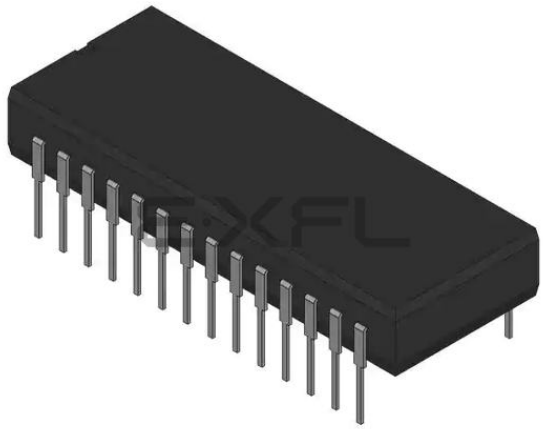


Welcome to E-XFL.COM

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	Active
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LED, LVD, POR, PWM
Number of I/O	23
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.3V
Data Converters	A/D 12x8b
Oscillator Type	External
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=mc908jl3ecdwe

Appendix C
MC68HC908KL3E/KK3E

Chapter 1
General Description

1.1 Introduction

The MC68H(R)C908JL3E is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). The M68HC08 Family is based on the customer-specified integrated circuit (CSIC) design strategy. All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

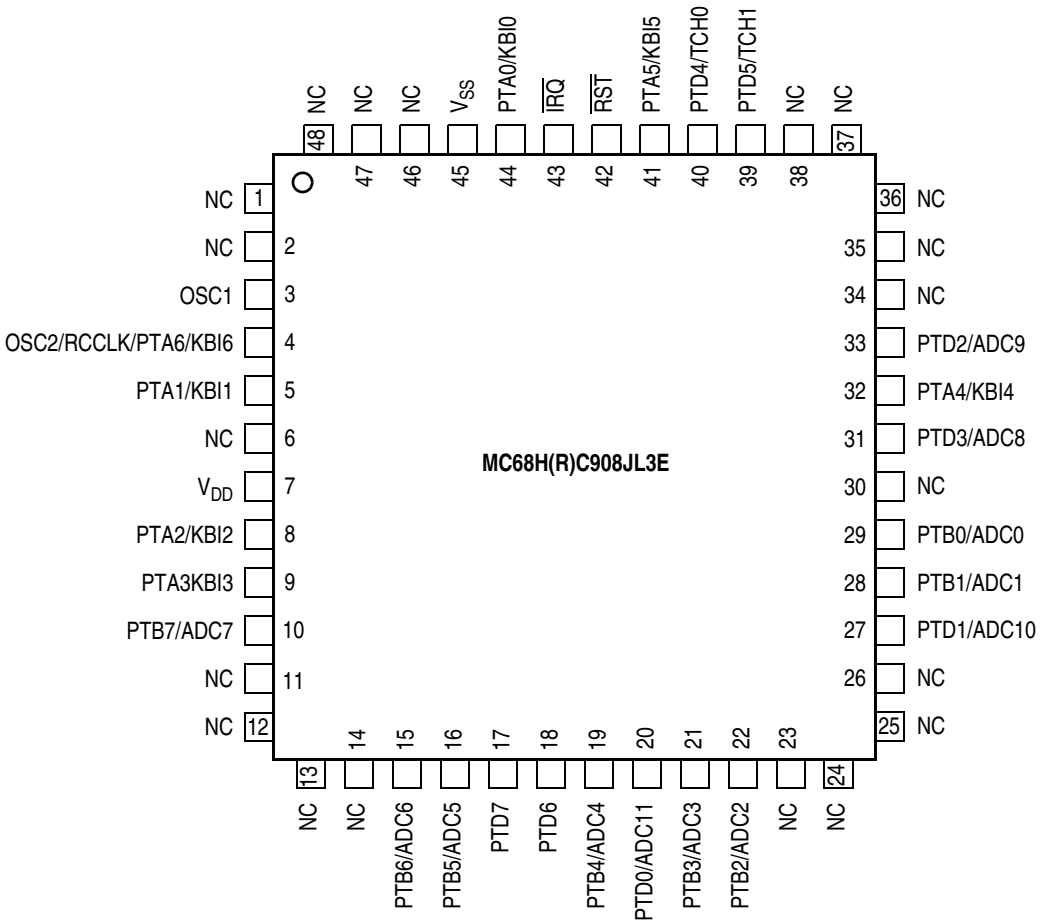
A list of MC68H(R)C908JL3E device variations is shown in Table 1-1.

