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	HC11
Core Size	8-Bit
Speed	2MHz
Connectivity	SCI, SPI
Peripherals	POR, WDT
Number of I/O	30
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.21x24.21)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mchc11f1cfne2r

TABLE OF CONTENTS

Paragraph **Title** **Page**

SECTION 1INTRODUCTION

SECTION 2 PIN DESCRIPTIONS

SECTION 3 CENTRAL PROCESSING UNIT

The add, subtract, and compare instructions associated with both A and B (ABA, SBA, and CBA) only operate in one direction, making it important to plan ahead to ensure that the correct operand is in the correct accumulator.

3.1.2 Index Register X (IX)

The IX register provides a 16-bit indexing value that can be added to the 8-bit offset provided in an instruction to create an effective address. The IX register can also be used as a counter or as a temporary storage register.

3.1.3 Index Register Y (IY)

The 16-bit IY register performs an indexed mode function similar to that of the IX register. However, most instructions using the IY register require an extra byte of machine code and an extra cycle of execution time because of the way the opcode map is implemented. Refer to **3.3 Opcodes and Operands** for further information.

3.1.4 Stack Pointer (SP)

The M68HC11 CPU has an automatic program stack. This stack can be located anywhere in the address space and can be any size up to the amount of memory available in the system. Normally the SP is initialized by one of the first instructions in an application program. The stack is configured as a data structure that grows downward from high memory to low memory. Each time a new byte is pushed onto the stack, the SP is decremented. Each time a byte is pulled from the stack, the SP is incremented. At any given time, the SP holds the 16-bit address of the next free location in the stack. **Figure 3-2** is a summary of SP operations.

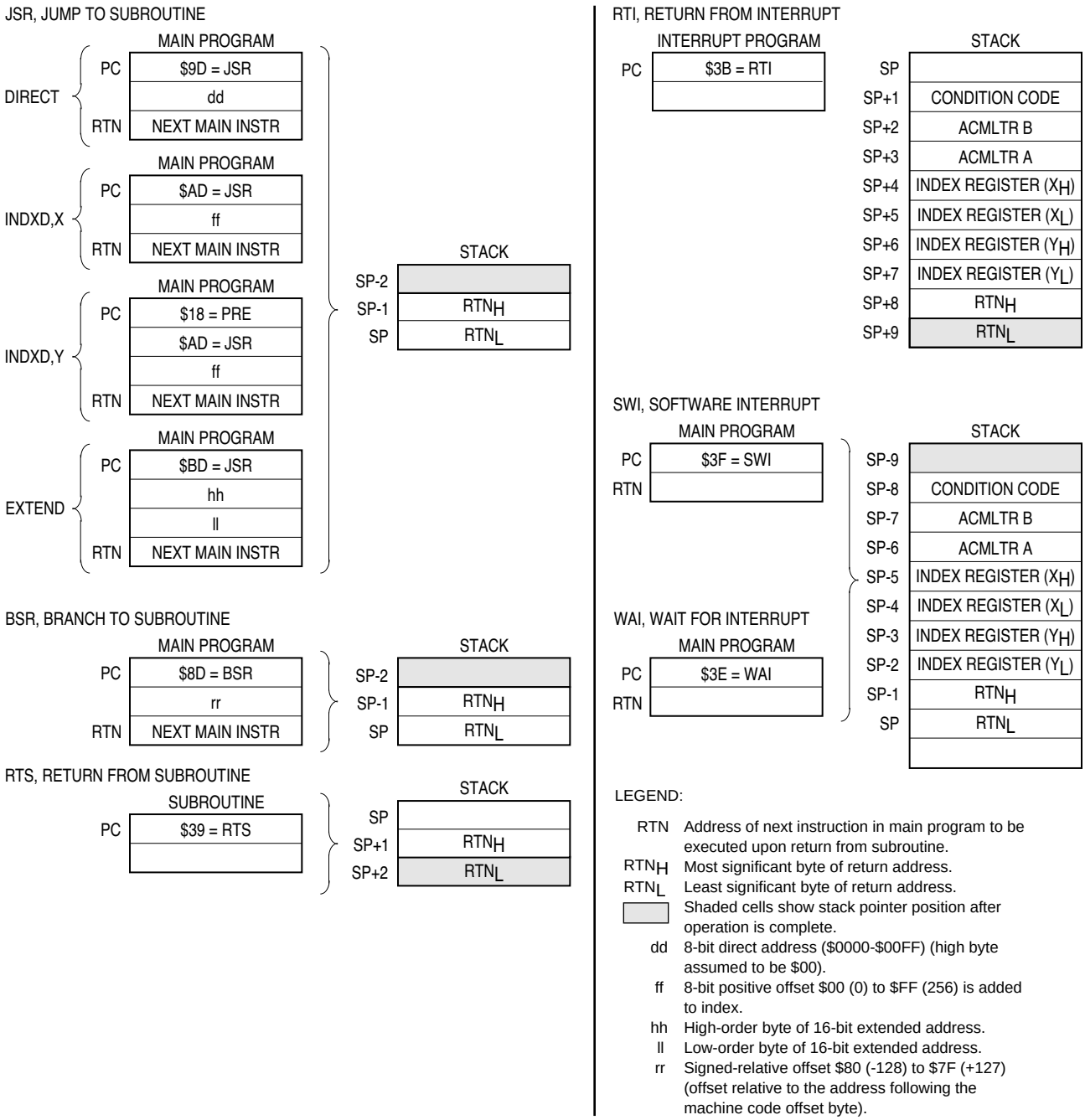

Figure 3-2 Stacking Operations

When a subroutine is called by a jump to subroutine (JSR) or branch to subroutine (BSR) instruction, the address of the instruction after the JSR or BSR is automatically pushed onto the stack, least significant byte first. When the subroutine is finished, a return from subroutine (RTS) instruction is executed. The RTS pulls the previously stacked return address from the stack, and loads it into the program counter. Execution then continues at this recovered return address.

