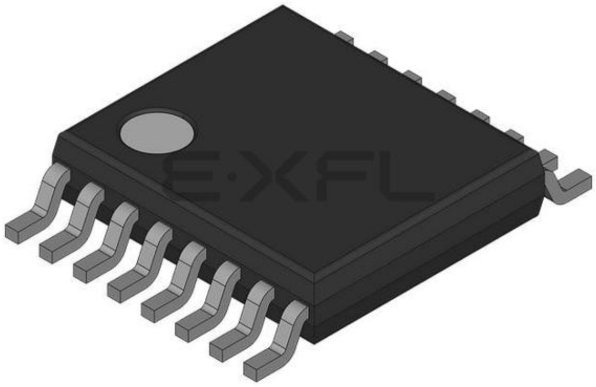
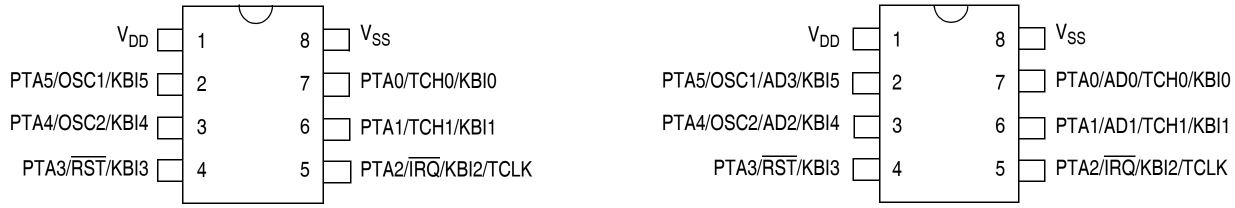
Freescale Semiconductor - MC908QY2CDTER Datasheet

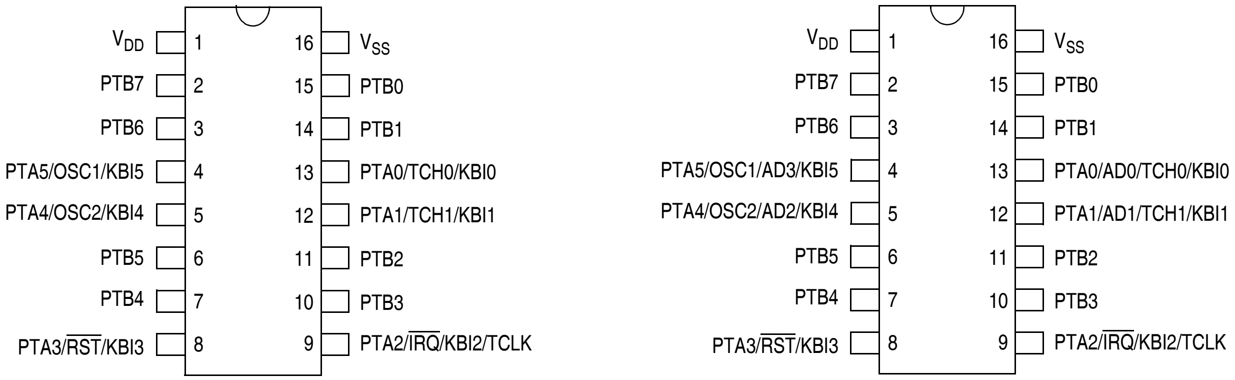


Details	
Product Status	Active
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LVD, POR, PWM
Number of I/O	13
Program Memory Size	1.5KB (1.5K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	16-TSSOP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908qy2cdter