Table 1-1. Summary of Device Variations

Device Type	Operating Voltage	LVI	ADC	Oscillator Option	Memory	Pin Count	Device
Flash	3V, 5V	Yes	Yes	XTAL	4,096 bytes Flash	28	MC68HC908JL3E
						20	MC68HC908JK3E
					1,536 bytes Flash	20	MC68HC908JK1E
				RC	4,096 bytes Flash	28	MC68HRC908JL3E
						20	MC68HRC908JK3E
					1,536 bytes Flash	20	MC68HRC908JK1E
Low Voltage Flash(1)	2.2 to 5.5V	No	Yes	XTAL	4,096 bytes Flash	28	MC68HLC908JL3E
						20	MC68HLC908JK3E
					1,536 bytes Flash	20	MC68HLC908JK1E
ROM(2)	3V, 5V	Yes	Yes	XTAL	4,096 bytes ROM	28	MC68HC08JL3E
						20	MC68HC08JK3E
				RC		28	MC68HRC08JL3E
						20	MC68HRC08JK3E
Flash, ADC-less(3)	3V, 5V	Yes	No	XTAL	4,096 bytes Flash	28	MC68HC908KL3E
						20	MC68HC908KK3E

1. Low-voltage Flash devices are documented in Appendix A MC68HLC908JL3E/JK3E/JK1E.
2. ROM devices are documented in Appendix B MC68H(R)C08JL3E/JK3E.
3. Flash, ADC-less devices are documented in Appendix C MC68HC908KL3E/KK3E.

All references to the MC68H(R)C908JL3E in this data book apply equally to the MC68H(R)C908JK3E and MC68H(R)C908JK1E, unless otherwise stated.



NC: No connection

Figure 1-4. 48-Pin LQFP Pin Assignment

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE04	Interrupt Status Register 1 (INT1)	Read:	0	IF5	IF4	IF3	0	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE05	Interrupt Status Register 2 (INT2)	Read:	IF14	0	0	0	0	0	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE06	Interrupt Status Register 3 (INT3)	Read:	0	0	0	0	0	0	0	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE07	Reserved	Read: Write:	R	R	R	R	R	R	R	R
$FE08	Flash Control Register (FLCR)	Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE09	Flash Block Protect Register (FLBPR)	Read: Write:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Reset:	0	0	0	0	0	0	0	0
$FE0A ↓ $FE0B	Reserved	Read: Write:	R	R	R	R	R	R	R	R
$FE0C	Break Address High Register (BRKH)	Read: Write:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Reset:	0	0	0	0	0	0	0	0
$FE0D	Break Address Low Register (BRKL)	Read: Write:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Reset:	0	0	0	0	0	0	0	0
$FE0E	Break Status and Control Register (BRKSCR)	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FFFF	COP Control Register (COPCTL)	Read:	Low byte of reset vector							
		Write:	Writing clears COP counter (any value)							
		Reset:	Unaffected by reset							

= Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 4)

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 1. The following paragraphs describe the functions of the condition code register.

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	V	1	1	H	I	N	Z	C
Reset:	X	1	1	X	1	X	X	X

X = Indeterminate

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.

After the I bit is cleared, the highest-priority interrupt request is serviced first.

A return-from-interrupt (RTI) instruction pulls the CPU registers from the stack and restores the interrupt mask from the stack. After any reset, the interrupt mask is set and can be cleared only by the clear interrupt mask software instruction (CLI).

N — Negative Flag

The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result.

1 = Negative result
0 = Non-negative result

Table 4-1. 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				
CLR *opr* CLRA CLRX CLRH CLR *opr*,X CLR ,X CLR *opr*,SP	Clear	M ← $00 A ← $00 X ← $00 H ← $00 M ← $00 M ← $00 M ← $00	0	–	–	0	1	–	DIR INH INH INH IX1 IX SP1	3F 4F 5F 8C 6F 7F 9E6F	dd ff ff	3 1 1 1 3 2 4
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X CMP *opr*,SP CMP *opr*,SP	Compare A with M	(A) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A1 B1 C1 D1 E1 F1 9EE1 9ED1	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
COM *opr* COMA COMX COM *opr*,X COM ,X COM *opr*,SP	Complement (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (M) X ← ($\overline{X}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	0	–	–	↕	↕	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd ff ff	4 1 1 4 3 5
CPHX #*opr* CPHX *opr*	Compare H:X with M	(H:X) – (M:M + 1)	↕	–	–	↕	↕	↕	IMM DIR	65 75	ii ii+1 dd	3 4
CPX #*opr* CPX *opr* CPX *opr* CPX ,X CPX *opr*,X CPX *opr*,X CPX *opr*,SP CPX *opr*,SP	Compare X with M	(X) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A3 B3 C3 D3 E3 F3 9EE3 9ED3	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
DAA	Decimal Adjust A	$(A)_{10}$	U	–	–	↕	↕	↕	INH	72		2
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

