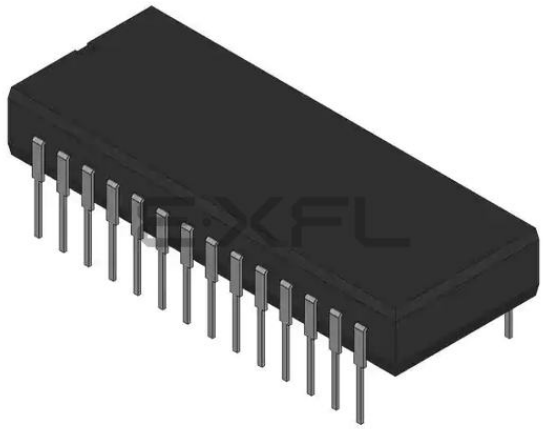


Welcome to E-XFL.COM

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
Peripherals	LED, LVD, POR, PWM
Number of I/O	23
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 13x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908jl8cdwe

  - 11 LED drivers (sink)
  - 2 × 25mA open-drain I/O with pull-up
- Resident routines for in-circuit programming and EEPROM emulation
- System protection features:
  - Optional computer operating properly (COP) reset, driven by internal RC oscillator
  - Optional low-voltage detection with reset and selectable trip points for 3V and 5V operation
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- Master reset pin with internal pull-up and power-on reset
- $\overline{\text{IRQ}}$ with schmitt-trigger input and programmable pull-up
- 20-pin dual in-line package (PDIP), 20-pin small outline integrated package (SOIC), 28-pin PDIP, 28-pin SOIC, 32-pin shrink dual in-line package (SDIP), and 32-pin low-profile quad flat pack (LQFP)
- Specific features of the MC68HC908JL8 in 28-pin packages are:
  - 23 general-purpose I/Os only
  - 7 keyboard interrupt with internal pull-up
  - 10 LED drivers (sink)
  - 12-channel ADC
  - Timer I/O pins on TIM1 only
- Specific features of the MC68HC908JL8 in 20-pin packages are:
  - 15 general-purpose I/Os only
  - 1 keyboard interrupt with internal pull-up
  - 4 LED drivers (sink)
  - 10-channel ADC
  - Timer I/O pins on TIM1 only

Features of the CPU08 include the following:

- Enhanced HC05 programming model
- Extensive loop control functions
- 16 addressing modes (eight more than the HC05)
- 16-bit index register and stack pointer
- Memory-to-memory data transfers
- Fast 8 × 8 multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Efficient C language support

1.3 MCU Block Diagram

Figure 1-1 shows the structure of the MC68HC908JL8.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA)	Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PTB)	Read: Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Reset:	Unaffected by reset							
$0002	Unimplemented	Read: Write:								
$0003	Port D Data Register (PTD)	Read: Write:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Reset:	Unaffected by reset							
$0004	Data Direction Register A (DDRA)	Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB)	Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Reset:	0	0	0	0	0	0	0	0
$0006	Unimplemented	Read: Write:								
$0007	Data Direction Register D (DDRD)	Read: Write:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Reset:	0	0	0	0	0	0	0	0
$0008	Port E Data Register (PTE)	Read: Write:							PTE1	PTE0
		Reset:	Unaffected by reset							
$0009	Unimplemented	Read: Write:								
$000A	Port D Control Register (PDCR)	Read: Write:	0	0	0	0	SLOWD7	SLOWD6	PTDPU7	PTDPU6
		Reset:	0	0	0	0	0	0	0	0
$000B	Unimplemented	Read: Write:								
$000C	Data Direction Register E (DDRE)	Read: Write:							DDRE1	DDRE0
		Reset:	0	0	0	0	0	0	0	0
$000D	Port A Input Pull-up Enable Register (PTAPUE)	Read: Write:	PTA6EN	PTAPUE6	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Reset:	0	0	0	0	0	0	0	0
$000E	PTA7 Input Pull-up Enable Register (PTA7PUE)	Read: Write:	PTAPUE7							
		Reset:	0	0	0	0	0	0	0	0
$000F ↓ $0012	Unimplemented	Read: Write:								

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 1 of 5)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0022	TIM1 Counter Register Low (T1CNTL)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0023	TIM Counter Modulo Register High (TMODH)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0024	TIM1 Counter Modulo Register Low (T1MODL)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0025	TIM1 Channel 0 Status and Control Register (T1SC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0026	TIM1 Channel 0 Register High (T1CH0H)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	Indeterminate after reset							
$0027	TIM1 Channel 0 Register Low (T1CH0L)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	Indeterminate after reset							
$0028	TIM1 Channel 1 Status and Control Register (T1SC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0029	TIM1 Channel 1 Register High (T1CH1H)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	Indeterminate after reset							
$002A	TIM1 Channel 1 Register Low (T1CH1L)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	Indeterminate after reset							
$002B ↓ $002F	Unimplemented	Read:								
		Write:								
$0030	TIM2 Status and Control Register (T2SC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0031	TIM2 Counter Register High (T2CNTH)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0032	TIM2 Counter Register Low (T2CNTL)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0033	TIM2 Counter Modulo Register High (T2MODH)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0034	TIM2 Counter Modulo Register Low (T2MODL)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	1	1	1	1	1	1	1	1

U = Unaffected X = Indeterminate = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 3 of 5)

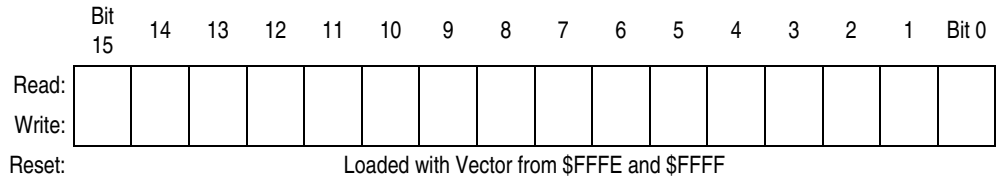


Figure 4-5. Program Counter (PC)

4.3.5 Condition Code Register

The 8-bit condition code register contains the interrupt mask and five flags that indicate the results of the instruction just executed. Bits 6 and 5 are set permanently to logic 1. The following paragraphs describe the functions of the condition code register.

