checking. An 11-bit delay at the end of each command allows the host to send a break character to cancel the command. A delay of two bit times occurs before each echo and before READ, IREAD, or READSP data is returned. The data returned by a read command appears after the echo of the last byte of the command.

NOTE

Wait one bit time after each echo before sending the next byte.

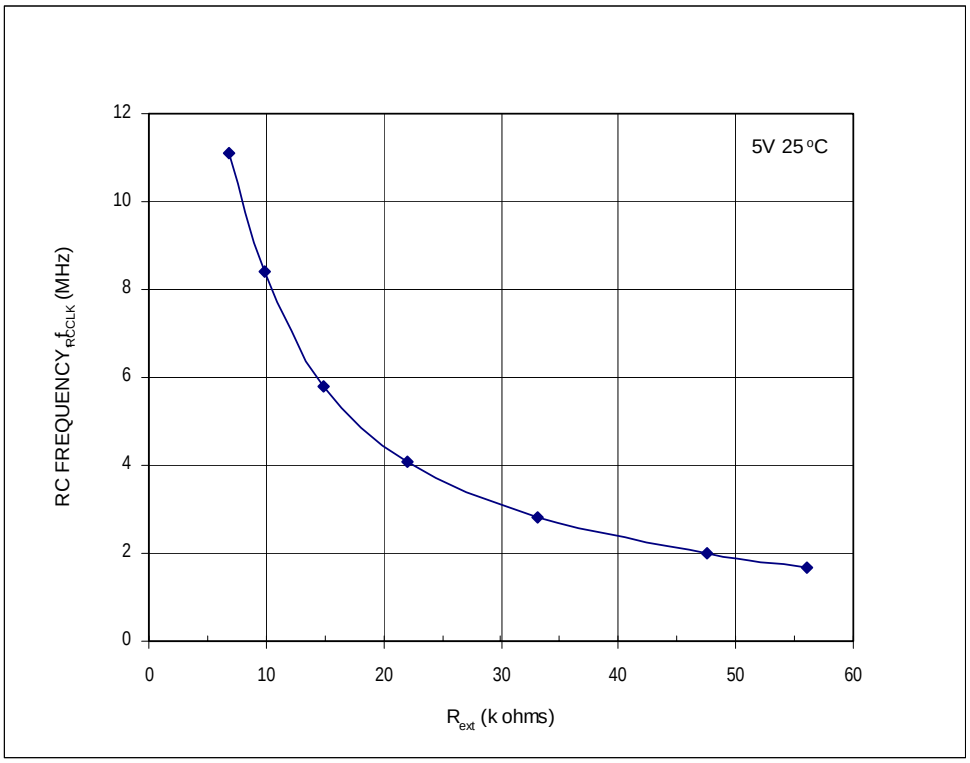

Figure 17-7. RC versus Frequency (5 Volts @ 25°C)