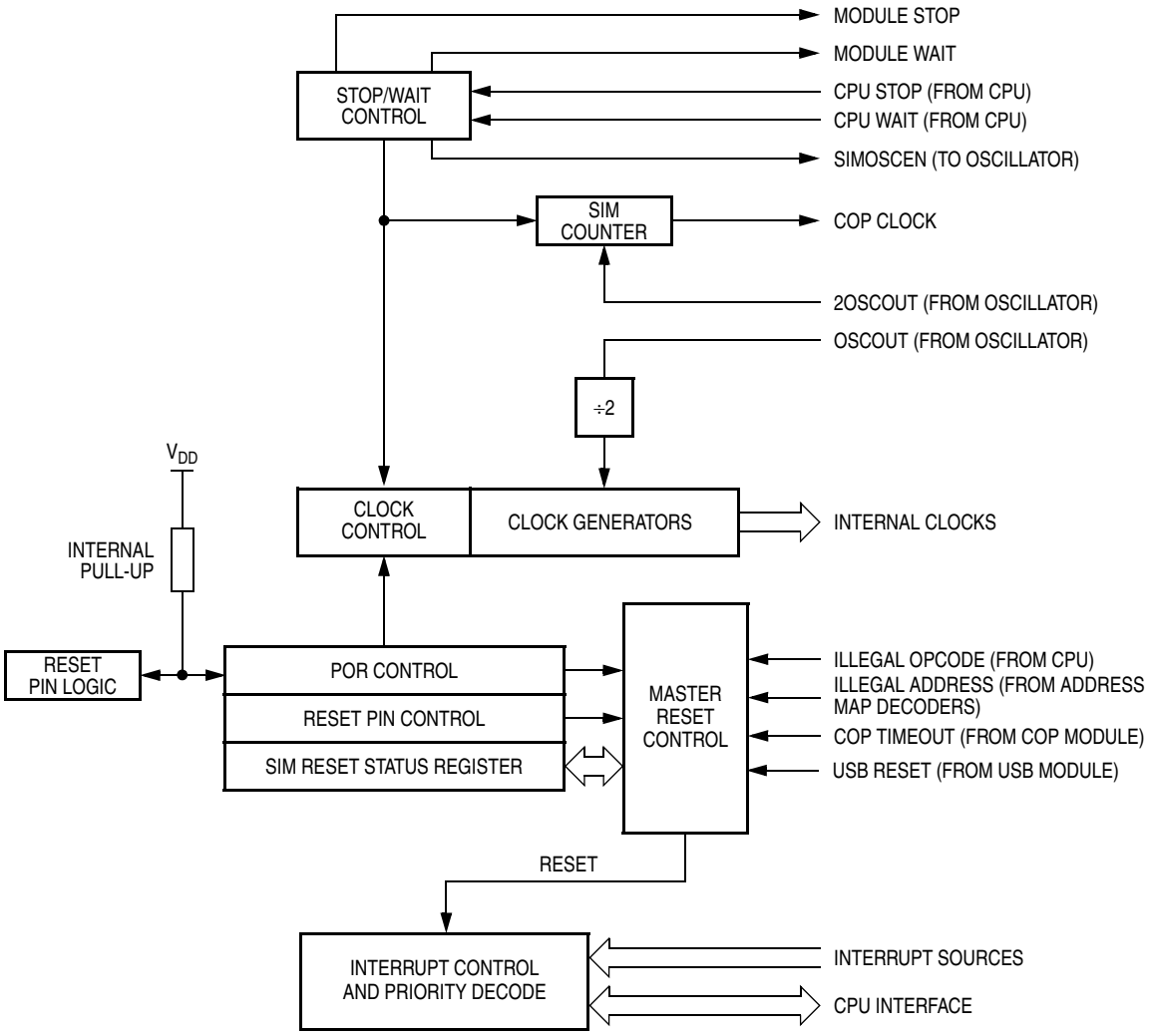


Figure 5-1. SIM Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE00	Break Status Register (BSR)	Read: Write:	R	R	R	R	R	R	SBSW NOTE	R
		Reset:	0	0	0	0	0	0	0	0
Note: Writing a 0 clears SBSW.										
$FE01	Reset Status Register (RSR)	Read: Write:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
		POR:	1	0	0	0	0	0	0	0
$FE02	Reserved	Read: Write:	R	R	R	R	R	R	R	R
		Reset:								
$FE03	Break Flag Control Register (BFCR)	Read: Write:	BCFE	R	R	R	R	R	R	R
		Reset:	0							
			= Unimplemented				R	= Reserved		

Figure 5-2. SIM I/O Register Summary

5.3 Reset and System Initialization

The MCU has these reset sources:

- Power-on reset module (POR)
- External reset pin ($\overline{RST}$)
- Computer operating properly module (COP)
- Low-voltage inhibit module (LVI)
- Illegal opcode
- Illegal address

All of these resets produce the vector $FFFE–$FFFF ($FEFE–$FEFF in Monitor mode) and assert the internal reset signal (IRST). IRST causes all registers to be returned to their default values and all modules to be returned to their reset states.