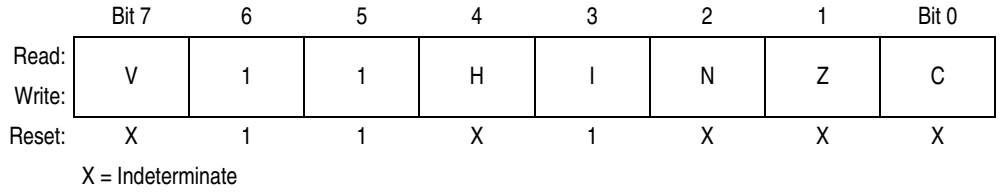


Figure 4-6. Condition Code Register (CCR)

V — Overflow Flag

The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag.

1 = Overflow
0 = No overflow

H — Half-Carry Flag

The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C flags to determine the appropriate correction factor.

1 = Carry between bits 3 and 4
0 = No carry between bits 3 and 4

I — Interrupt Mask

When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the interrupt vector is fetched.

1 = Interrupts disabled
0 = Interrupts enabled

NOTE

To maintain M6805 Family compatibility, the upper byte of the index register (H) is not stacked automatically. If the interrupt service routine modifies H, then the user must stack and unstack H using the PSHH and PULH instructions.

Table 4-1. Instruction Set Summary

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
DBNZ *opr,rel* DBNZA *rel* DBNZX *rel* DBNZ *opr*,X,*rel* DBNZ X,*rel* DBNZ *opr*,SP,*rel*	Decrement and Branch if Not Zero	A ← (A)–1 or M ← (M)–1 or X ← (X)–1 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 4 + *rel* ? (result) ≠ 0	–	–	–	–	–	–	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	5 3 3 5 4 6
DEC *opr* DECA DECX DEC *opr*,X DEC ,X DEC *opr*,SP	Decrement	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1 M ← (M) – 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff	4 1 1 4 3 5
DIV	Divide	A ← (H:A)/(X) H ← Remainder	–	–	–	–	↕	↕	INH	52		7
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X EOR *opr*,SP EOR *opr*,SP	Exclusive OR M with A	A ← (A ⊕ M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A8 B8 C8 D8 E8 F8 9EE8 9ED8	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
INC *opr* INCA INCX INC *opr*,X INC ,X INC *opr*,SP	Increment	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1 M ← (M) + 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff	4 1 1 4 3 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff 	2 3 4 3 2
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + *n* (*n* = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Unconditional Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff 	4 5 6 5 4
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X LDA *opr*,SP LDA *opr*,SP	Load A from M	A ← (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A6 B6 C6 D6 E6 F6 9EE6 9ED6	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LDHX #*opr* LDHX *opr*	Load H:X from M	H:X ← (M:M + 1)	0	–	–	↕	↕	–	IMM DIR	45 55	ii jj dd	3 4

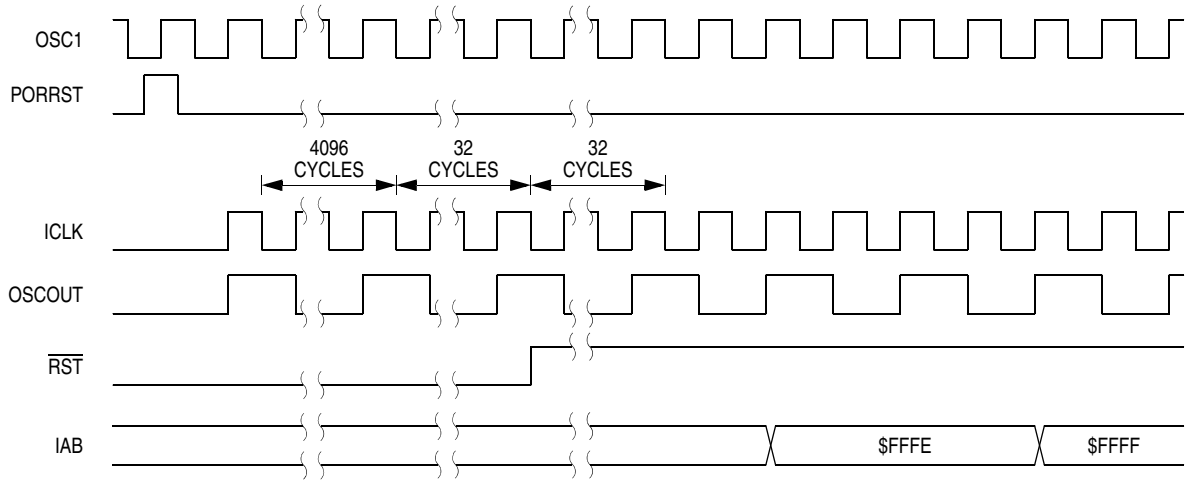


Figure 5-7. POR Recovery

5.3.2.2 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 reset status register (RSR). The SIM actively pulls down the $\overline{\text{RST}}$ 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 through 5 of the SIM counter. The SIM counter output, which occurs at least every ($2^{12} - 2^4$) ICLK 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 if the $\overline{\text{RST}}$ pin or the $\overline{\text{IRQ}}$ pin is held at V_{TST} while the MCU is in monitor mode. The COP module can be disabled only through combinational logic conditioned with the high voltage signal on the $\overline{\text{RST}}$ or the $\overline{\text{IRQ}}$ pin. This prevents the COP from becoming disabled as a result of external noise. During a break state, V_{TST} on the $\overline{\text{RST}}$ pin disables the COP module.