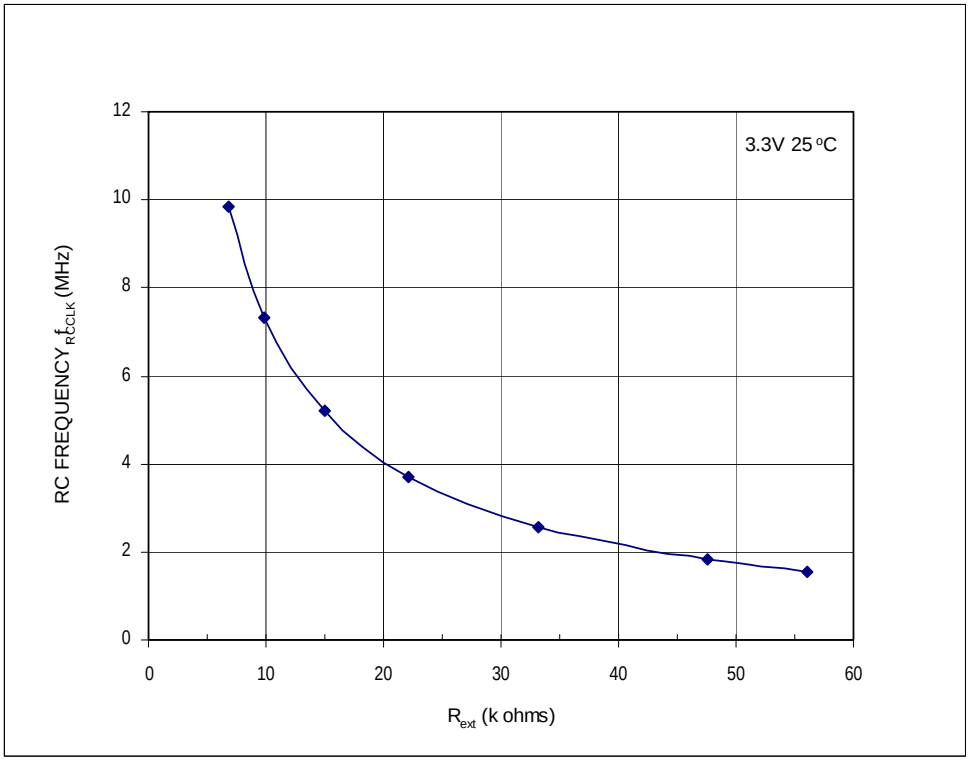

Figure 17-8. RC versus Frequency (3.3 Volts @ 25°C)

Characteristic	Conditions	Symbol	Min	Typ(1)	Max	Unit	Comment
Integral non-linearity	10-bit mode	INL	0	±0.5	—	LSB	
	8-bit mode		0	±0.3	—		
Zero-scale error	10-bit mode	E_{ZS}	0	±0.5	—	LSB	$V_{ADIN} = V_{SS}$
	8-bit mode		0	±0.3	—		
Full-scale error	10-bit mode	E_{FS}	0	±0.5	—	LSB	$V_{ADIN} = V_{DD}$
	8-bit mode		0	±0.3	—		
Quantization error	10-bit mode	E_Q	—	—	±0.5	LSB	8-bit mode is not truncated
	8-bit mode		—	—	±0.5		
Input leakage error	10-bit mode	E_{IL}	0	±0.2	±5	LSB	Pad leakage(5) * R_{AS}
	8-bit mode		0	±0.1	±1.2		
Bandgap voltage input(6)		V_{BG}	1.17	1.245	1.32	V	

1. Typical values assume V_{DD} = 5.0 V, temperature = 25°C, f_{ADCK} = 1.0 MHz unless otherwise stated. Typical values are for reference only and are not tested in production.
2. Incremental I_{DD} added to MCU mode current.
3. Values are based on characterization results, not tested in production.
4. Reference the ADC module specification for more information on calculating conversion times.
5. Based on typical input pad leakage current.
6. LVI must be enabled, (LVIPWRD = 0, in CONFIG1). Voltage input to ADCH4:0 = $1A, an ADC conversion on this channel allows user to determine supply voltage.

NOTES:

1. CONTROLLING DIMENSION: MILLIMETER
2. DIMENSIONS AND TOLERANCES PER ANSI Y14.5M-1982.
3. DIMENSION DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS. MOLD FLASH OR GATE BURRS SHALL NOT EXCEED 0.15 PER SIDE.
4. DIMENSION DOES NOT INCLUDE INTERLEAD FLASH OR PROTRUSION. INTERLEAD FLASH OR PROTRUSION SHALL NOT EXCEED 0.25 PER SIDE.
5. DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 TOTAL IN EXCESS OF THE DIMENSION AT MAXIMUM MATERIAL CONDITION.
6. TERMINAL NUMBERS ARE SHOWN FOR REFERENCE ONLY.
7. DIMENSIONS ARE TO BE DETERMINED AT DATUM PLANE -W-.