An internal reset clears the SIM counter (see 5.4 SIM Counter), but an external reset does not. Each of the resets sets a corresponding bit in the reset status register (RSR). (See 5.7 SIM Registers.)

5.3.1 External Pin Reset

The $\overline{RST}$ pin circuits include an internal pull-up device. Pulling the asynchronous $\overline{RST}$ pin low halts all processing. The PIN bit of the reset status register (RSR) is set as long as $\overline{RST}$ is held low for a minimum of 67 2OSCOUT cycles, assuming that the POR was not the source of the reset. See Table 5-2 for details. Figure 5-4 shows the relative timing.

Table 5-2. PIN Bit Set Timing

Reset Type	Number of Cycles Required to Set PIN
POR	4163 (4096 + 64 + 3)
All others	67 (64 + 3)

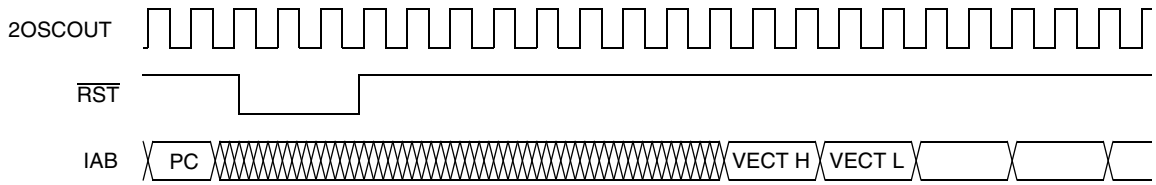


Figure 5-4. External Reset Timing

5.3.2 Active Resets from Internal Sources

All internal reset sources actively pull the $\overline{RST}$ pin low for 32 2OSCOUT cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles (Figure 5-5). An internal reset can be caused by an illegal address, illegal opcode, COP time-out, or POR. (See Figure 5-6.) Note that for POR resets, the SIM cycles through 4096 2OSCOUT cycles during which the SIM forces the $\overline{RST}$ pin low. The internal reset signal then follows the sequence from the falling edge of $\overline{RST}$ shown in Figure 5-5.

6.4 I/O Signals

The following paragraphs describe the oscillator I/O signals.

6.4.1 Crystal Amplifier Input Pin (OSC1)

OSC1 pin is an input to the crystal oscillator amplifier or the input to the RC oscillator circuit.

6.4.2 Crystal Amplifier Output Pin (OSC2/PTA6/RCCLK)

For the X-tal oscillator device, OSC2 pin is the output of the crystal oscillator inverting amplifier.

For the RC oscillator device, OSC2 pin can be configured as a general purpose I/O pin PTA6, or the output of the internal RC oscillator clock, RCCLK.

Device	Oscillator	OSC2 pin function
MC68HC908JL3E/JK3E/JK1E	X-tal	Inverting OSC1
MC68HRC908JL3E/JK3E/JK1E	RC	Controlled by PTA6EN bit in PTAPUER ($0D) PTA6EN = 0: RCCLK output PTA6EN = 1: PTA6 I/O

6.4.3 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal comes from the system integration module (SIM) and enables/disables the X-tal oscillator circuit or the RC-oscillator.

6.4.4 X-tal Oscillator Clock (XTALCLK)

XTALCLK is the X-tal oscillator output signal. It runs at the full speed of the crystal (f_{XCLK}) and comes directly from the crystal oscillator circuit. Figure 6-1 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.

6.4.5 RC Oscillator Clock (RCCLK)

RCCLK is the RC oscillator output signal. Its frequency is directly proportional to the time constant of the external R and C. Figure 6-2 shows only the logical relation of RCCLK to OSC1 and may not represent the actual circuitry.

6.4.6 Oscillator Out 2 (2OSCOUT)

2OSCOUT is same as the input clock (XTALCLK or RCCLK). This signal is driven to the SIM module and is used to determine the COP cycles.

6.4.7 Oscillator Out (OSCOUT)

The frequency of this signal is equal to half of the 2OSCOUT, this signal is driven to the SIM for generation of the bus clocks used by the CPU and other modules on the MCU. OSCOUT will be divided again in the SIM and results in the internal bus frequency being one fourth of the XTALCLK or RCCLK frequency.