5.3.2.3 Illegal Opcode Reset

The SIM decodes signals from the CPU to detect illegal instructions. An illegal instruction sets the ILOP bit in the reset status register (RSR) and causes a reset.

If the stop enable bit, STOP, in the mask option register is logic zero, the SIM treats the STOP instruction as an illegal opcode and causes an illegal opcode reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

5.3.2.4 Illegal Address Reset

An opcode fetch from an unmapped address generates an illegal address reset. The SIM verifies that the CPU is fetching an opcode prior to asserting the ILAD bit in the reset status register (RSR) and resetting the MCU. A data fetch from an unmapped address does not generate a reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

5.3.2.5 Low-Voltage Inhibit (LVI) Reset

The low-voltage inhibit module (LVI) asserts its output to the SIM when the V_{DD} voltage falls to the LVI trip voltage V_{TRIP}. The LVI bit in the reset status register (RSR) is set, and the external reset pin ($\overline{\text{RST}}$) is

Table 7-4. READ (Read Memory) Command

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

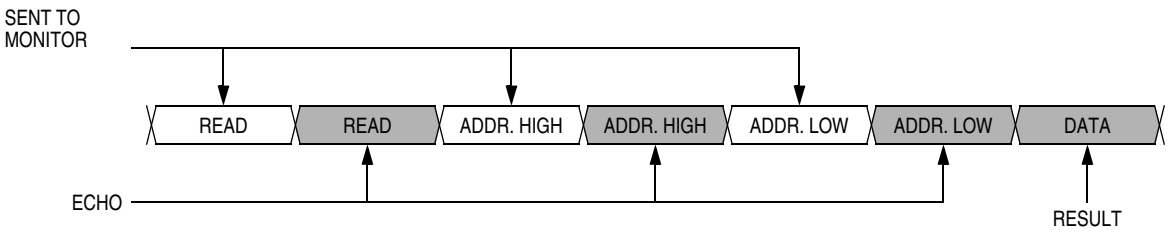
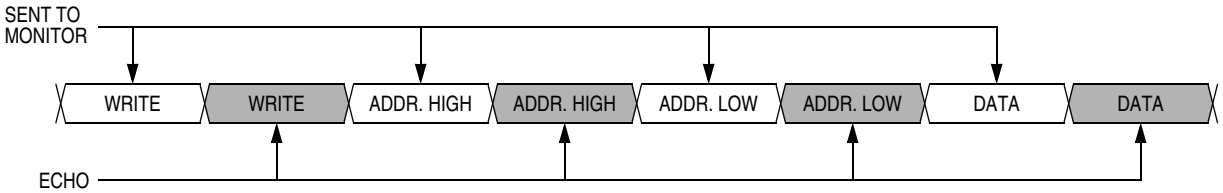


Table 7-5. WRITE (Write Memory) Command

Description	Write byte to memory
Operand	Specifies 2-byte address in high byte:low byte order; low byte followed by data byte
Data Returned	None
Opcode	$49
Command Sequence	



cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare can also cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.

5. In the TIM status control register (TSC), clear the TIM stop bit, TSTOP.

Setting MS0B links channels 0 and 1 and configures them for buffered PWM operation. The TIM channel 0 registers (TCH0H:TCH0L) initially control the buffered PWM output. TIM status control register 0 (TSCR0) controls and monitors the PWM signal from the linked channels.

Clearing the toggle-on-overflow bit, TOVx, inhibits output toggles on TIM overflows. Subsequent output compares try to force the output to a state it is already in and have no effect. The result is a 0% duty cycle output.

Setting the channel x maximum duty cycle bit (CHxMAX) and setting the TOVx bit generates a 100% duty cycle output. (See 8.9.4 TIM Channel Status and Control Registers.)

8.5 Interrupts

The following TIM sources can generate interrupt requests:

- TIM overflow flag (TOF) — The TOF bit is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. The TIM overflow interrupt enable bit, TOIE, enables TIM overflow CPU interrupt requests. TOF and TOIE are in the TIM status and control register.
- TIM channel flags (CH1F:CH0F) — The CHxF bit is set when an input capture or output compare occurs on channel x. Channel x TIM CPU interrupt requests are controlled by the channel x interrupt enable bit, CHxIE. Channel x TIM CPU interrupt requests are enabled when CHxIE = 1. CHxF and CHxIE are in the TIM channel x status and control register.

8.6 Low-Power Modes

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

8.6.1 Wait Mode

The TIM remains active after the execution of a WAIT instruction. In wait mode, the TIM registers are not accessible by the CPU. Any enabled CPU interrupt request from the TIM can bring the MCU out of wait mode.