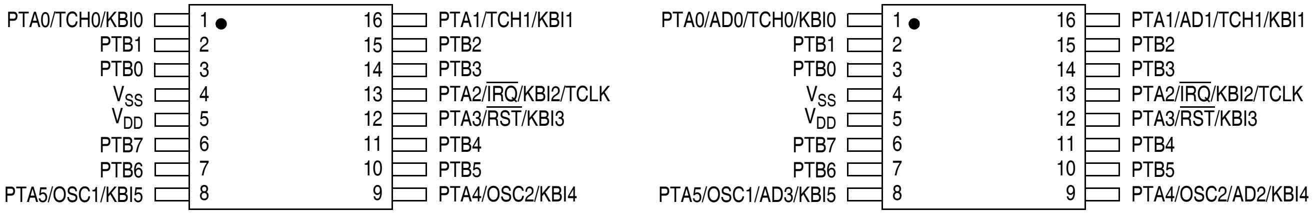
8-PIN ASSIGNMENT
MC68HC908QT1 PDIP/SOIC

8-PIN ASSIGNMENT
MC68HC908QT2 AND MC68HC908QT4 PDIP/SOIC



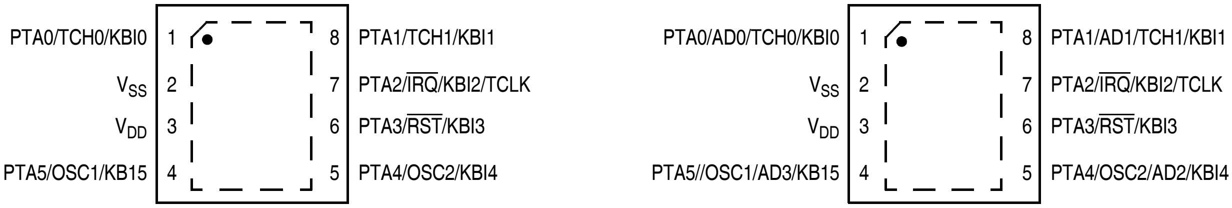
16-PIN ASSIGNMENT
MC68HC908QY1 PDIP/SOIC

16-PIN ASSIGNMENT
MC68HC908QY2 AND MC68HC908QY4 PDIP/SOIC



16-PIN ASSIGNMENT
MC68HC908QY1 TSSOP

16-PIN ASSIGNMENT
MC68HC908QY2 AND MC68HC908QY4 TSSOP



8-PIN ASSIGNMENT
MC68HC908QT1 DFN

8-PIN ASSIGNMENT
MC68HC908QT2 AND MC68HC908QT4 DFN

Figure 1-2. MCU Pin Assignments

2.6.3 FLASH Mass Erase Operation

Use the following procedure to erase the entire FLASH memory to read as a 1:

1. Set both the ERASE bit and the MASS bit in the FLASH control register.
2. Read the FLASH block protect register.
3. Write any data to any FLASH address[1] within the FLASH memory address range.
4. Wait for a time, t_{NVS} (minimum 10 μs).
5. Set the HVEN bit.
6. Wait for a time, t_{MErase} (minimum 4 ms).
7. Clear the ERASE and MASS bits.

NOTE

Mass erase is disabled whenever any block is protected (FLBPR does not equal $FF).

8. Wait for a time, t_{NVHL} (minimum 100 μs).
9. Clear the HVEN bit.
10. After time, t_{RCV} (typical 1 μs), the memory can be accessed in read mode again.

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 as shown, but other unrelated operations may occur between the steps.

CAUTION

A mass erase will erase the internal oscillator trim values at $FFC0 and $FFC1.

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

Only bytes which are currently $FF may be programmed.

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} (minimum 10 μs).
5. Set the HVEN bit.
6. Wait for a time, t_{PGS} (minimum 5 μs).
7. Write data to the FLASH address being programmed[2].

---

[1] When in monitor mode, with security sequence failed (see 15.3.2 Security), write to the FLASH block protect register instead of any FLASH address.

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

OSCOPT1 and OSCOPT0 — Selection Bits for Oscillator Option

(0, 0) Internal oscillator
(0, 1) External oscillator
(1, 0) External RC oscillator
(1, 1) External XTAL oscillator

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.

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3	SSREC	STOP	COPD
Reset:	0	0	0	0	U	0	0	0
POR:	0	0	0	0	0	0	0	0

U = Unaffected

Figure 5-2. Configuration Register 1 (CONFIG1)

COPRS (Out of STOP Mode) — COP Reset Period Selection Bit

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

COPRS (In STOP Mode) — Auto Wakeup Period Selection Bit

1 = Auto wakeup short cycle = 512 × INTRCOSC
0 = Auto wakeup long cycle = 16,384 × INTRCOSC

LVISTOP — LVI Enable in Stop Mode Bit

When the LVIPWRD bit is clear, setting the LVISTOP bit enables the LVI to operate during stop mode. Reset clears LVISTOP.

1 = LVI enabled during stop mode
0 = LVI disabled during stop mode

LVIRSTD — LVI Reset Disable Bit

LVIRSTD disables the reset signal from the LVI module.

1 = LVI module resets disabled
0 = LVI module resets enabled

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

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

Chapter 8
External Interrupt (IRQ)

8.1 Introduction

The $\overline{\text{IRQ}}$ pin (external interrupt), shared with PTA2 (general purpose input) and keyboard interrupt (KBI), provides a maskable interrupt input

8.2 Features

Features of the IRQ module include the following:

- External interrupt pin, $\overline{\text{IRQ}}$
- $\overline{\text{IRQ}}$ interrupt control bits
- Programmable edge-only or edge and level interrupt sensitivity
- Automatic interrupt acknowledge
- Selectable internal pullup resistor

8.3 Functional Description

$\overline{\text{IRQ}}$ pin functionality is enabled by setting configuration register 2 (CONFIG2) IRQEN bit accordingly. A zero disables the IRQ function and PTA2 will assume the other shared functionalities. A one enables the IRQ function.