3.1.6.3 Zero (Z)

The Z bit is set if the result of an arithmetic, logic, or data manipulation operation is zero. Otherwise, the Z bit is cleared. Compare instructions do an internal implied subtraction and the condition codes, including Z, reflect the results of that subtraction. A few operations (INX, DEX, INY, and DEY) affect the Z bit and no other condition flags. For these operations, only = and - conditions can be determined.

3.1.6.4 Negative (N)

The N bit is set if the result of an arithmetic, logic, or data manipulation operation is negative (MSB = 1). Otherwise, the N bit is cleared. A result is said to be negative if its most significant bit (MSB) is a one. A quick way to test whether the contents of a memory location has the MSB set is to load it into an accumulator and then check the status of the N bit.

3.1.6.5 Interrupt Mask (I)

The interrupt request (IRQ) mask (I bit) is a global mask that disables all maskable interrupt sources. While the I bit is set, interrupts can become pending, but the operation of the CPU continues uninterrupted until the I bit is cleared. After any reset, the I bit is set by default and can only be cleared by a software instruction. When an interrupt is recognized, the I bit is set after the registers are stacked, but before the interrupt vector is fetched. After the interrupt has been serviced, a return from interrupt instruction is normally executed, restoring the registers to the values that were present before the interrupt occurred. Normally, the I bit is zero after a return from interrupt is executed. Although the I bit can be cleared within an interrupt service routine, “nesting” interrupts in this way should only be done when there is a clear understanding of latency and of the arbitration mechanism. Refer to **SECTION 5 RESETS AND INTERRUPTS**.

3.1.6.6 Half Carry (H)

The H bit is set when a carry occurs between bits 3 and 4 of the arithmetic logic unit during an ADD, ABA, or ADC instruction. Otherwise, the H bit is cleared. Half carry is used during BCD operations.

3.1.6.7 X Interrupt Mask (X)

The $\overline{\text{XIRQ}}$ mask (X) bit disables interrupts from the $\overline{\text{XIRQ}}$ pin. After any reset, X is set by default and must be cleared by a software instruction. When an $\overline{\text{XIRQ}}$ interrupt is recognized, the X and I bits are set after the registers are stacked, but before the interrupt vector is fetched. After the interrupt has been serviced, an RTI instruction is normally executed, causing the registers to be restored to the values that were present before the interrupt occurred. The X interrupt mask bit is set only by hardware ($\overline{\text{RESET}}$ or $\overline{\text{XIRQ}}$ acknowledge). X is cleared only by program instruction (TAP, where the associated bit of A is zero; or RTI, where bit 6 of the value loaded into the CCR from the stack has been cleared). There is no hardware action for clearing X.

Table 4-1 Register and Control Bit Assignments

The register block can be remapped to any 4-Kbyte boundary.

	Bit 7	6	5	4	3	2	1	Bit 0	
$1000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
$1001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
$1002	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	PORTG
$1003	DDG7	DDG6	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	DDRG
$1004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
$1005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	PORTF
$1006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
$1007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
$1008	0	0	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
$1009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
$100A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE
$100B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	CFORC
$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	OC1M
$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	OC1D
$100E	Bit 15	14	13	12	11	10	9	Bit 8	TCNT (High)
$100F	Bit 7	6	5	4	3	2	1	Bit 0	TCNT (Low)
$1010	Bit 15	14	13	12	11	10	9	Bit 8	TIC1 (High)
$1011	Bit 7	6	5	4	3	2	1	Bit 0	TIC1 (Low)
$1012	Bit 15	14	13	12	11	10	9	Bit 8	TIC2 (High)
$1013	Bit 7	6	5	4	3	2	1	Bit 0	TIC2 (Low)
$1014	Bit 15	14	13	12	11	10	9	Bit 8	TIC3 (High)
$1015	Bit 7	6	5	4	3	2	1	Bit 0	TIC3 (Low)
$1016	Bit 15	14	13	12	11	10	9	Bit 8	TOC1 (High)
$1017	Bit 7	6	5	4	3	2	1	Bit 0	TOC1 (Low)
$1018	Bit 15	14	13	12	11	10	9	Bit 8	TOC2 (High)
$1019	Bit 7	6	5	4	3	2	1	Bit 0	TOC2 (Low)
$101A	Bit 15	14	13	12	11	10	9	Bit 8	TOC3 (High)
$101B	Bit 7	6	5	4	3	2	1	Bit 0	TOC3 (Low)
$101C	Bit 15	14	13	12	11	10	9	Bit 8	TOC4 (High)
$101D	Bit 7	6	5	4	3	2	1	Bit 0	TOC4 (Low)
$101E	Bit 15	14	13	12	11	10	9	Bit 8	TI4/O5 (High)
$101F	Bit 7	6	5	4	3	2	1	Bit 0	TI4/O5 (Low)
$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
$1022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	TMSK1
$1023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	TFLG1
$1024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	TMSK2
$1025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	TFLG2
$1026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	PACTL
$1027	Bit 7	6	5	4	3	2	1	Bit 0	PACNT

ADPU — A/D Power-Up

Refer to **SECTION 10 ANALOG-TO-DIGITAL CONVERTER**.

0 = A/D system disabled
1 = A/D system power enabled

CSEL — Clock Select

Selects alternate clock source for on-chip EEPROM and A/D charge pumps. On-chip RC clock should be used when E clock falls below 1 MHz. Refer to **SECTION 10 ANALOG-TO-DIGITAL CONVERTER**.

0 = A/D and EEPROM use system E clock
1 = A/D and EEPROM use internal RC clock

IRQE — Configure $\overline{\text{IRQ}}$ for Falling Edge-Sensitive Operation

Refer to **SECTION 5 RESETS AND INTERRUPTS**.

0 = Low level-sensitive operation.
1 = Falling edge-sensitive only operation.