If TIM functions are not required during wait mode, reduce power consumption by stopping the TIM before executing the WAIT instruction.

8.6.2 Stop Mode

The TIM is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions or the state of the TIM counter. TIM operation resumes when the MCU exits stop mode after an external interrupt.

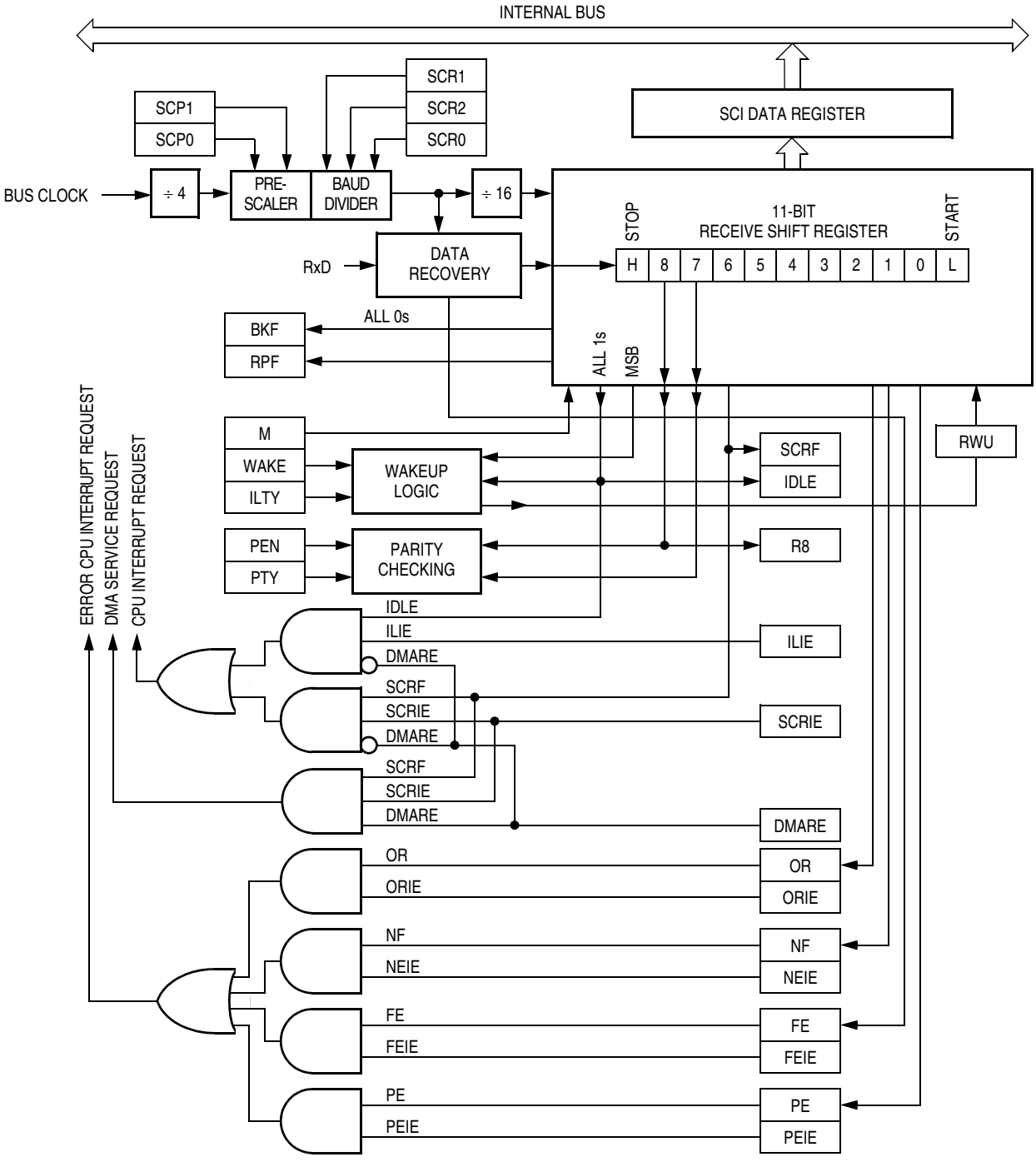


Figure 9-5. SCI Receiver Block Diagram

Fast Data Tolerance

Figure 9-8 shows how much a fast received character can be misaligned without causing a noise error or a framing error. The fast stop bit ends at RT10 instead of RT16 but is still there for the stop bit data samples at RT8, RT9, and RT10.

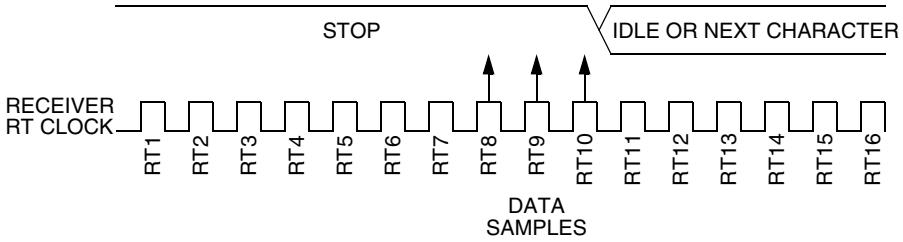


Figure 9-8. Fast Data

For an 8-bit character, data sampling of the stop bit takes the receiver
9 bit times × 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in Figure 9-8, the receiver counts 154 RT cycles at the point when the count of the transmitting device is
10 bit times × 16 RT cycles = 160 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is

$$\left|\frac{154 - 160}{154}\right| \times 100 = 3.90\%$$

For a 9-bit character, data sampling of the stop bit takes the receiver
10 bit times × 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in Figure 9-8, the receiver counts 170 RT cycles at the point when the count of the transmitting device is
11 bit times × 16 RT cycles = 176 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is

$$\left|\frac{170 - 176}{170}\right| \times 100 = 3.53\%$$

9.4.3.6 Receiver Wakeup

So that the MCU can ignore transmissions intended only for other receivers in multiple-receiver systems, the receiver can be put into a standby state. Setting the receiver wakeup bit, RWU, in SCC2 puts the receiver into a standby state during which receiver interrupts are disabled.