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

8.4 Interrupts

The following IRQ source can generate interrupt requests:

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

8.5 Low-Power Modes

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

8.5.1 Wait Mode

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

8.5.2 Stop Mode

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

8.6 IRQ Module During Break Interrupts

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

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

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

8.7 I/O Signals

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

NOTE

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

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

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

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

8.8 Registers

The IRQ status and control register (INTSCR) controls and monitors operation of the IRQ module. See Chapter 5 Configuration Register (CONFIG).

The INTSCR has the following functions:

- Shows the state of the IRQ flag
- Clears the IRQ latch
- Masks the IRQ interrupt request
- Controls triggering sensitivity of the $\overline{\mathrm{IRQ}}$ interrupt pin

Address: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	IRQF	0	IMASK	MODE
Write:	▨	▨	▨	▨	▨	ACK		
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 8-3. IRQ Status and Control Register (INTSCR)

IRQF — IRQ Flag

This read-only status bit is set when the IRQ interrupt is pending.

1 = $\overline{\mathrm{IRQ}}$ interrupt pending
0 = $\overline{\mathrm{IRQ}}$ interrupt not pending

ACK — IRQ Interrupt Request Acknowledge Bit

Writing a 1 to this write-only bit clears the IRQ latch. ACK always reads as 0.

IMASK — IRQ Interrupt Mask Bit

Writing a 1 to this read/write bit disables the IRQ interrupt request.

1 = IRQ interrupt request disabled
0 = IRQ interrupt request enabled

MODE — IRQ Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the $\overline{\mathrm{IRQ}}$ pin.

1 = $\overline{\mathrm{IRQ}}$ interrupt request on falling edges and low levels
0 = $\overline{\mathrm{IRQ}}$ interrupt request on falling edges only

Chapter 9
Keyboard Interrupt Module (KBI)

9.1 Introduction

The keyboard interrupt module (KBI) provides six independently maskable external interrupts, which are accessible via the PTA0–PTA5 pins.

9.2 Features

Features of the keyboard interrupt module include:
- Six keyboard interrupt pins with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Software configurable pullup device if input pin is configured as input port bit
- Programmable edge-only or edge and level interrupt sensitivity
- Exit from low-power modes

9.3 Functional Description

The keyboard interrupt module controls the enabling/disabling of interrupt functions on the six port A pins. These six pins can be enabled/disabled independently of each other. Refer to Figure 9-2.

9.3.1 Keyboard Operation

Writing to the KBIE0–KBIE5 bits in the keyboard interrupt enable register (KBIER) independently enables or disables each port A pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin in port A also enables its internal pullup device irrespective of PTAPUEx bits in the port A input pullup enable register (see 12.2.3 Port A Input Pullup Enable Register). A logic 0 applied to an enabled keyboard interrupt pin latches a keyboard interrupt request.

A keyboard interrupt is latched when one or more keyboard interrupt inputs goes low after all were high. The MODEK bit in the keyboard status and control register controls the triggering mode of the keyboard interrupt.
- If the keyboard interrupt is edge-sensitive only, a falling edge on a keyboard interrupt input does not latch an interrupt request if another keyboard pin is already low. To prevent losing an interrupt request on one input because another input is still low, software can disable the latter input while it is low.
- If the keyboard interrupt is falling edge and low-level sensitive, an interrupt request is present as long as any keyboard interrupt input is low.

To determine the logic level on a keyboard interrupt pin, use the data direction register to configure the pin as an input and then read the data register.

> ***NOTE***
>
> *Setting a keyboard interrupt enable bit (KBIEx) forces the corresponding keyboard interrupt pin to be an input, overriding the data direction register. However, the data direction register bit must be a 0 for software to read the pin.*

9.3.2 Keyboard Initialization

When a keyboard interrupt pin is enabled, it takes time for the internal pullup to reach a logic 1. Therefore a false interrupt can occur as soon as the pin is enabled.

To prevent a false interrupt on keyboard initialization:

1. Mask keyboard interrupts by setting the IMASKK bit in the keyboard status and control register.
2. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.
3. Write to the ACKK bit in the keyboard status and control register to clear any false interrupts.
4. Clear the IMASKK bit.

An interrupt signal on an edge-triggered pin can be acknowledged immediately after enabling the pin. An interrupt signal on an edge- and level-triggered interrupt pin must be acknowledged after a delay that depends on the external load.

