68HC05CL48
68HC705CL48
68HC05CL16
68HC705CL16

SPECIFICATION
(General Release)

Hong Kong Design Center
CSIC Group

TABLE OF CONTENTS

Section **Page**

SECTION 1
GENERAL DESCRIPTION

SECTION 2
MEMORY

TABLE OF CONTENTS

Section **Page**

TABLE OF CONTENTS

Section **Page**

SECTION 8
TIMER

SECTION 9
SERIAL PERIPHERAL INTERFACE

SECTION 10
ANALOG TO DIGITAL CONVERTER

TABLE OF CONTENTS

Section **Page**

TABLE OF CONTENTS

Section **Page**

SECTION 14
INSTRUCTION SET

SECTION 15
ELECTRICAL SPECIFICATIONS

SECTION 16
MECHANICAL SPECIFICATIONS

APPENDIX A
MC68HC705CL48

TABLE OF CONTENTS

Section **Page**

APPENDIX B
MC68HC05CL16

APPENDIX C
MC68HC705CL16

LIST OF FIGURES

LIST OF FIGURES

Figure **Title** **Page**

LIST OF TABLES

LIST OF TABLES

SECTION 1
GENERAL DESCRIPTION

The MC68HC05CL48 HCMOS Microcontroller is a member of the M68HC05 family of low cost single chip microcontrollers. It is particularly suitable as a Caller ID telephone controller. This 8-bit microcontroller unit (MCU) contains on-chip oscillator, CPU, RAM, ROM, I/O, Timer, Watchdog Timer, SPI, A/D, Caller ID subsystem, LCD driver and a 32kHz PLL.

1.1 FEATURES

- Industry standard M68HC05 8-bit CPU core
- 32kHz PLL to generate the 3.6MHz system clock
- On-chip clock generator can be driven by crystal or external clock
- 47.5k-bytes of user ROM
- 2k-bytes of user RAM (64 bytes for stack)
- 16-bit Timer with 2 input captures and 2 output compares, the two output compare are used for interrupt generation only
- 15-stage Multi-function Timer
- One second Watch Timer
- COP (Computer Operating Properly) Watchdog Reset
- Serial Peripheral Interface (SPI)
- Four channel 8-bit A/D Converter
- Hardware Caller-ID subsystem
- 45 x 16 or 53 x 8 LCD driver
- 24 Bidirectional I/O lines
- Three software programmable external interrupts in addition to the standard $\overline{IRQ}$ interrupt.

- Five keyboard interrupts.
- Power saving STOP and WAIT modes
- Available in 112-pin TQFP package

NOTE

A line over a signal name indicates an active low signal. Any reference to voltage, current, or frequency specified in the following sections will refer to the nominal values. The exact values and their tolerance or limits are specified in **Section 15**.