Depending on the state of the WAKE bit in SCC1, either of two conditions on the RxD pin can bring the receiver out of the standby state:

9.8.7 SCI Baud Rate Register

The baud rate register (SCBR) selects the baud rate for both the receiver and the transmitter.

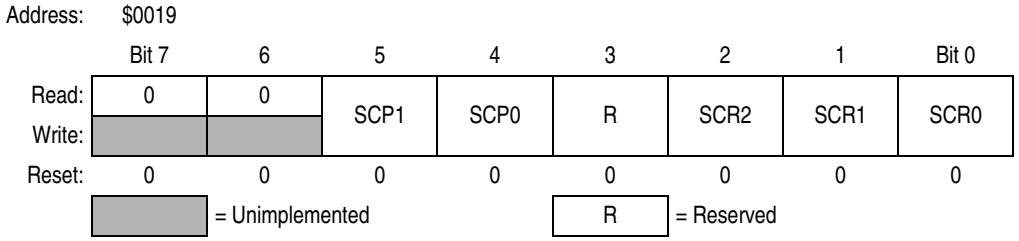


Figure 9-16. SCI Baud Rate Register (SCBR)

SCP1 and SCP0 — SCI Baud Rate Prescaler Bits

These read/write bits select the baud rate prescaler divisor as shown in Table 9-6. Reset clears SCP1 and SCP0.

Table 9-6. SCI Baud Rate Prescaling

SCP1 and SCP0	Prescaler Divisor (PD)
00	1
01	3
10	4
11	13

SCR2–SCR0 — SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate divisor as shown in Table 9-7. Reset clears SCR2–SCR0.

Table 9-7. SCI Baud Rate Selection

SCR2, SCR1, and SCR0	Baud Rate Divisor (BD)
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

Use this formula to calculate the SCI baud rate:

$$\text{baud rate} = \frac{\text{SCI clock source}}{64 \times \text{PD} \times \text{BD}}$$

where:

SCI clock source = bus clock
PD = prescaler divisor
BD = baud rate divisor

Table 9-8 shows the SCI baud rates that can be generated with a 4.9152MHz bus clock.

Address:	$0007							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Reset:	0	0	0	0	0	0	0	0

Figure 11-11. Data Direction Register D (DDRD)

DDRD[7:0] — Data Direction Register D Bits

These read/write bits control port D data direction. Reset clears DDRD[7:0], configuring all port D pins as inputs.

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

NOTE

Avoid glitches on port D pins by writing to the port D data register before changing data direction register D bits from 0 to 1. Figure 11-12 shows the port D I/O logic.

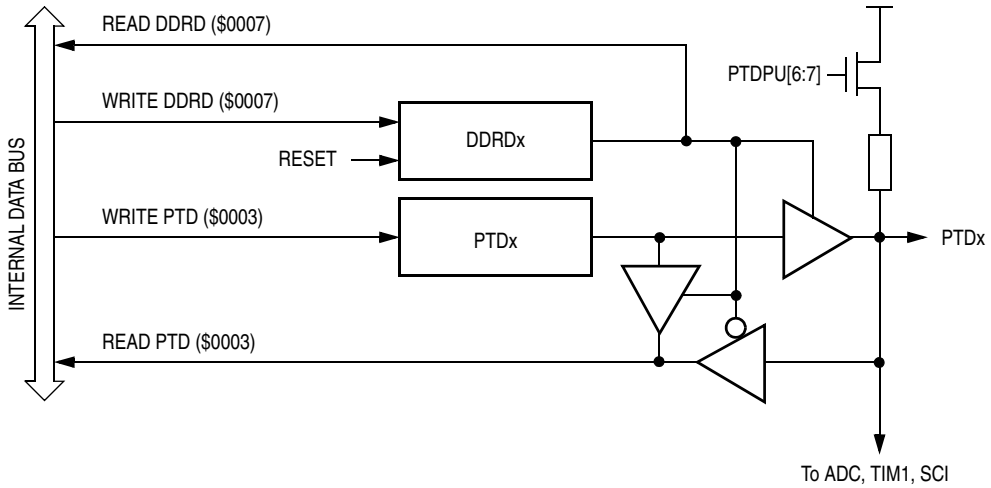


Figure 11-12. Port D I/O Circuit

When DDRDx is a logic 1, reading address $0003 reads the PTDx data latch. When DDRDx is a logic 0, reading address $0003 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 11-4 summarizes the operation of the port D pins.