DLY — Enable Oscillator Start-up Delay

Refer to **SECTION 5 RESETS AND INTERRUPTS**.

0 = The oscillator start-up delay coming out of STOP is bypassed and the MCU resumes processing within about four bus cycles.
1 = A delay of approximately 4000 E-clock cycles is imposed as the MCU is started up from the STOP power-saving mode.

CME — Clock Monitor Enable

In order to use both STOP and clock monitor, the CME bit must be written to zero before executing STOP, then written to one after recovering from STOP. Refer to **SECTION 5 RESETS AND INTERRUPTS**.

0 = Clock monitor disabled
1 = Clock monitor enabled

FCME — Force Clock Monitor Enable

When FCME equals one, slow or stopped clocks will cause a clock failure reset. To use STOP mode, FCME must always equal zero. Refer to **SECTION 5 RESETS AND INTERRUPTS**.

0 = Clock monitor follows state of CME bit
1 = Clock monitor enabled and cannot be disabled until next reset

CR[1:0] — COP Timer Rate Select Bits

These control bits determine a scaling factor for the watchdog timer. Refer to **SECTION 5 RESETS AND INTERRUPTS**.

4.3.2.4 OPT2 Register

The system configuration options 2 register (OPT2) controls three additional system options.

The illegal opcode trap mechanism works for all unimplemented opcodes on all four opcode map pages. The address stacked as the return address for the illegal opcode interrupt is the address of the first byte of the illegal opcode. Otherwise, it would be almost impossible to determine whether the illegal opcode had been one or two bytes. The stacked return address can be used as a pointer to the illegal opcode so the illegal opcode service routine can evaluate the offending opcode.

5.4.4 Software Interrupt

SWI is an instruction, and thus cannot be interrupted until complete. SWI is not inhibited by the global mask bits in the CCR. Because execution of SWI sets the I mask bit, once an SWI interrupt begins, other interrupts are inhibited until SWI is complete, or until user software clears the I bit in the CCR.

5.4.5 Maskable Interrupts

The maskable interrupt structure of the MCU can be extended to include additional external interrupt sources through the $\overline{\text{IRQ}}$ pin. The default configuration of this pin is a low-level sensitive wired-OR network. When an event triggers an interrupt, a software accessible interrupt flag is set. When enabled, this flag causes a constant request for interrupt service. After the flag is cleared, the service request is released.

5.4.6 Reset and Interrupt Processing

Figure 5-1 and **Figure 5-3** illustrate the reset and interrupt process. **Figure 5-1** illustrates how the CPU begins from a reset and how interrupt detection relates to normal opcode fetches. **Figure 5-3** is an expansion of a block in **Figure 5-1** and illustrates interrupt priorities. **Figure 5-5** shows the resolution of interrupt sources within the SCI subsystem.

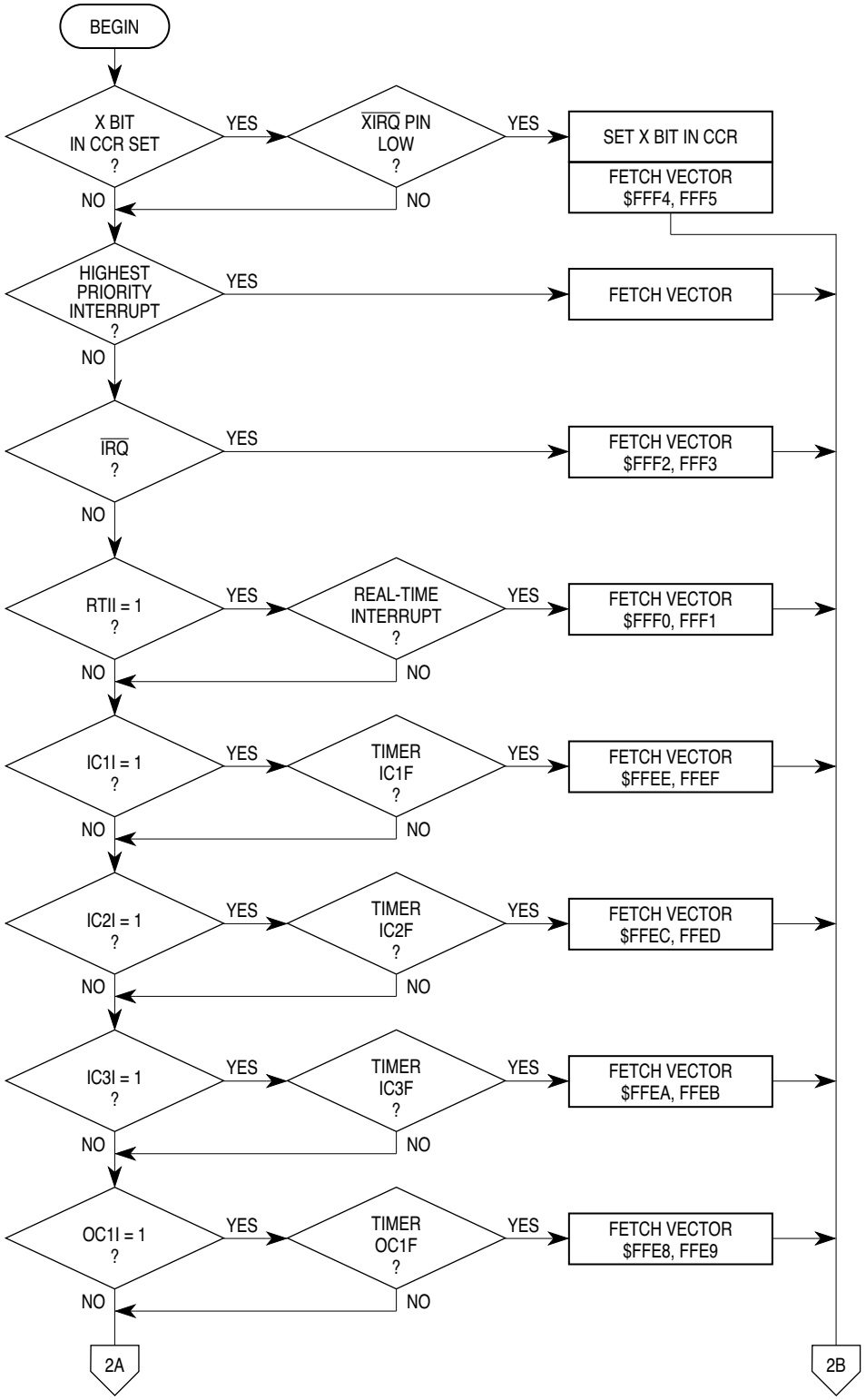

Figure 5-3 Interrupt Priority Resolution (1 of 2)