Chapter 7
Monitor ROM (MON)

7.1 Introduction

This section describes the monitor ROM (MON) and the monitor mode entry methods. The monitor ROM allows complete testing of the MCU through a single-wire interface with a host computer. This mode is also used for programming and erasing of Flash memory in the MCU. Monitor mode entry can be achieved without use of the higher test voltage, V_{TST}, as long as vector addresses $FFFE and $FFFF are blank, thus reducing the hardware requirements for in-circuit programming.

7.2 Features

Features of the monitor ROM include the following:

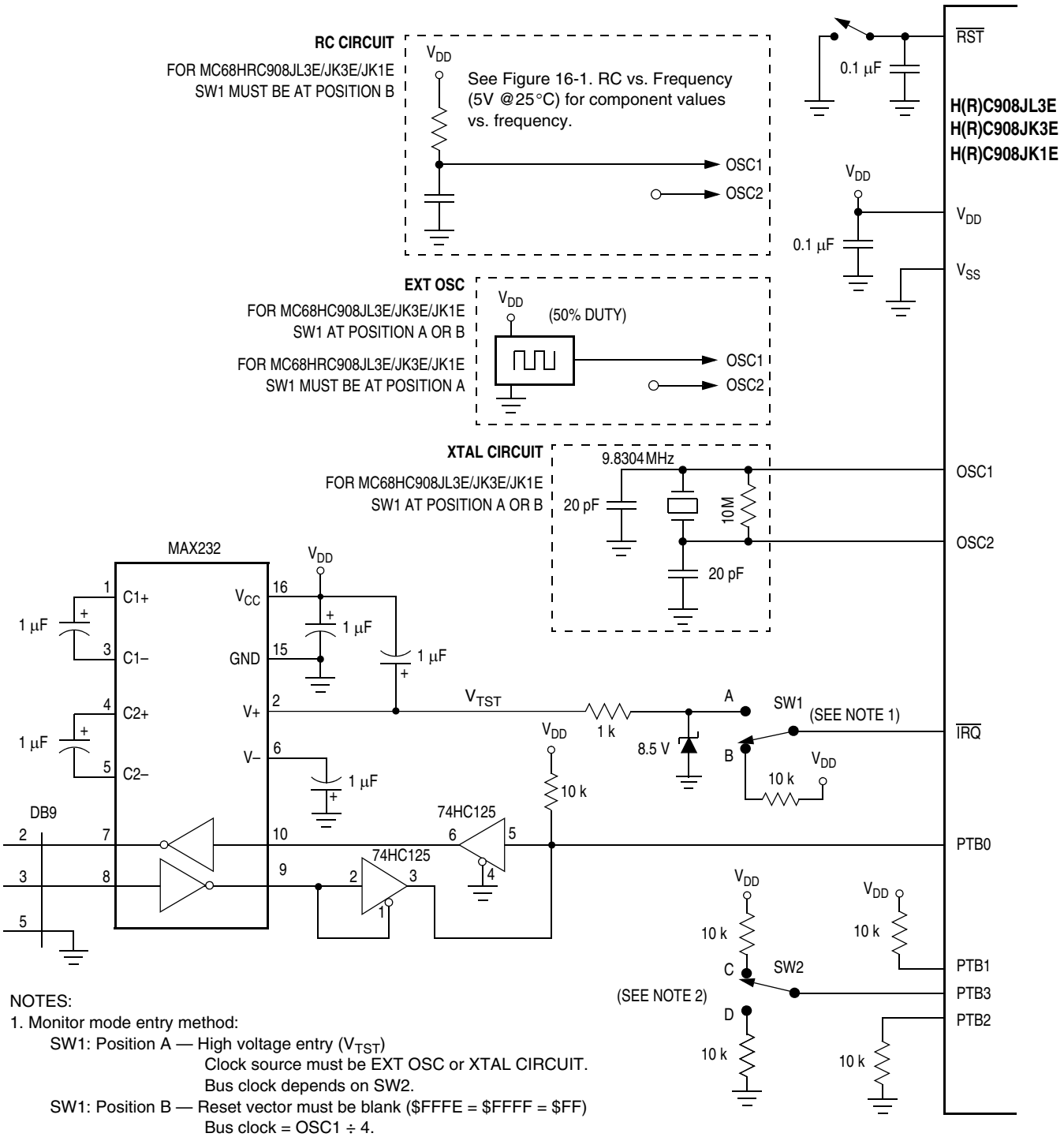
- Normal user-mode pin functionality
- One pin dedicated to serial communication between monitor ROM and host computer
- Standard mark/space non-return-to-zero (NRZ) communication with host computer
- Execution of code in RAM or Flash
- Flash memory security feature[1]
- Flash memory programming interface
- 960 bytes monitor ROM code size
- Monitor mode entry without high voltage, V_{TST}, if reset vector is blank ($FFFE and $FFFF contain $FF)
- Standard monitor mode entry if high voltage, V_{TST}, is applied to $\overline{IRQ}$

7.3 Functional Description

The monitor ROM receives and executes commands from a host computer. Figure 7-1 shows a example circuit used to enter monitor mode and communicate with a host computer via a standard RS-232 interface.