Table 11-4. Port D Pin Functions

DDRD Bit	PTD Bit	I/O Pin Mode	Accesses to DDRD	Accesses to PTD	
			Read/Write	Read	Write
0	X(1)	Input, Hi-Z(2)	DDRD[7:0]	Pin	PTD[7:0](3)
1	X	Output	DDRD[7:0]	PTD[7:0]	PTD[7:0]

1. X = don't care.
2. Hi-Z = high impedance.
3. Writing affects data register, but does not affect the input.

Chapter 13
Keyboard Interrupt Module (KBI)

13.1 Introduction

The keyboard interrupt module (KBI) provides eight independently maskable external interrupts which are accessible via PTA0–PTA7. When a port pin is enabled for keyboard interrupt function, an internal pull-up device is also enabled on the pin.

13.2 Features

Features of the keyboard interrupt module include the following:

- Eight keyboard interrupt pins with pull-up devices
- Separate keyboard interrupt enable bits and one keyboard interrupt mask
- Programmable edge-only or edge- and level- interrupt sensitivity
- Exit from low-power modes

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001A	Keyboard Status and Control Register (KBSCR)	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$001B	Keyboard Interrupt Enable Register (KBIER)	Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 13-1. KBI I/O Register Summary

13.3 I/O Pins

The eight keyboard interrupt pins are shared with standard port I/O pins. The full name of the KBI pins are listed in Table 13-1. The generic pin name appear in the text that follows.

Table 13-1. Pin Name Conventions

KBI Generic Pin Name	Full MCU Pin Name	Pin Selected for KBI Function by KBIEx Bit in KBIER
KBI0–KBI5	PTA0/KBI0–PTA5/KBI5	KBIE0–KBIE5
KBI6	OSC2/RCCLK/PTA6/KBI6(1)	KBIE6
KBI7	PTA7/KBI7	KBIE7

1. PTA6/KBI6 is only available when OSCSEL=0 at $FFD0 (RC option), and PTA6EN=1 at $000D.

13.4 Functional Description

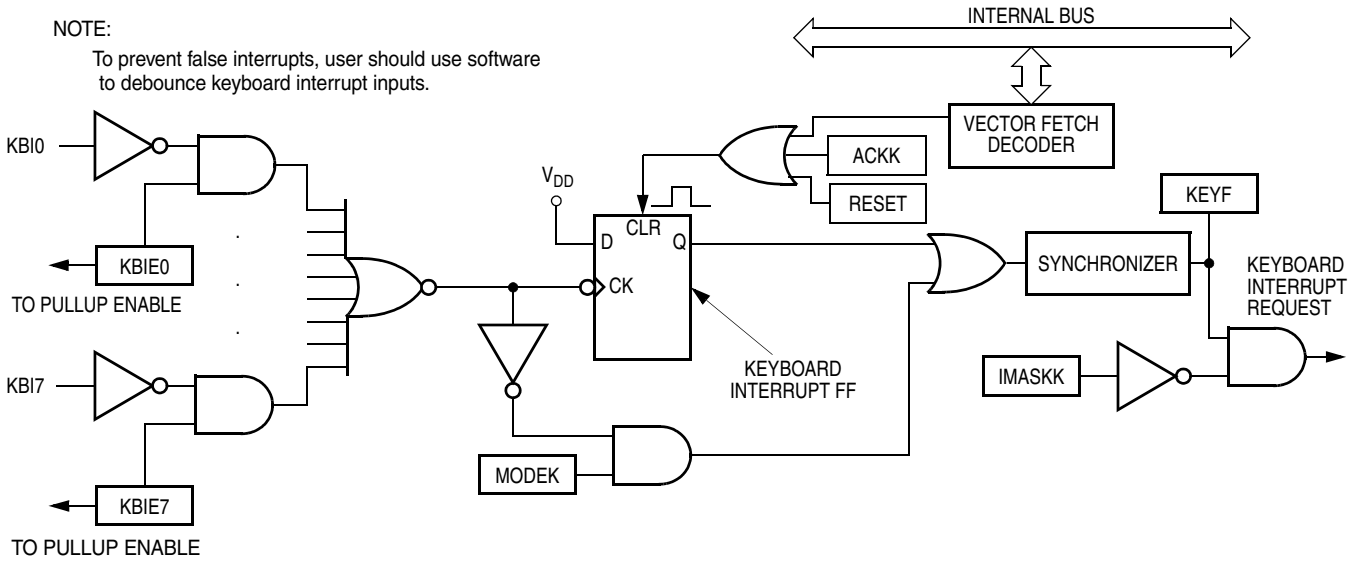


Figure 13-2. Keyboard Interrupt Block Diagram

Writing to the KBIE7–KBIE0 bits in the keyboard interrupt enable register 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 pull-up device regardless of PTAPUEx bits in the port A input pull-up enable register (see 11.2.3 Port A Input Pull-Up Enable Registers). 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 pins 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 pin does not latch an interrupt request if another keyboard pin is already low. To prevent losing an interrupt request on one pin because another pin is still low, software can disable the latter pin 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 pin is low.

If the MODEK bit is set, the keyboard interrupt pins are both falling edge- and low level-sensitive, and both of the following actions must occur to clear a keyboard interrupt request:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the interrupt request. Software may generate the interrupt acknowledge signal by writing a logic 1 to the ACKK bit in the keyboard status and control register KBSCR. The ACKK bit is useful in applications that poll the keyboard interrupt pins and require software to clear the keyboard interrupt request. Writing to the ACKK bit 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 pins. A falling edge that occurs after writing to the ACKK bit latches another interrupt request. If the keyboard interrupt mask bit, IMASKK, is clear, the CPU loads the program counter with the vector address at locations $FFE0 and $FFE1.
- Return of all enabled keyboard interrupt pins to logic 1 — As long as any enabled keyboard interrupt pin is at logic 0, the keyboard interrupt remains set.