Another way to avoid a false interrupt:

1. Configure the keyboard pins as outputs by setting the appropriate DDRA bits in the data direction register A.
2. Write 1s to the appropriate port A data register bits.
3. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.

9.4 Wait Mode

The keyboard module remains active in wait mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of wait mode.

9.5 Stop Mode

The keyboard module remains active in stop mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of stop mode.

9.6 Keyboard Module During Break Interrupts

The system integration module (SIM) controls whether the keyboard interrupt latch 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.

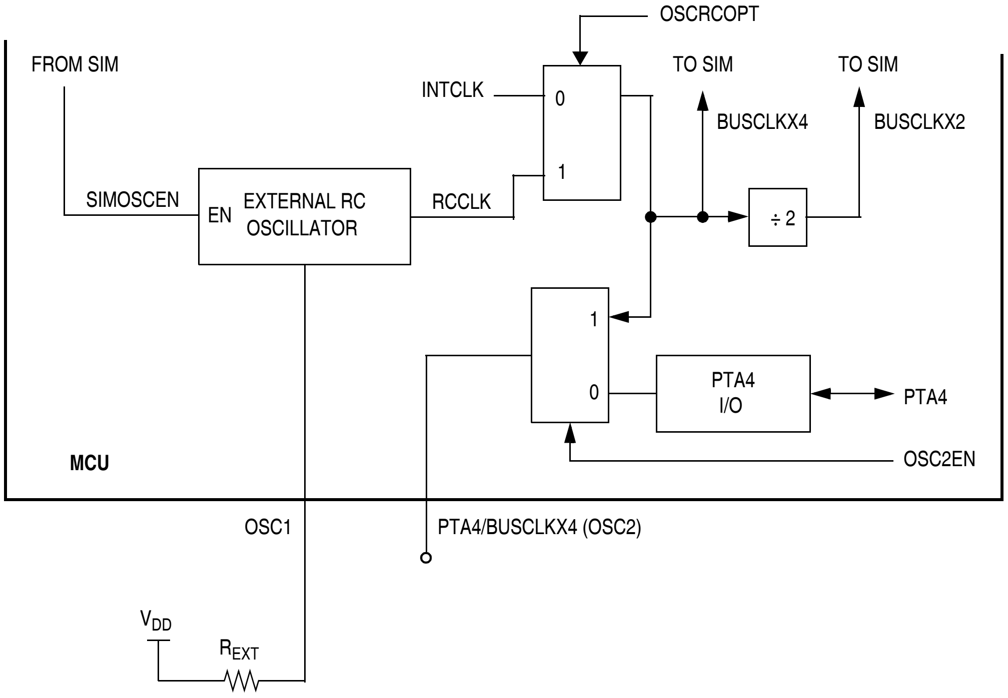
To allow software to clear the keyboard interrupt latch during a break interrupt, write a 1 to the BCFE bit. If a latch is cleared during the break state, it remains cleared when the MCU exits the break state.

11.3.4 RC Oscillator

The RC oscillator circuit is designed for use with an external resistor (R_{EXT}) to provide a clock source with a tolerance within 25% of the expected frequency. See Figure 11-3.

The capacitor (C) for the RC oscillator is internal to the MCU. The R_{EXT} value must have a tolerance of 1% or less to minimize its effect on the frequency.

In this configuration, the OSC2 pin can be left in the reset state as PTA4. Or, the OSC2EN bit in the port A pullup enable register can be set to enable the OSC2 output function on the pin. Enabling the OSC2 output slightly increases the external RC oscillator frequency, f_{RCCLK}.



See Chapter 16 Electrical Specifications for component value requirements.

Figure 11-3. RC Oscillator External Connections

11.4 Oscillator Module Signals

The following paragraphs describe the signals that are inputs to and outputs from the oscillator module.

11.4.1 Crystal Amplifier Input Pin (OSC1)

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

For the internal oscillator configuration, the OSC1 pin can assume other functions according to Table 1-3. Function Priority in Shared Pins.

11.4.2 Crystal Amplifier Output Pin (OSC2/PTA4/BUSCLKX4)

For the XTAL oscillator device**,** the OSC2 pin is the crystal oscillator inverting amplifier output.

For the external clock option, the OSC2 pin is dedicated to the PTA4 I/O function. The OSC2EN bit has no effect.

For the internal oscillator or RC oscillator options, the OSC2 pin can assume other functions according to Table 1-3. Function Priority in Shared Pins, or the output of the oscillator clock (BUSCLKX4).