Simple monitor commands can access any memory address. In monitor mode, the MCU can execute host-computer code in RAM while most MCU pins retain normal operating mode functions. All communication between the host computer and the MCU is through the PTB0 pin. A level-shifting and multiplexing interface is required between PTB0 and the host computer. PTB0 is used in a wired-OR configuration and requires a pull-up resistor.

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



NOTES:

1. Monitor mode entry method:
   - SW1: Position A — High voltage entry (V_{TST})
     Clock source must be EXT OSC or XTAL CIRCUIT.
     Bus clock depends on SW2.
   - SW1: Position B — Reset vector must be blank ($FFFE = $FFFF = $FF)
     Bus clock = OSC1 ÷ 4.
2. Affects high voltage entry to monitor mode only (SW1 at position A):
   - SW2: Position C — Bus clock = OSC1 ÷ 4
   - SW2: Position D — Bus clock = OSC1 ÷ 2
5. See Table 16-4. DC Electrical Characteristics (5V) for V_{TST} voltage level requirements.

Figure 7-1. Monitor Mode Circuit

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0020	TIM Status and Control Register (TSC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0021	TIM Counter Register High (TCNTH)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0022	TIM Counter Register Low (TCNTL)	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	TIM Counter Modulo Register Low (TMODL)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0025	TIM Channel 0 Status and Control Register (TSC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0026	TIM Channel 0 Register High (TCH0H)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	Indeterminate after reset							
$0027	TIM Channel 0 Register Low (TCH0L)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	Indeterminate after reset							
$0028	TIM Channel 1 Status and Control Register (TSC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0029	TIM Channel 1 Register High (TCH1H)	Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:								
		Reset:	Indeterminate after reset							
$002A	TIM Channel 1 Register Low (TCH1L)	Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:								
		Reset:	Indeterminate after reset							

= Unimplemented

Figure 8-2. TIM I/O Register Summary

8.9.2 TIM Counter Registers (TCNTH:TCNTL)

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

NOTE

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

Address: $0021 TCNTH

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Write:								
Reset:	0	0	0	0	0	0	0	0

Address: $0022 TCNTL

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 8-5. TIM Counter Registers (TCNTH:TCNTL)

8.9.3 TIM Counter Modulo Registers (TMODH:TMODL)

The read/write TIM modulo registers contain the modulo value for the TIM counter. When the TIM counter reaches the modulo value, the overflow flag (TOF) becomes set, and the TIM counter resumes counting from $0000 at the next timer clock. Writing to the high byte (TMODH) inhibits the TOF bit and overflow interrupts until the low byte (TMODL) is written. Reset sets the TIM counter modulo registers.

Address: $0023 TMODH

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Reset:	1	1	1	1	1	1	1	1

Address: $0024 TMODL

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reset:	1	1	1	1	1	1	1	1

Figure 8-6. TIM Counter Modulo Registers (TMODH:TMODL)

NOTE

Reset the TIM counter before writing to the TIM counter modulo registers.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000A	Port D Control Register (PDCR)	Read:	0	0	0	0	SLOWD7	SLOWD6	PTDPU7	PTDPU6
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000D	Port A Input Pull-up Enable Register (PTAPUE)	Read:	PTA6EN	PTAPUE6	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-1. I/O Port Register Summary

Table 10-1. Port Control Register Bits Summary

Port	Bit	DDR	Module Control			Pin
			Module	Register	Control Bit	
A	0	DDRA0	KBI	KBIER ($001B)	KBIE0	PTA0/KBI0
	1	DDRA1			KBIE1	PTA1/KBI1
	2	DDRA2			KBIE2	PTA2/KBI2
	3	DDRA3			KBIE3	PTA3/KBI3
	4	DDRA4			KBIE4	PTA4/KBI4
	5	DDRA5			KBIE5	PTA5/KBI5
	6	DDRA6	OSC KBI	PTAPUE ($000D) KBIER ($001B)	PTA6EN KBIE6	RCCLK/PTA6/KBI6[1]
B	0	DDRB0	ADC	ADSCR ($003C)	ADCH[4:0]	PTB0/ADC0
	1	DDRB1				PTB1/ADC1
	2	DDRB2				PTB2/ADC2
	3	DDRB3				PTB3/ADC3
	4	DDRB4				PTB4/ADC4
	5	DDRB5				PTB5/ADC5
	6	DDRB6				PTB6/ADC6
	7	DDRB7				PTB7/ADC7
D	0	DDRD0	ADC	ADSCR ($003C)	ADCH[4:0]	PTD0/ADC11
	1	DDRD1				PTD1/ADC10
	2	DDRD2				PTD2/ADC9
	3	DDRD3				PTD3/ADC8
	4	DDRD4	TIM	TSC0 ($0025)	ELS0B:ELS0A	PTD4/TCH0
	5	DDRD5		TSC1 ($0028)	ELS1B:ELS1A	PTD5/TCH1
	6	DDRD6	—	—	—	PTD6
	7	DDRD7	—	—	—	PTD7