The vector fetch or software clear and the return of all enabled keyboard interrupt pins to logic 1 may occur in any order.

Address: $001A

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

= Unimplemented

Figure 13-3. Keyboard Status and Control Register (KBSCR)

KEYF — Keyboard Flag Bit

This read-only bit is set when a keyboard interrupt is pending on port A. Reset clears the KEYF bit.

1 = Keyboard interrupt pending
0 = No keyboard interrupt pending

ACKK — Keyboard Acknowledge Bit

Writing a logic 1 to this write-only bit clears the keyboard interrupt request on port A. ACKK always reads as logic 0. Reset clears ACKK.

IMASKK— Keyboard Interrupt Mask Bit

Writing a logic 1 to this read/write bit prevents the output of the keyboard interrupt mask from generating interrupt requests on port A. Reset clears the IMASKK bit.

1 = Keyboard interrupt requests masked
0 = Keyboard interrupt requests not masked

MODEK — Keyboard Triggering Sensitivity Bit

This read/write bit controls the triggering sensitivity of the keyboard interrupt pins on port A. Reset clears MODEK.

1 = Keyboard interrupt requests on falling edges and low levels
0 = Keyboard interrupt requests on falling edges only

13.5.2 Keyboard Interrupt Enable Register

The port-A keyboard interrupt enable register enables or disables each port-A pin to operate as a keyboard interrupt pin

Address: $001B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 13-4. Keyboard Interrupt Enable Register (KBIER)

KBIE7–KBIE0 — Port-A Keyboard Interrupt Enable Bits

Each of these read/write bits enables the corresponding keyboard interrupt pin on port-A to latch interrupt requests. Reset clears the keyboard interrupt enable register.

1 = KBIx pin enabled as keyboard interrupt pin
0 = KBIx pin not enabled as keyboard interrupt pin

17.5 5V DC Electrical Characteristics

Table 17-4. DC Electrical Characteristics (5V)

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage (I_{LOAD} = –2.0mA) PTA0–PTA7, PTB0–PTB7, PTD0–PTD7, PTE0–PTE1	V_{OH}	V_{DD}–0.8	—	—	V
Output low voltage (I_{LOAD} = 1.6mA) PTA6, PTB0–PTB7, PTD0, PTD1, PTD4, PTD5, PTE0–PTE1	V_{OL}	—	—	0.4	V
Output low voltage (I_{LOAD} = 25mA) PTD6, PTD7	V_{OL}	—	—	0.5	V
LED drives (V_{OL} = 3V) PTA0–PTA5, PTA7, PTD2, PTD3, PTD6, PTD7	I_{OL}	10	16	25	mA
Input high voltage PTA0–PTA7, PTB0–PTB7, PTD0–PTD7, PTE0–PTE1, $\overline{RST}$, $\overline{IRQ}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage PTA0–PTA7, PTB0–PTB7, PTD0–PTD7, PTE0–PTE1, $\overline{RST}$, $\overline{IRQ}$, OSC1	V_{IL}	V_{SS}	—	0.3 × V_{DD}	V
V_{DD} supply current, f_{OP} = 8MHz Run[3] XTAL oscillator option RC oscillator option Wait[4] XTAL oscillator option RC oscillator option Stop[5] (–40°C to 125°C) XTAL oscillator option RC oscillator option	I_{DD}	 — — — — — —	 7.5 11 3 3.5 1.5 0.5	 10 13 5.5 6 8 3	 mA mA mA mA µA µA
Digital I/O ports Hi-Z leakage current	I_{IL}	—	—	± 10	µA
Input current	I_{IN}	—	—	± 1	µA
Capacitance Ports (as input or output)	C_{OUT} C_{IN}	— —	— —	12 8	pF
POR rearm voltage[6]	V_{POR}	0	—	100	mV
POR rise time ramp rate[7]	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{TST}	1.5 × V_{DD}	—	8.5	V
Pullup resistors[8] PTD6, PTD7 $\overline{RST}$, $\overline{IRQ}$, PTA0–PTA7	 R_{PU1} R_{PU2}	 1.8 16	 3.3 26	 4.8 36	 kΩ kΩ
Low-voltage inhibit, trip falling voltage	V_{TRIPF}	3.60	4.25	4.48	V
Low-voltage inhibit, trip rising voltage	V_{TRIPR}	3.75	4.40	4.63	V

1. V_{DD} = 4.5 to 5.5 Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
3. Run (operating) I_{DD} measured using external square wave clock source (f_{OP} = 8MHz). All inputs 0.2V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD}. Measured with all modules enabled.
4. Wait I_{DD} measured using external square wave clock source (f_{OP} = 8MHz). All inputs 0.2V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects wait I_{DD}.
5. Stop I_{DD} measured with OSC1 grounded; no port pins sourcing current. LVI is disabled.
6. Maximum is highest voltage that POR is guaranteed.
7. If minimum V_{DD} is not reached before the internal POR reset is released, $\overline{RST}$ must be driven low externally until minimum V_{DD} is reached.
8. R_{PU1} and R_{PU2} are measured at V_{DD} = 5.0V.