Table 11-1. OSC2 Pin Function

Option	OSC2 Pin Function
XTAL oscillator	Inverting OSC1
External clock	PTA4 I/O
Internal oscillator or RC oscillator	Controlled by OSC2EN bit in PTAPUE register OSC2EN = 0: PTA4 I/O OSC2EN = 1: BUSCLKX4 output

11.4.3 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal comes from the system integration module (SIM) and enables/disables either the XTAL oscillator circuit, the RC oscillator, or the internal oscillator.

11.4.4 XTAL Oscillator Clock (XTALCLK)

XTALCLK is the XTAL oscillator output signal. It runs at the full speed of the crystal (f_{XCLK}) and comes directly from the crystal oscillator circuit. Figure 11-2 shows only the logical relation of XTALCLK to OSC1 and OSC2 and may not represent the actual circuitry. The duty cycle of XTALCLK is unknown and may depend on the crystal and other external factors. Also, the frequency and amplitude of XTALCLK can be unstable at start up.

11.4.5 RC Oscillator Clock (RCCLK)

RCCLK is the RC oscillator output signal. Its frequency is directly proportional to the time constant of external R and internal C. Figure 11-3 shows only the logical relation of RCCLK to OSC1 and may not represent the actual circuitry.

11.4.6 Internal Oscillator Clock (INTCLK)

INTCLK is the internal oscillator output signal. Its nominal frequency is fixed to 12.8 MHz, but it can be also trimmed using the oscillator trimming feature of the OSCTRIM register (see 11.3.1.1 Internal Oscillator Trimming).

11.4.7 Oscillator Out 2 (BUSCLKX4)

BUSCLKX4 is the same as the input clock (XTALCLK, RCCLK, or INTCLK). This signal is driven to the SIM module and is used to determine the COP cycles.

11.4.8 Oscillator Out (BUSCLKX2)

The frequency of this signal is equal to half of the BUSCLKX4, this signal is driven to the SIM for generation of the bus clocks used by the CPU and other modules on the MCU. BUSCLKX2 will be divided

11.8.1 Oscillator Status Register

The oscillator status register (OSCSTAT) contains the bits for switching from internal to external clock sources.

Address: $0036

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	R	R	R	R	R	ECGON	ECGST
Write:								▨
Reset:	0	0	0	0	0	0	0	0

R = Reserved ▨ = Unimplemented

Figure 11-4. Oscillator Status Register (OSCSTAT)

ECGON — External Clock Generator On Bit

This read/write bit enables external clock generator, so that the switching process can be initiated. This bit is forced low during reset. This bit is ignored in monitor mode with the internal oscillator bypassed, PTM or CTM mode.

1 = External clock generator enabled
0 = External clock generator disabled

ECGST — External Clock Status Bit

This read-only bit indicates whether or not an external clock source is engaged to drive the system clock.

1 = An external clock source engaged
0 = An external clock source disengaged

11.8.2 Oscillator Trim Register (OSCTRIM)

Address: $0038

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
Write:								
Reset:	1	0	0	0	0	0	0	0

Figure 11-5. Oscillator Trim Register (OSCTRIM)

TRIM7–TRIM0 — Internal Oscillator Trim Factor Bits

These read/write bits change the size of the internal capacitor used by the internal oscillator. By measuring the period of the internal clock and adjusting this factor accordingly, the frequency of the internal clock can be fine tuned. Increasing (decreasing) this factor by one increases (decreases) the period by approximately 0.2% of the untrimmed period (the period for TRIM = $80). The trimmed frequency is guaranteed not to vary by more than ±5% over the full specified range of temperature and voltage. The reset value is $80, which sets the frequency to 12.8 MHz (3.2 MHz bus speed) ±25%.

Applications using the internal oscillator should copy the internal oscillator trim value at location $FFC0 or $FFC1 into this register to trim the clock source.

12.2.1 Port A Data Register

The port A data register (PTA) contains a data latch for each of the six port A pins.

Address: $0000

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	AWUL	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Write:		▨				▨		
Reset:	Unaffected by reset							
Additional Functions:	▨	▨	KBI5	KBI4	KBI3	KBI2	KBI1	KBI0

R = Reserved ▨ = Unimplemented

Figure 12-1. Port A Data Register (PTA)

PTA[5:0] — Port A Data Bits

These read/write bits are software programmable. Data direction of each port A pin is under the control of the corresponding bit in data direction register A. Reset has no effect on port A data.

AWUL — Auto Wakeup Latch Data Bit

This is a read-only bit which has the value of the auto wakeup interrupt request latch. The wakeup request signal is generated internally (see Chapter 4 Auto Wakeup Module (AWU)). There is no PTA6 port nor any of the associated bits such as PTA6 data register, pullup enable or direction.