5.5.1 WAIT

The WAI opcode places the MCU in the WAIT condition, during which the CPU registers are stacked and CPU processing is suspended until a qualified interrupt is detected. The interrupt can be an external $\overline{\text{IRQ}}$, an $\overline{\text{XIRQ}}$, or any of the internally generated interrupts, such as the timer or serial interrupts. The on-chip crystal oscillator remains active throughout the WAIT standby period.

The reduction of power in the WAIT condition depends on how many internal clock signals driving on-chip peripheral functions can be shut down. The CPU is always shut down during WAIT. While in the wait state, the address/data bus repeatedly runs read cycles to the address where the CCR contents were stacked. Ensuring that the stack contents are placed in internal RAM will further reduce power consumption. The MCU leaves the wait state when it senses any interrupt that has not been masked.

The free-running timer system is shut down only if the I bit is set to one and the COP system is disabled by NOCOP being set to one. Several other systems can also be in a reduced power consumption state depending on the state of software-controlled configuration control bits. Power consumption by the analog-to-digital (A/D) converter is not affected significantly by the WAIT condition. However, the A/D converter current can be eliminated by writing the ADPU bit to zero. The SPI system is enabled or disabled by the SPE control bit. The SCI transmitter is enabled or disabled by the TE bit, and the SCI receiver is enabled or disabled by the RE bit. Therefore the power consumption in WAIT is dependent on the particular application.

5.5.2 STOP

Executing the STOP instruction while the S bit in the CCR is equal to zero places the MCU in the STOP condition. If the S bit is not zero, the STOP opcode is treated as a no-op (NOP). The STOP condition offers minimum power consumption because all clocks, including the crystal oscillator, are stopped while in this mode. To exit STOP and resume normal processing, a logic low level must be applied to one of the external interrupts ($\overline{\text{IRQ}}$ or $\overline{\text{XIRQ}}$) or to the $\overline{\text{RESET}}$ pin. A pending edge-triggered $\overline{\text{IRQ}}$ can also bring the CPU out of STOP.

Because all clocks are stopped in this mode, all internal peripheral functions also stop. The data in the internal RAM is retained as long as V_{DD} power is maintained. The CPU state and I/O pin levels are static and are unchanged by STOP. Therefore, when an interrupt restarts the system, the MCU resumes processing as if there were no interruption. If reset is used to restart the system a normal reset sequence results where all I/O pins and functions are also restored to their initial states.

To use the $\overline{\text{IRQ}}$ pin as a means of recovering from STOP, the I bit in the CCR must be clear ($\overline{\text{IRQ}}$ not masked). The $\overline{\text{XIRQ}}$ pin can be used to wake up the MCU from STOP regardless of the state of the X bit in the CCR, although the recovery sequence depends on the state of the X bit. If X is set to zero ($\overline{\text{XIRQ}}$ not masked), the MCU starts up, beginning with the stacking sequence leading to normal service of the $\overline{\text{XIRQ}}$ request. If X is set to one ($\overline{\text{XIRQ}}$ masked or inhibited), then processing continues with the instruction that immediately follows the STOP instruction, and no $\overline{\text{XIRQ}}$ interrupt service is requested or pending.

7.5 SCI Error Detection

Three error conditions, SCDR overrun, received bit noise, and framing can occur during generation of SCI system interrupts. Three bits (OR, NF, and FE) in the serial communications status register (SCSR) indicate if one of these error conditions exists.

The overrun error (OR) bit is set when the next byte is ready to be transferred from the receive shift register to the SCDR and the SCDR is already full (RDRF bit is set). When an overrun error occurs, the data that caused the overrun is lost and the data that was already in SCDR is not disturbed. The OR is cleared when the SCSR is read (with OR set), followed by a read of the SCDR.

The noise flag (NF) bit is set if there is noise on any of the received bits, including the start and stop bits. The NF bit is not set until the RDRF flag is set. The NF bit is cleared when the SCSR is read (with FE equal to one) followed by a read of the SCDR.

When no stop bit is detected in the received data character, the framing error (FE) bit is set. FE is set at the same time as the RDRF. If the byte received causes both framing and overrun errors, the processor only recognizes the overrun error. The framing error flag inhibits further transfer of data into the SCDR until it is cleared. The FE bit is cleared when the SCSR is read (with FE equal to one) followed by a read of the SCDR.

7.6 SCI Registers

There are five addressable registers associated with the SCI. SCCR1, SCCR2, and BAUD are control registers. SCDR is the SCI data register and SCSR is the SCI status register. Refer to the BAUD register description as well as the block diagram for the baud rate generator.

7.6.1 Serial Communications Data Register

SCDR is a parallel register that performs two functions. It is the receive data register when it is read, and the transmit data register when it is written. Reads access the receive data buffer and writes access the transmit data buffer. Receive and transmit are double buffered.

SCDR — SCI Data Register **$102F**

	Bit 7	6	5	4	3	2	1	Bit 0
	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
RESET:	I	I	I	I	I	I	I	I

7.6.2 Serial Communications Control Register 1

The SCCR1 register provides the control bits that determine word length and select the method used for the wakeup feature.