1. RCCLK/PTA6/KBI6 pin is only available on MC68HRC908JL3E/JK3E/JK1E devices (RC option); PTAPUE register has priority control over the port pin. RCCLK/PTA6/KBI6 is the OSC2 pin on MC68HC908JL3E/JK3E/JK1E devices (X-TAL option).

11.3.1 $\overline{\text{IRQ}}$ Pin

A zero on the $\overline{\text{IRQ}}$ pin can latch an interrupt request into the IRQ latch. A vector fetch, software clear, or reset clears the IRQ latch.

If the MODE bit is set, the $\overline{\text{IRQ}}$ pin is both falling-edge-sensitive and low-level-sensitive. With MODE set, both of the following actions must occur to clear IRQ:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the latch. Software may generate the interrupt acknowledge signal by writing a logic one to the ACK bit in the interrupt status and control register (INTSCR). The ACK bit is useful in applications that poll the $\overline{\text{IRQ}}$ pin and require software to clear the IRQ latch. Writing to the ACK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACK does not affect subsequent transitions on the $\overline{\text{IRQ}}$ pin. A falling edge that occurs after writing to the ACK bit latches another interrupt request. If the IRQ mask bit, IMASK, is clear, the CPU loads the program counter with the vector address at locations $FFFA and $FFFB.
- Return of the $\overline{\text{IRQ}}$ pin to logic one — As long as the $\overline{\text{IRQ}}$ pin is at logic zero, IRQ remains active.

The vector fetch or software clear and the return of the $\overline{\text{IRQ}}$ pin to logic one may occur in any order. The interrupt request remains pending as long as the $\overline{\text{IRQ}}$ pin is at logic zero. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

If the MODE bit is clear, the $\overline{\text{IRQ}}$ pin is falling-edge-sensitive only. With MODE clear, a vector fetch or software clear immediately clears the IRQ latch.

The IRQF bit in the INTSCR register can be used to check for pending interrupts. The IRQF bit is not affected by the IMASK bit, which makes it useful in applications where polling is preferred.

Use the BIH or BIL instruction to read the logic level on the $\overline{\text{IRQ}}$ pin.

NOTE

When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine.

NOTE

An internal pull-up 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).

11.4 IRQ Module During Break Interrupts

The system integration module (SIM) controls whether the IRQ latch can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear the latches during the break state. (See Chapter 5 System Integration Module (SIM).)

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

To protect the latches during the break state, write a zero to the BCFE bit. With BCFE at zero (its default state), writing to the ACK bit in the IRQ status and control register during the break state has no effect on the IRQ latch.

Chapter 12
Keyboard Interrupt Module (KBI)

12.1 Introduction

The keyboard interrupt module (KBI) provides seven independently maskable external interrupts which are accessible via PTA0–PTA6 pins.

12.2 Features

Features of the keyboard interrupt module include the following:

- Seven keyboard interrupt pins with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Software configurable pull-up device if input pin is configured as input port bit
- 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:	0	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 12-1. KBI I/O Register Summary

12.3 I/O Pins

The seven keyboard interrupt pins are shared with standard port I/O pins. The full name of the KBI pins are listed in Table 12-1. The generic pin name appear in the text that follows.

Table 12-1. 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	RCCLK/PTA6/KBI6(1)	KBIE6

1. RCCLK/PTA6/KBI6 pin is only available on MC68HRC908JL3E/JK3E/JK1E devices (RC option).

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

If the MODEK bit is clear, the keyboard interrupt pin is falling-edge-sensitive only. With MODEK clear, a vector fetch or software clear immediately clears the keyboard interrupt request.

Reset clears the keyboard interrupt request and the MODEK bit, clearing the interrupt request even if a keyboard interrupt pin stays at 0.

The keyboard flag bit (KEYF) in the keyboard status and control register can be used to see if a pending interrupt exists. The KEYF bit is not affected by the keyboard interrupt mask bit (IMASKK) which makes it useful in applications where polling is preferred.