1.2 MCU STRUCTURE

The block diagram of the MC68HC05CL48 is shown in **Figure 1-1**

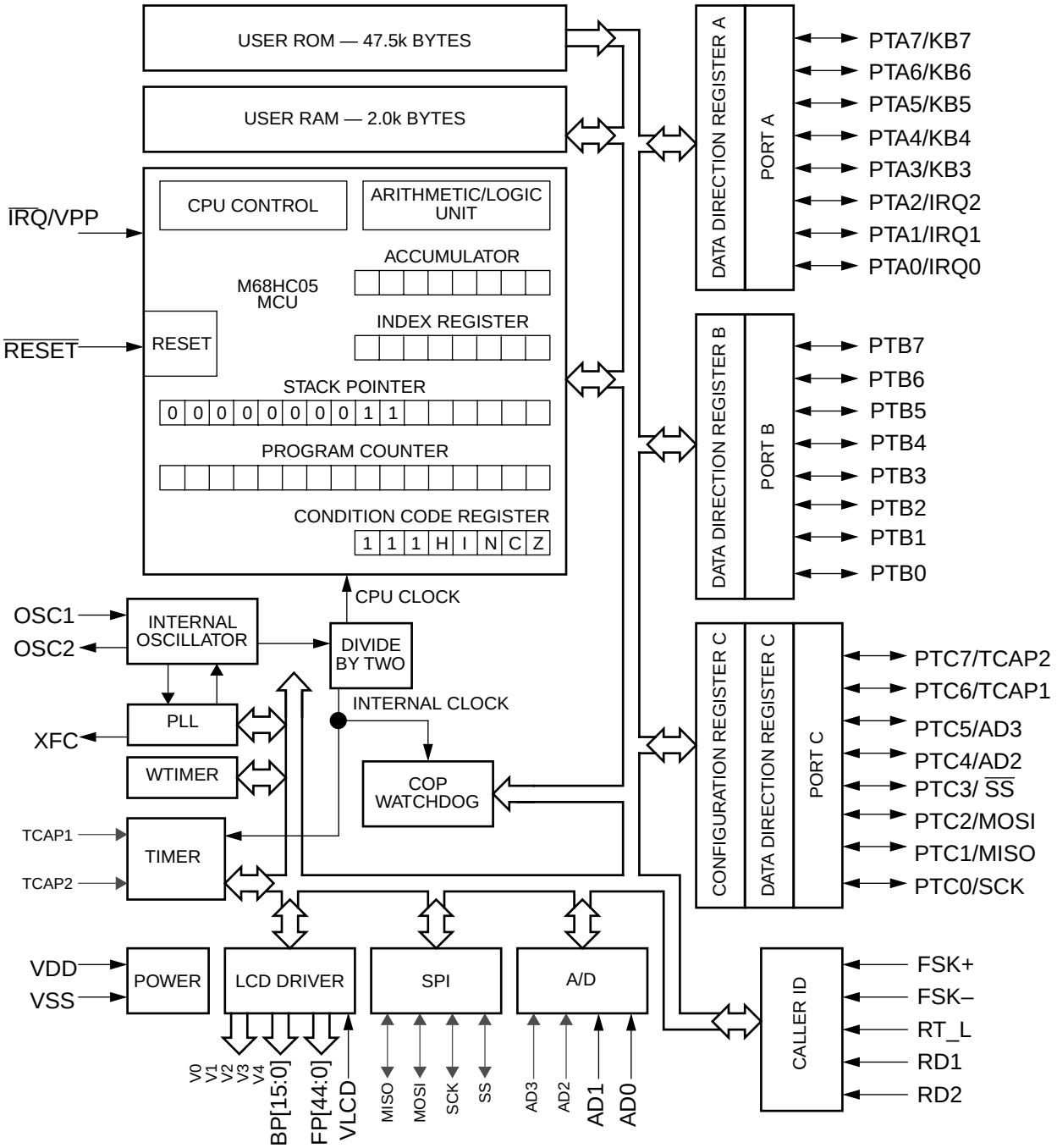
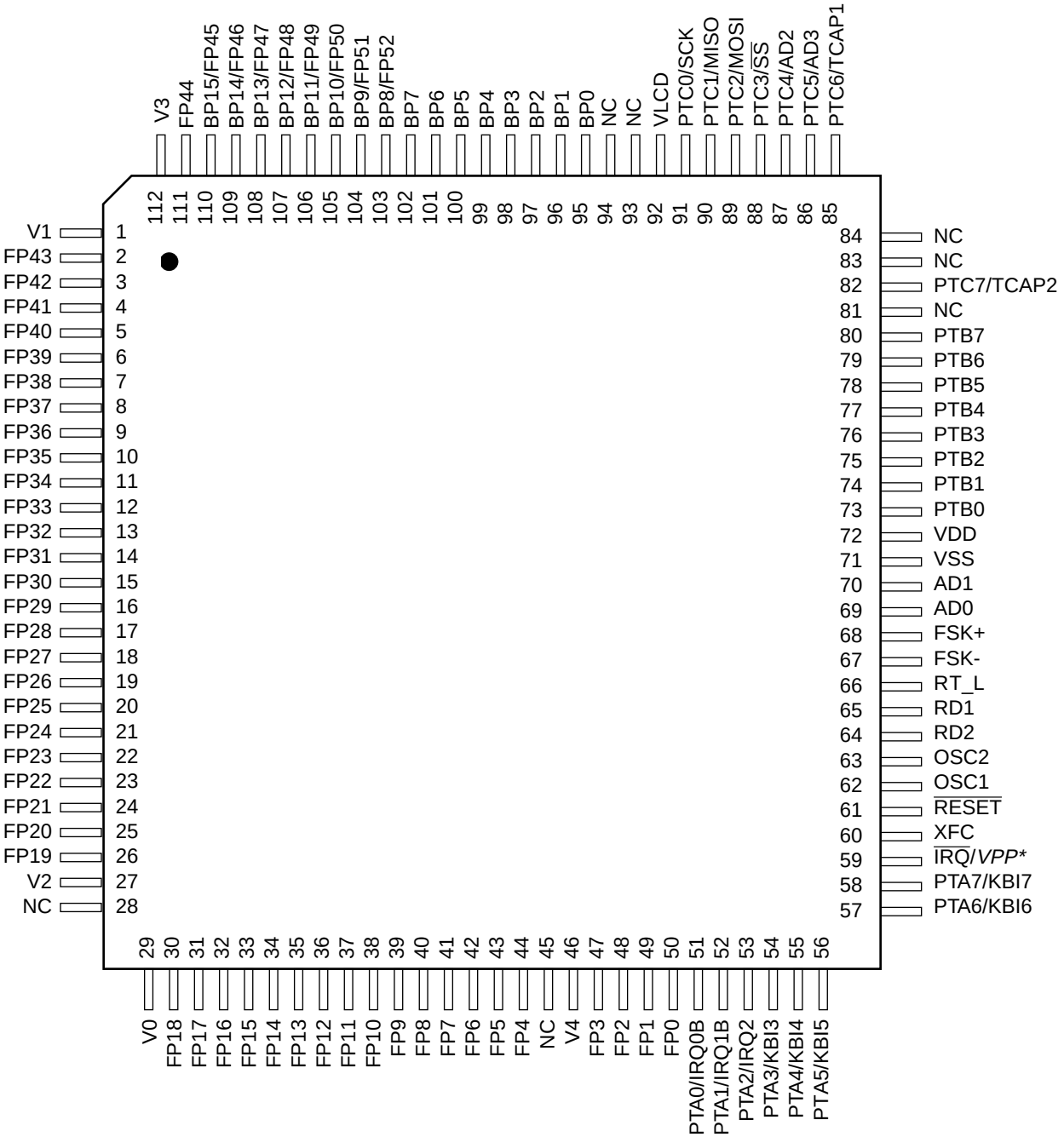