SCCR1 — SCI Control Register 1 **$102C**

	Bit 7	6	5	4	3	2	1	Bit 0
	R8	T8	—	M	WAKE	—	—	—
RESET:	I	I	0	0	0	0	0	0

RCKB — SCI Baud Rate Clock Check (Test)

SCR[2:0] — SCI Baud Rate Selects

Selects receiver and transmitter bit rate based on output from baud rate prescaler stage. Refer to the SCI baud rate generator block diagram.

Table 7-2 Baud Rate Selection

SCR[2:0]	Divide Prescaler By	Highest Baud Rate (Prescaler Output from Previous Table)			
		4800	19200	76800	312500
0 0 0	1	4800	19200	76800	312500
0 0 1	2	2400	9600	38400	156250
0 1 0	4	1200	4800	19200	78125
0 1 1	8	600	2400	9600	39063
1 0 0	16	300	1200	4800	19531
1 0 1	32	150	600	2400	9766
1 1 0	64	75	300	1200	4883
1 1 1	128	—	150	600	2441

The prescaler bits, SCP[2:0], determine the highest baud rate, and the SCR[2:0] bits select an additional binary submultiple (≥1, ≥2, ≥4, through ≥128) of this highest baud rate. The result of these two dividers in series is the 16X receiver baud rate clock. The SCR[2:0] bits are not affected by reset and can be changed at any time, although they should not be changed when any SCI transfer is in progress.

Figure 7-3 and **Figure 7-4** illustrate the SCI baud rate timing chain. The prescaler select bits determine the highest baud rate. The rate select bits determine additional divide by two stages to arrive at the receiver timing (RT) clock rate. The baud rate clock is the result of dividing the RT clock by 16.

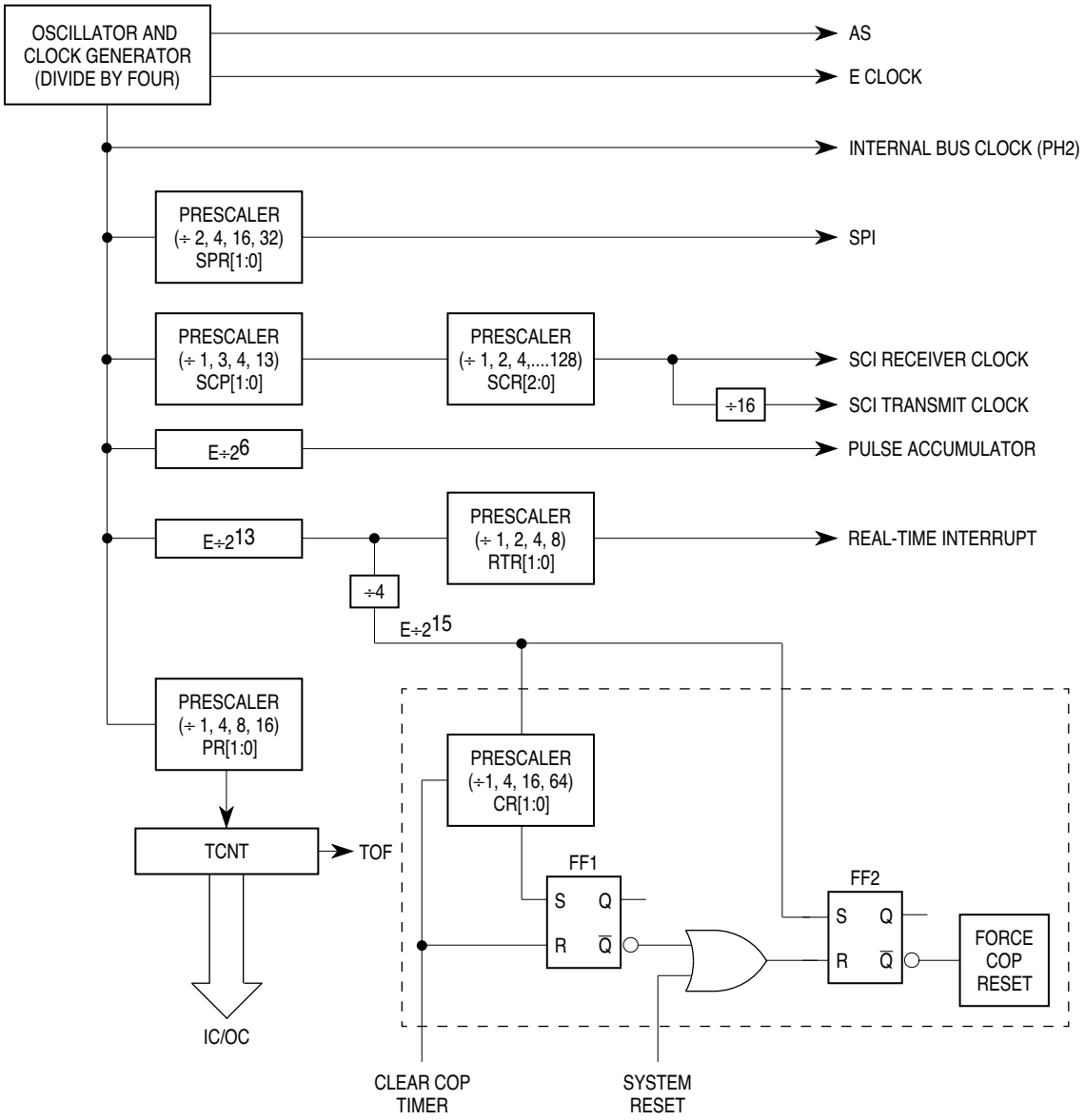

Figure 9-1 Timer Clock Divider Chains

9.3.8 Timer Interrupt Flag Register 1

Bits in this register indicate when timer system events have occurred. Coupled with the bits of TMSK1, the bits of TFLG1 allow the timer subsystem to operate in either a polled or interrupt driven system. Each bit of TFLG1 corresponds to a bit in TMSK1 in the same position.