KBI[5:0] — Port A Keyboard Interrupts

The keyboard interrupt enable bits, KBIE5–KBIE0, in the keyboard interrupt control enable register (KBIER) enable the port A pins as external interrupt pins (see Chapter 9 Keyboard Interrupt Module (KBI)).

12.2.2 Data Direction Register A

Data direction register A (DDRA) determines whether each port A pin is an input or an output. Writing a 1 to a DDRA bit enables the output buffer for the corresponding port A pin; a 0 disables the output buffer.

Address: $0004

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	R	DDRA5	DDRA4	DDRA3	0	DDRA1	DDRA0
Write:						▨		
Reset:	0	0	0	0	0	0	0	0

R = Reserved ▨ = Unimplemented

Figure 12-2. Data Direction Register A (DDRA)

DDRA[5:0] — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA[5:0], configuring all port A pins as inputs.

1 = Corresponding port A pin configured as output
0 = Corresponding port A pin configured as input

NOTE

Avoid glitches on port A pins by writing to the port A data register before changing data direction register A bits from 0 to 1.

14.4.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in 14.4.4 Pulse Width Modulation (PWM). The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change a pulse width value could cause incorrect operation for up to two PWM 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 PWM period. Also, using a TIM overflow interrupt routine to write a new, smaller pulse width value may cause the compare to be missed. The TIM may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the PWM pulse width on channel x:

- When changing to a shorter pulse width, 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 pulse. The interrupt routine has until the end of the PWM period to write the new value.
- When changing to a longer pulse width, enable TIM overflow interrupts and write the new value in the TIM overflow interrupt routine. The TIM overflow interrupt occurs at the end of the current PWM 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 PWM period.

NOTE

In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare also can cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.

14.4.4.2 Buffered PWM Signal Generation

Channels 0 and 1 can be linked to form a buffered PWM channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The TIM channel 0 registers initially control the pulse width on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIM channel registers (0 or 1) that control the pulse width are the ones written to last. TSC0 controls and monitors the buffered PWM function, and TIM channel 1 status and control register (TSC1) is unused. While the MS0B bit is set, the channel 1 pin, TCH1, is available as a general-purpose I/O pin.

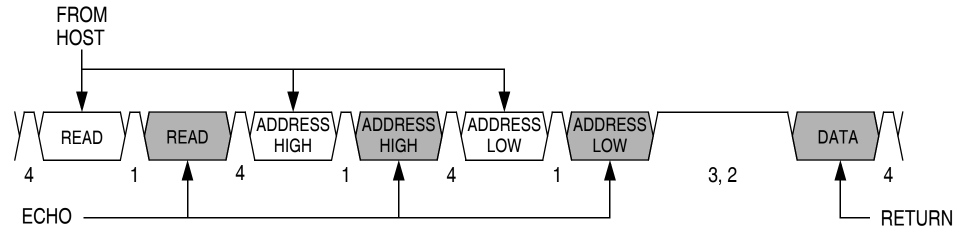
NOTE

In buffered PWM signal generation, do not write new pulse width values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered PWM signals.

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.

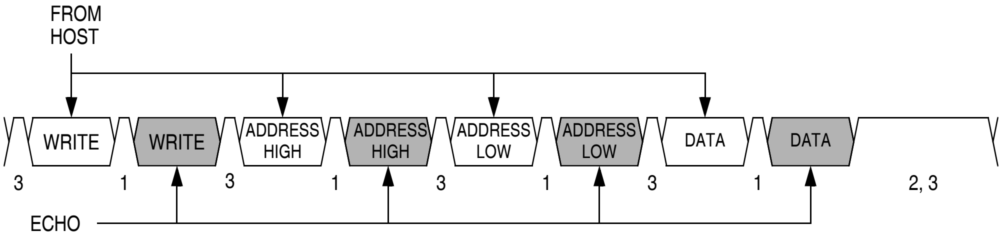
NOTE

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



Notes:
1 = Echo delay, 2 bit times
2 = Data return delay, 2 bit times
3 = Cancel command delay, 11 bit times
4 = Wait 1 bit time before sending next byte.

Figure 15-15. Read Transaction

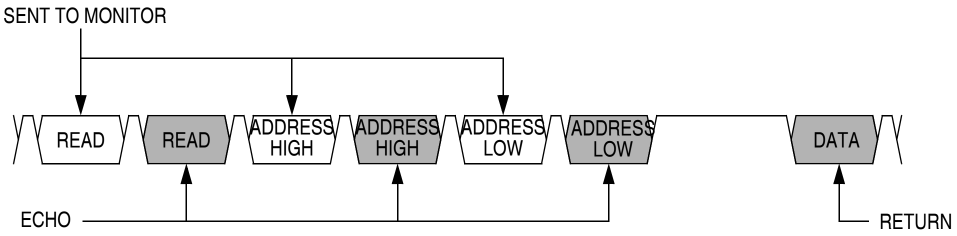


Notes:
1 = Echo delay, 2 bit times
2 = Cancel command delay, 11 bit times
3 = Wait 1 bit time before sending next byte.