Figure 1-1. MC68HC05CL48 Block Diagram

1.3 PIN ASSIGNMENT

The MC68HC05CL48 pin assignments for the 112-pin TQFP package is shown in **Figure 1-2**.



**VPP is available on MC68HC705CL48 only*

Figure 1-2. MC68HC05CL48 Pin Assignments

1.4 SIGNAL DESCRIPTION

The following paragraphs give a description of the general function of each pin.

1.4.1 V_{DD} and V_{SS}

Power is supplied to the MCU through V_{DD} and V_{SS} pins. V_{DD} is connected to a regulated positive supply and V_{SS} is connected to ground. These pins are located close to each other for low EMI, Electro Magnetic Interference, emission.

Very fast signal transitions occur on the MCU pins. The short rise and fall times place very high short-duration current demands on the power supply. To prevent noise problems, take special care to provide good power supply bypassing at the MCU. Use bypass capacitors with good high-frequency characteristics, and position them as close to the MCU as possible. Bypassing requirements vary, depending on how heavily the MCU pins are loaded.

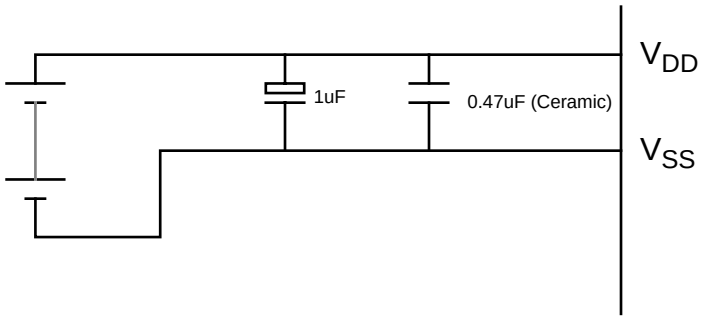


Figure 1-3. Power Supply Decoupling

1.4.2 $\overline{\text{RESET}}$

This pin can be used as an input to reset the MCU to a known start-up state by pulling it to the low state. The $\overline{\text{RESET}}$ pin contains a steering diode to discharge any voltage on the pin to VDD, when the power is removed. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger to improve its noise immunity as an input. The $\overline{\text{RESET}}$ pin has an internal pulldown device that pulls the $\overline{\text{RESET}}$ pin low when there is an internal COP Watchdog reset or during the power-on reset cycles. Refer to **Section 5**.

1.4.3 $\overline{\text{IRQ}}$

This pin has an option in the option register that provides two different choices of interrupt triggering sensitivity. The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. Refer to **Section 4**.

1.4.4 $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, IRQ2

These three pins provide additional sources of interrupt. They are all edge-triggered only. $\overline{\text{IRQ0}}$ and $\overline{\text{IRQ1}}$ are both negative-edge triggered only and IRQ2 can be triggered on both rising and falling edges. Refer to **Section 4.3.2**. The $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$ and IRQ2 pins contain internal Schmitt trigger as part of its input to improve noise immunity.

1.4.5 OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator or an external signal to these pins provides the clock to the on chip PLL.

The OSC1 and OSC2 pins can accept the following:

1. A crystal as shown in **Figure 1-4(a)**
2. A ceramic resonator as shown in **Figure 1-4(a)**
3. An external clock signal as shown in **Figure 1-4(b)**

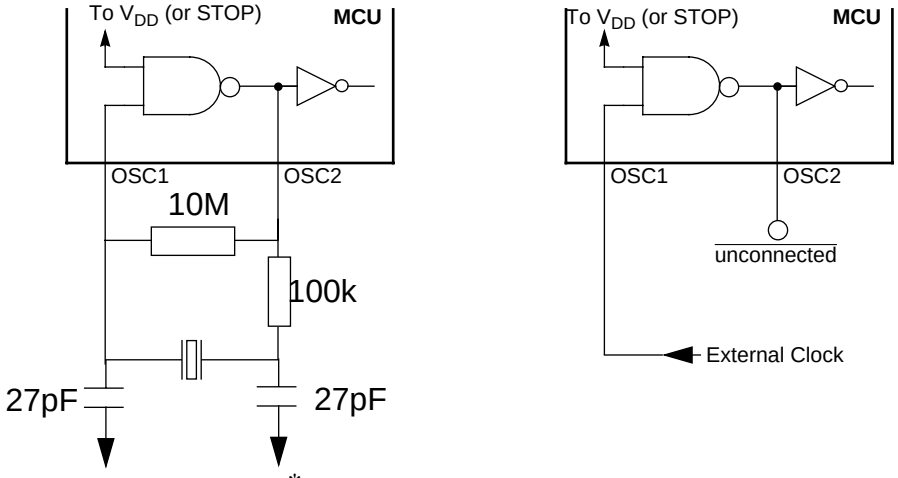
The system clock can either be the 32kHz external clock or a higher frequency clock from the PLL. See **Section 13**.

1.4.5.1 Crystal with no Internal Components

The circuit in **Figure 1-4(a)** shows a typical oscillator circuit for an AT-cut, parallel resonant crystal. Follow the crystal manufacturer's recommendations, as the crystal parameters determine the external component values required to provide maximum stability and reliable start-up. The load capacitance values used in the oscillator circuit design should include all stray capacitance. Mount the crystal and components as close as possible to the pins for start-up stabilization and to minimize output distortion.