TFLG1 — Timer Interrupt Flag 1 **$1023**

	Bit 7	6	5	4	3	2	1	Bit 0
	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F
RESET:	0	0	0	0	0	0	0	0

Clear flags by writing a one to the corresponding bit position(s).

OC1F–OC4F — Output Compare x Flag
Set each time the counter matches output compare x value

I4/O5F — Input Capture 4/Output Compare 5 Flag
Set by IC4 or OC5, depending on the function enabled by I4/O5 bit in PACTL

IC1F–IC3F — Input Capture x Flag
Set each time a selected active edge is detected on the ICx input line

9.3.9 Timer Interrupt Mask Register 2

Use this 8-bit register to enable or inhibit timer overflow and real-time interrupts. The timer prescaler control bits are included in this register.

TMSK2 — Timer Interrupt Mask 2 **$1024**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOI	RTII	PAOVI	PAII	—	—	PR1	PR0
RESET:	0	0	0	0	0	0	0	0

TOI — Timer Overflow Interrupt Enable
0 = TOF interrupts disabled
1 = Interrupt requested when TOF is set to one

RTII — Real-Time Interrupt Enable
Refer to **9.4 Real-Time Interrupt**.

PAOVI — Pulse Accumulator Overflow Interrupt Enable
Refer to **9.6.3 Pulse Accumulator Status and Interrupt Bits**.

PAII — Pulse Accumulator Input Edge Interrupt Enable
Refer to **9.6.3 Pulse Accumulator Status and Interrupt Bits**.

PR[1:0] — Timer Prescaler Select
These bits are used to select the prescaler divide-by ratio. In normal modes, PR[1:0] can only be written once, and the write must be within 64 cycles after reset. Refer to **Table 9-1** and **Table 9-4** for specific timing values.

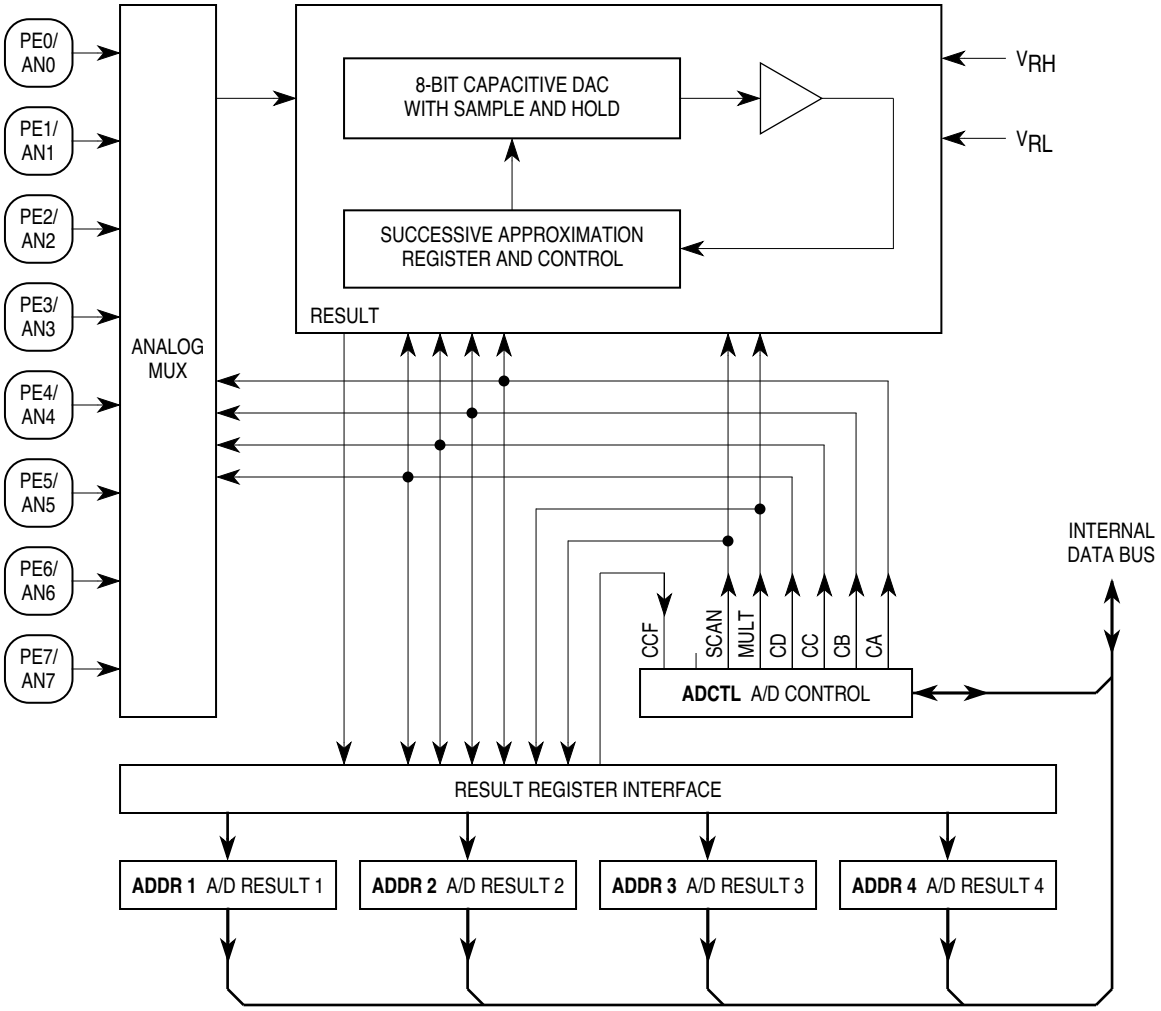

Figure 10-1 A/D Converter Block Diagram

Port E pins can also be used as digital inputs. Reads of port E pins are not recommended during the sample portion of an A/D conversion cycle, when the gate signal to the N-channel input gate is on. Because no P-channel devices are directly connected to either input pins or reference voltage pins, voltages above V_{DD} do not cause a latchup problem, although current should be limited according to maximum ratings. Refer to **Figure 10-2**, which is a functional diagram of an input pin.