Figure 15-16. Write Transaction

A brief description of each monitor mode command is given in Table 15-3 through Table 15-8.

Table 15-3. READ (Read Memory) Command

Description	Read byte from memory
Operand	2-byte address in high-byte:low-byte order
Data Returned	Returns contents of specified address
Opcode	$4A
Command Sequence 	

16.8 5-V Oscillator Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Internal oscillator frequency(1)	f_{INTCLK}	—	12.8	—	MHz
Deviation from trimmed Internal oscillator (2)(3) 12.8 MHz, fixed voltage, fixed temp 12.8 MHz, V_{DD} ± 10%, 0 to 70°C 12.8 MHz, V_{DD} ± 10%, –40 to 125°C	ACC_{INT}	 — — —	 ± 0.4 ± 2 —	 — — ± 5	%
Crystal frequency, XTALCLK(1)	$f_{OSCXCLK}$	1	—	24	MHz
External RC oscillator frequency, RCCLK(1)	f_{RCCLK}	2	—	12	MHz
External clock reference frequency(1) (4)	$f_{OSCXCLK}$	dc	—	32	MHz
Crystal load capacitance(5)	C_L	—	20	—	pF
Crystal fixed capacitance(3)	C_1	—	2 x C_L	—	—
Crystal tuning capacitance(3)	C_2	—	2 x C_L	—	—
Feedback bias resistor	R_B	0.5	1	10	MΩ
RC oscillator external resistor	R_{EXT}	See Figure 16-4			—
Crystal series damping resistor $f_{OSCXCLK}$ = 1 MHz $f_{OSCXCLK}$ = 4 MHz $f_{OSCXCLK}$ = > 8 MHz	R_S	 — — —	 20 10 0	 — — —	kΩ

1. Bus frequency, f_{OP}, is oscillator frequency divided by 4.
2. Deviation values assumes trimming @25•C and midpoint of voltage range.
3. Values are based on characterization results, not tested in production.
4. No more than 10% duty cycle deviation from 50%.
5. Consult crystal vendor data sheet.