17.10 3V Oscillator Characteristics

Table 17-9. Oscillator Specifications (3V)

Characteristic	Symbol	Min	Typ	Max	Unit
Internal oscillator clock frequency	f_{ICLK}		45k[1]		Hz
External reference clock to OSC1 [2]	f_{OSC}	dc	—	16M	Hz
Crystal reference frequency [3]	$f_{XTALCLK}$		—	16M	Hz
Crystal load capacitance [4]	C_L	—	—	—	
Crystal fixed capacitance [3]	C_1	—	$2 \times C_L$	—	
Crystal tuning capacitance [3]	C_2	—	$2 \times C_L$	—	
Feedback bias resistor	R_B	—	10 MΩ	—	
Series resistor [3], [5]	R_S	—	—	—	
External RC clock frequency	f_{RCCLK}	2M	—	10M	Hz
RC oscillator external R	R_{EXT}	See Figure 17-4			Ω
RC oscillator external C	C_{EXT}	—	10	—	pF

1. Typical value reflect average measurements at midpoint of voltage range, 25 °C only. See Figure 17-5 for plot.
2. No more than 10% duty cycle deviation from 50%.
3. Fundamental mode crystals only.
4. Consult crystal vendor data sheet.
5. Not required for high frequency crystals.

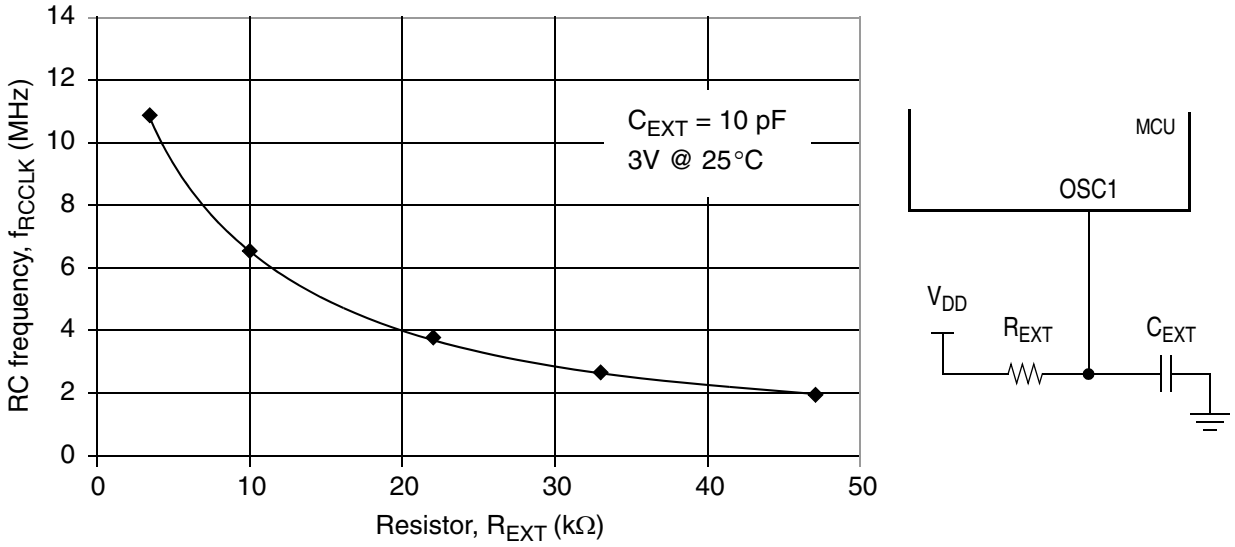


Figure 17-4. RC vs. Frequency (3V @25°C)

Appendix B
MC68HC908KL8

B.1 Introduction

This appendix introduces the MC68HC908KL8, an ADC-less device of the MC68HC908JL8. The entire data book applies to this device, with exceptions outlined in this appendix.

Table B-1. Summary of MC68HC908KL8 and MC68HC908JL8 Differences

	MC68HC908KL8	MC68HC908JL8
Analog-to-Digital Converter (ADC)	—	13-channel, 8-bit.
Registers at: $003C, $003E, and $003E	Not used; locations are reserved.	ADC registers.
Interrupt Vector at: $FFDE and $FFDF	Not used.	ADC interrupt vector.
Available Packages	— — 28-pin PDIP 28-pin SOIC 32-pin SDIP —	20-pin PDIP (MC68HC908JK8) 20-pin SOIC (MC68HC908JK8) 28-pin PDIP 28-pin SOIC 32-pin SDIP 32-pin LQFP

B.2 MCU Block Diagram

Figure B-1 shows the block diagram of the MC68HC908KL8.

B.3 Pin Assignments

Figure B-2 and Figure B-3 show the pin assignments for the MC68HC908KL8.

B.4 Reserved Registers

The following registers are reserved location on the MC68HC908KL8.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$003C	Reserved	Read: Write: Reset:	R	R	R	R	R	R	R	R
$003D	Reserved	Read: Write: Reset:	R	R	R	R	R	R	R	R
$003E	Reserved	Read: Write: Reset:	R	R	R	R	R	R	R	R

Figure B-4. Reserved Registers

B.5 Reserved Vectors

The following are reserved interrupt vectors on the MC68HC908KL8.

Table B-2. Reserved Vectors

Vector Priority	INT Flag	Address	Vector
—	IF15	$FFDE	Reserved
		$FFDF	Reserved

B.6 MC68HC908KL8 Order Numbers

Table B-3. MC68HC908KL8 Order Numbers

MC Order Number	Operating Temperature Range	Package
MC68HC908KL8CP	–40 °C to +85 °C	28-pin PDIP
MC68HC908KL8CDW	–40 °C to +85 °C	28-pin SOIC
MC68HC908KL8CSP	–40 °C to +85 °C	32-pin SDIP