Table A-5 Peripheral Port Timing

V_{DD} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H

Characteristic	Symbol	2.0 MHz		3.0 MHz		4.0 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation (E-Clock Frequency)	f_o	dc	2.0	dc	3.0	dc	4.0	MHz
E-Clock Period	t_{cyc}	500	—	333	—	250	—	ns
Peripheral Data Setup Time (MCU Read of Ports A, C, D, E, G)	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral Data Hold Time (MCU Read of Ports A, C, D, E, G)	t_{PDH}	50	—	50	—	50	—	ns
Delay Time, Peripheral Data Write (MCU Write to Port A) (MCU Write to Ports B, C, D, F, and G t_{PWD} = 1/4 t_{cyc} + 100 ns)	t_{PWD}	— —	200 225	— —	200 183	— —	200 162	ns

NOTES:
1. Ports C, D, and G timing is valid for active drive (CWOM, DWOM, and GWOM bits **cleared**).
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.

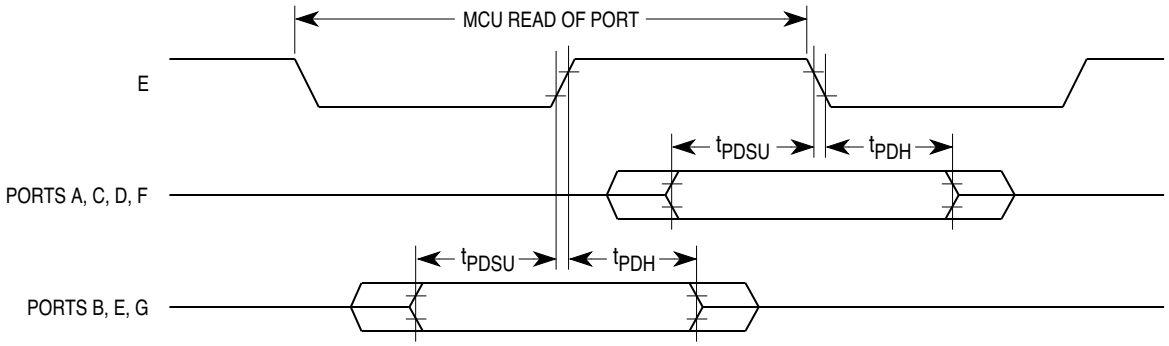

Figure A-7 Port Read Timing Diagram

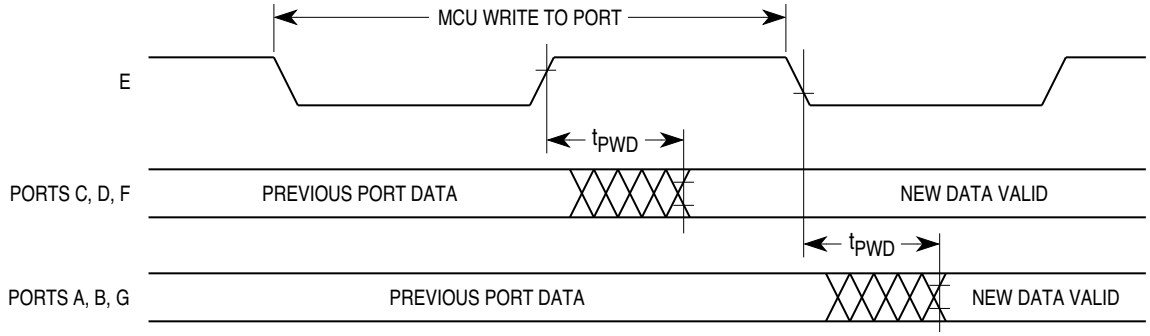

Figure A-8 Port Write Timing Diagram

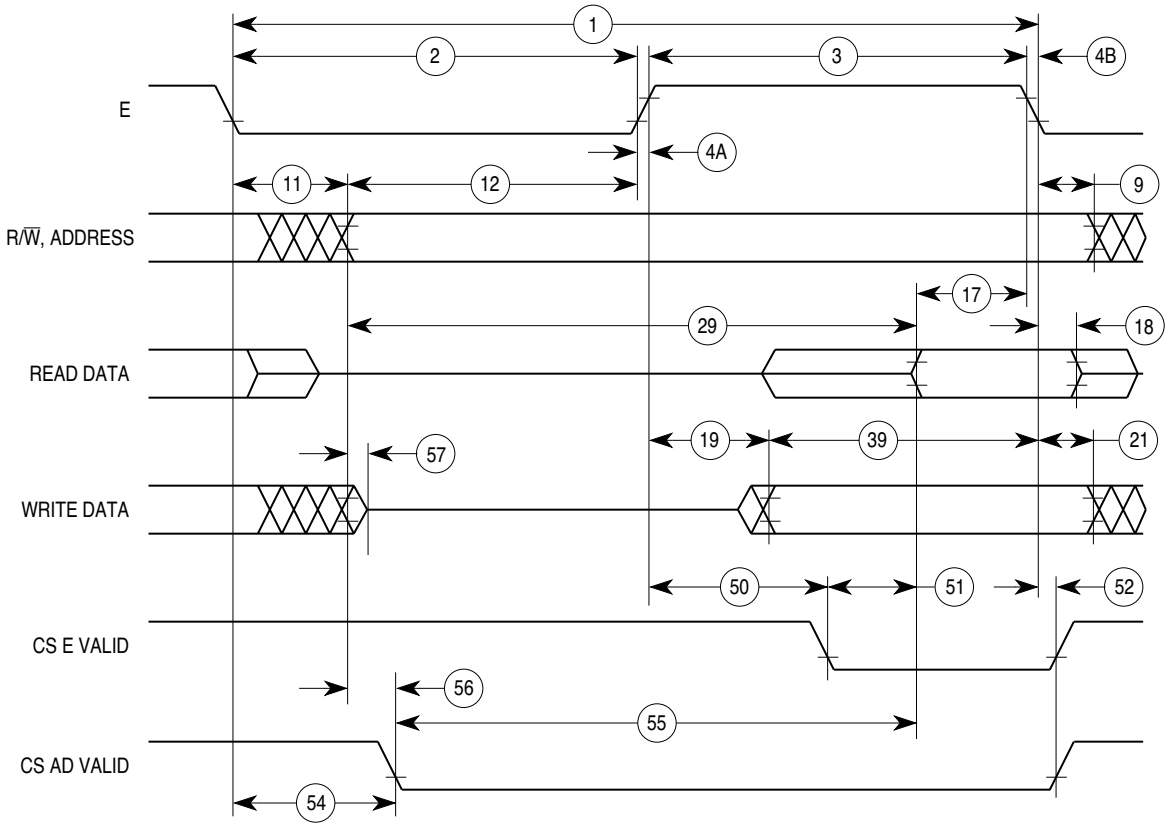