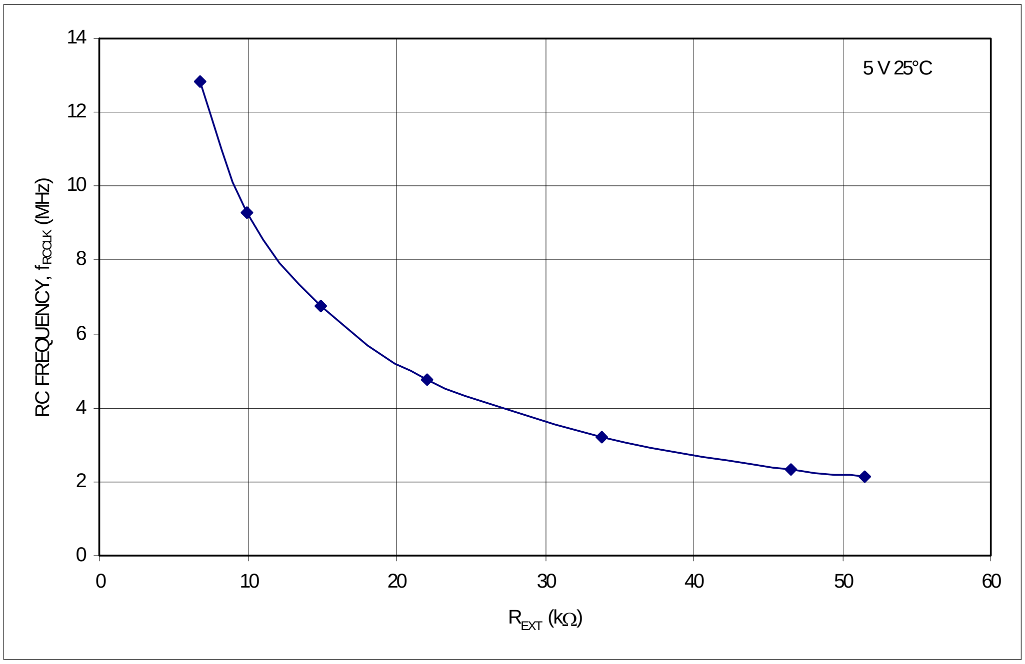
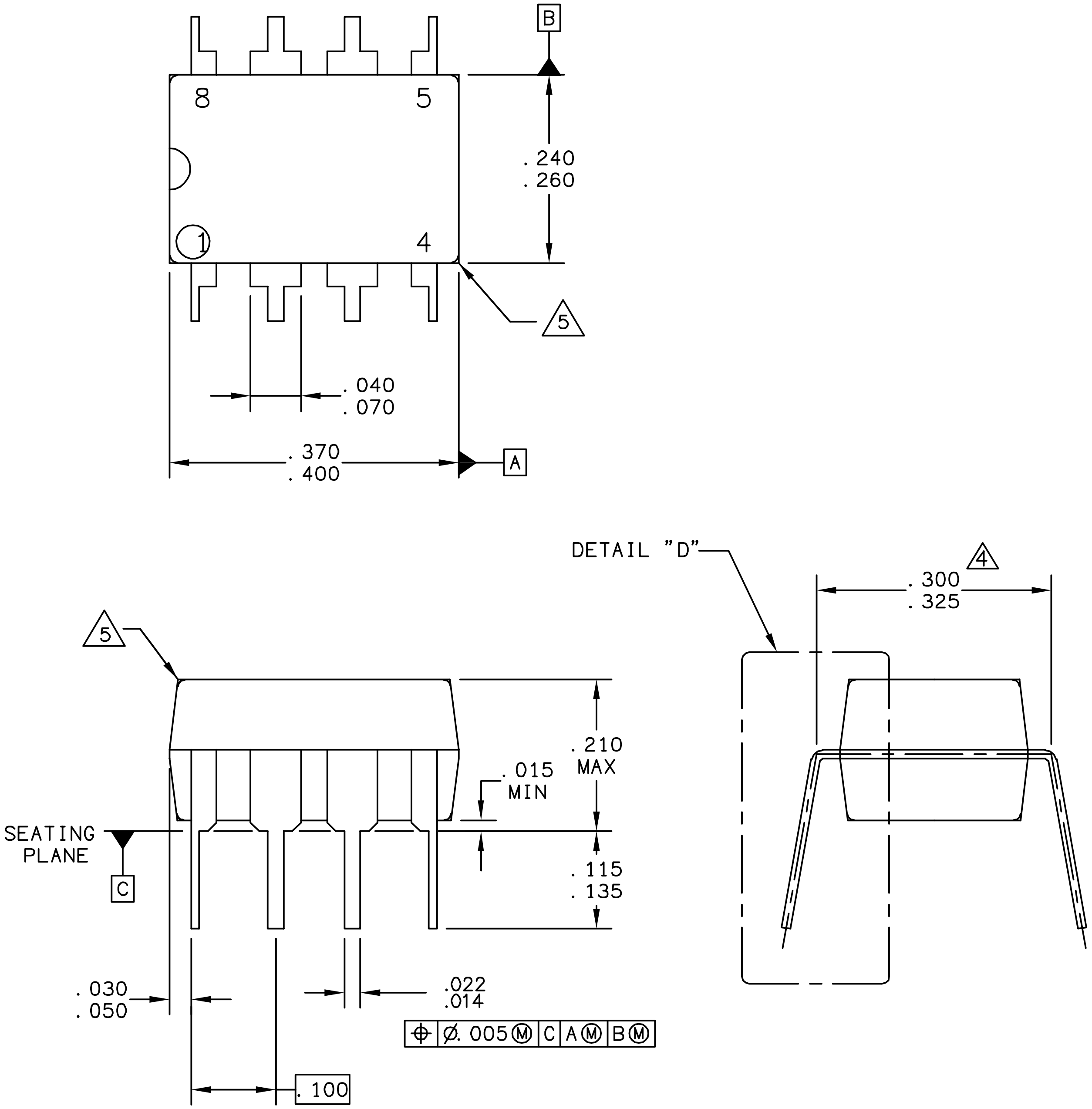


Figure 16-4. RC versus Frequency (5 Volts @ 25•C)



MECHANICAL OUTLINE

PRINT VERSION NOT TO SCALE

TITLE:

8 LD PDIP

DOCUMENT NO: 98ASB42420B

REV: N

CASE NUMBER: 626-06

19 MAY 2005

STANDARD: NON-JEDEC

freescale semiconductor

MECHANICAL OUTLINES DICTIONARY

DO NOT SCALE THIS DRAWING

DOCUMENT NO: 98ASB42567B

PAGE: 751G

REV: E

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.

TITLE:
16LD SOIC W/B, 1.27 PITCH,
CASE OUTLINE

CASE NUMBER: 751G–05

STANDARD: JEDEC MS–013AA

PACKAGE CODE: 2003

SHEET: 2 OF 3

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

freescale™
semiconductor