To determine the logic level on a keyboard interrupt pin, disable the pull-up device, 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.

12.4.1 Keyboard Initialization

When a keyboard interrupt pin is enabled, it takes time for the internal pull-up 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.

12.5 Keyboard Interrupt Registers

Two registers control the operation of the keyboard interrupt module:

- Keyboard status and control register
- Keyboard interrupt enable register

Table 16-7. DC Electrical Characteristics (3V) (Continued)

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
LVI reset voltage	V_{LVR3}	2.0	2.4	2.69	V

1. V_{DD} = 2.7 to 3.3 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} = 2MHz). 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} = 2MHz). 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.

16.9 3V Control Timing

Table 16-8. Control Timing (3V)

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency[2]	f_{OP}	—	4	MHz
$\overline{RST}$ input pulse width low[3]	t_{IRL}	1.5	—	µs

1. V_{DD} = 2.7 to 3.3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H; timing shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Some modules may require a minimum frequency greater than dc for proper operation; see appropriate table for this information.
3. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.

16.11 Typical Supply Currents

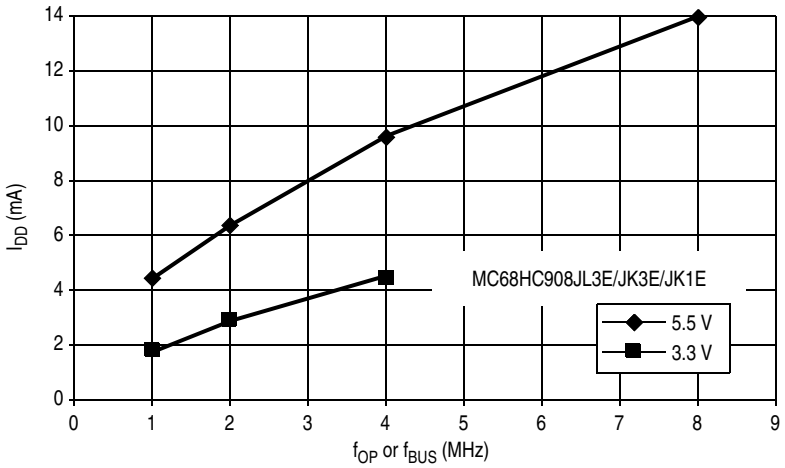


Figure 16-3. Typical Operating I_{DD} (MC68HC908JL3E/JK3E/JK1E), with All Modules Turned On (25°C)

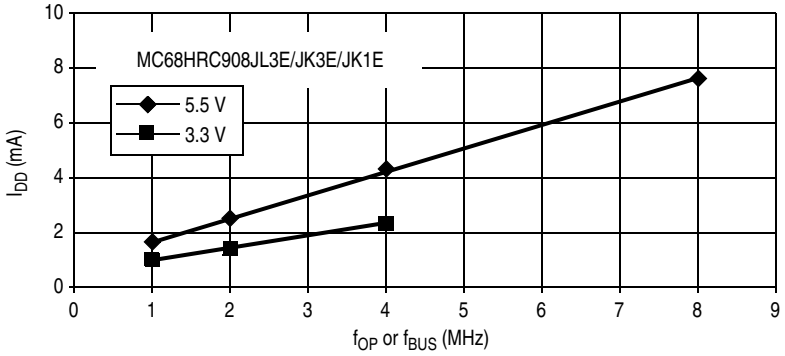


Figure 16-4. Typical Operating I_{DD} (MC68HRC908JL3E/JK3E/JK1E), with All Modules Turned On (25°C)

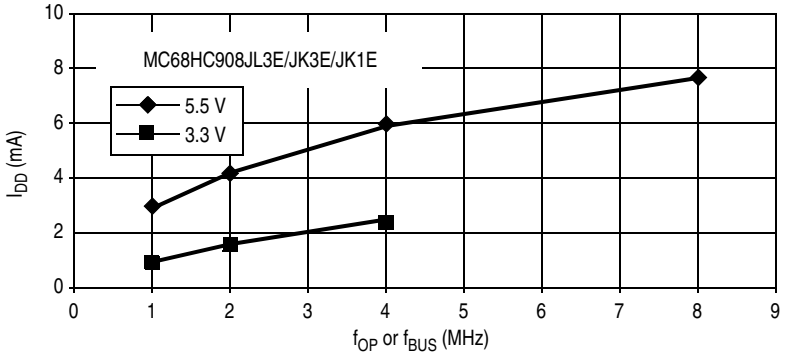


Figure 16-5. Typical Wait Mode I_{DD} (MC68HC908JL3E/JK3E/JK1E), with All Modules Turned Off (25°C)