Figure A-9 Expansion Bus Timing

Table A-9 EEPROM Characteristics

V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H

Characteristic		Temperature Range			Unit
		0 to 70, – 40 to 85	**–40 to 105**	**–40 to 125**	**° C**
Programming Time (Note 1)	<1.0 MHz, RCO Enabled	10	15	20	ms
	1.0 to 2.0 MHz, RCO Disabled	20	Must use RCO	Must use RCO	
	≥ 2.0 MHz (or Anytime RCO Enabled)	10	15	20	
Erase Time (Note 1)	Byte, Row and Bulk	10	10	10	ms
Write/Erase Endurance (Note 2)		10,000	10,000	10,000	Cycles
Data Retention (Note 2)		10	10	10	Years

NOTES:

1. The RC oscillator (RCO) must be enabled (by setting the CSEL bit in the OPTION register) for EEPROM programming and erasure when the E-clock frequency is below 1.0 MHz.
2. Refer to Reliability Monitor Report (current quarterly issue) for current failure rate information.

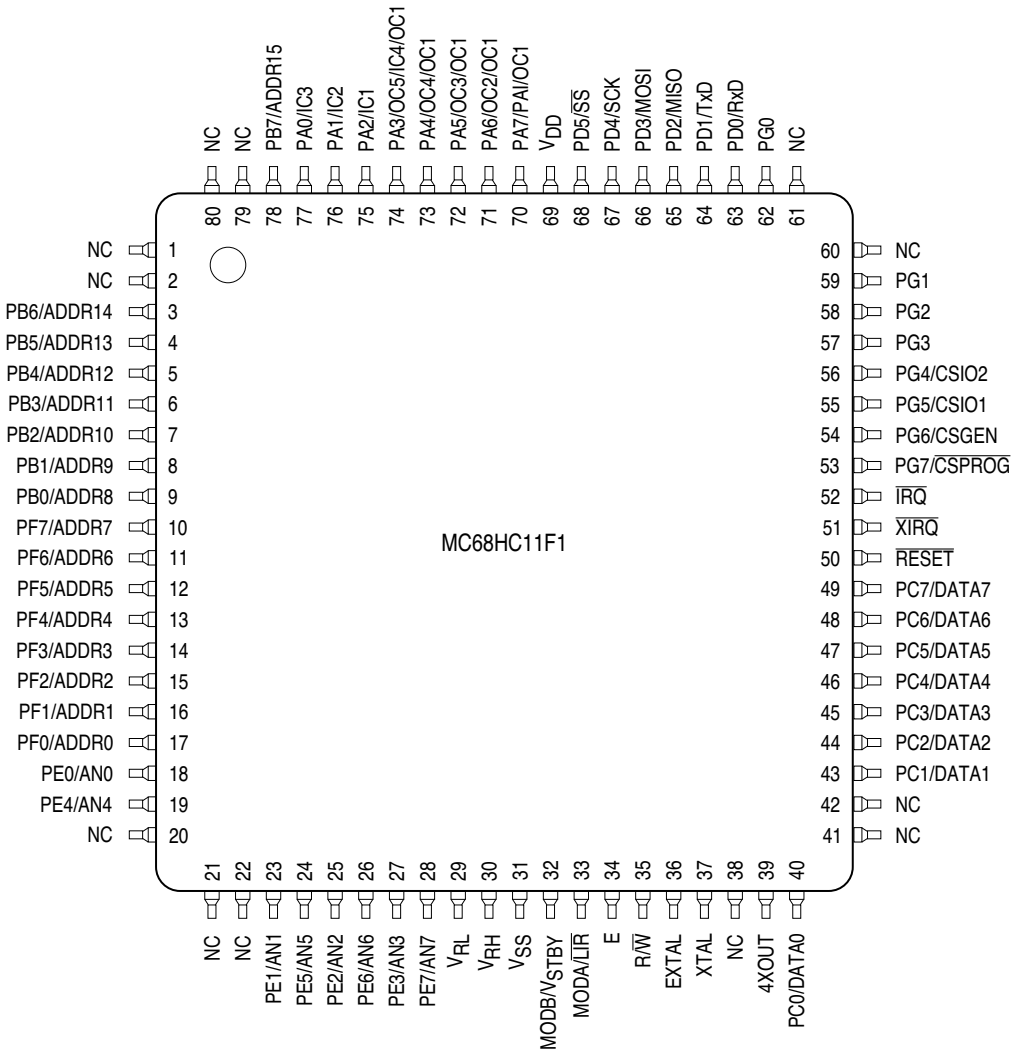

Figure B-2 MC68HC11F1 80-Pin Quad Flat Pack

B.2 Package Dimensions

For case outlines please visit our website at http://design-net.sps.mot.com.