1.4.5.2 Ceramic Resonator

In cost-sensitive applications, use a ceramic resonator in place of a crystal. Use the circuit in **Figure 1-4(a)** for a ceramic resonator and follow the resonator manufacturer's recommendations, as the resonator parameters determine the external component values required for maximum stability and reliable starting. The load capacitance values used in the oscillator circuit design should include all stray capacitance. Mount the resonator and components as close as possible to the pins for start-up stabilization and to minimize output distortion



(a) Crystal or Ceramic* Resonator Connections

X-tal = Nitsuko EX-0228-0150-NTK

(b) External Clock Source Connections

*: Additional capacitance may be required for Ceramic Resonator option. Follow the Ceramic Resonator manufacturer's recommendations.

Figure 1-4. Oscillator Connections

1.4.6 External Clock

An external clock from another CMOS-compatible device can be connected to the OSC1 input, with the OSC2 input not connected, as shown in **Figure 1-4 (b)**

1.4.7 PTA0-PTA7

These I/O lines comprise Port A. The state of any pin is software programmable and all port lines are configured as inputs during power-on or reset. PTA0-PTA7 are also associated with three external IRQ and five Keyboard interrupt functions. See **Section 7** for more details on the I/O ports.

1.4.8 PTB0-PTB7

These I/O lines comprise Port B. The state of any pin is software programmable and all port lines are configured as inputs during power-on or reset.

1.4.9 PTC0-PTC7

These I/O lines comprise Port C. The state of any pin is software programmable and all port lines are configured as inputs during power-on or reset. PTC0-PTC7

when not used as I/O pins can be configured as two A-to-D pins, two Timer pins and four SPI pins. See **Section 7** for more details on the I/O ports.

1.4.10 SCK

This is the serial clock signal of the SPI. This pin can be configured as PTC0 pin when the SPI function is not used.

1.4.11 MISO, MOSI

These are the master-in slave-out (MISO) and master-out slave-in (MOSI) data pins for the SPI. These two pins can be configured as PTC1 and PTC2 pins respectively when the SPI function is not used.

1.4.12 $\overline{SS}$

This is the slave select pin for the SPI. This pin can be configured as PTC3 pin when the SPI function is not used.

1.4.13 AD0-AD3

These four lines are the A/D inputs. AD2 and AD3 can be configured as PTC4 and PTC5 pins respectively when not used as A-to-D inputs.

1.4.14 TCAP1

The TCAP1 input controls the input capture 1 feature of the on-chip programmable timer system. Refer to **Section 8** for additional information. This pin can be configured as PTC6 pin when the timer input capture function is not used.

1.4.15 TCAP2

The TCAP2 input controls the input capture 2 feature of the on-chip programmable timer system. Refer to **Section 8** for additional information. This pin can be configured as PTC7 pin when the timer input capture function is not used.

1.4.16 FSK+, FSK–

These two inputs are the non-inverting and inverting FSK signals respectively.

1.4.17 RT_L

This pin should be connected with an external pull-up resistor to vDD. This pin will be pulled low when RDI1 rises above one volt.

1.4.18 RD1, RD2

These inputs are incoming ring qualifiers.

1.4.19 BP0 - BP15

These are the back plane drivers dedicated to the LCD subsystem. **BP8 - BP15** will be used as **FP45 - FP52** when the LCD driver is only driving 8 back planes.

1.4.20 FP0 - FP44

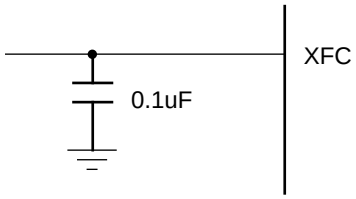
These are the front plane drivers dedicated to the LCD subsystem.

1.4.21 VLCD, V0, V1, V2, V3, V4

VLCD is the power supply input for the LCD voltage generator. V0, V1, V2, V3 and V4 are the internal resistor ladder node voltage. An external cap with one terminal connected to GND should be connected to each of these four pins.

1.4.22 XFC

One terminal of the Phase-Locked Loop external capacitor should be connected here, the other terminal should be connected to GND.



Phase-Locked Loop External Capacitor

SECTION 2
MEMORY

The MC68HC05CL48 has a 64k-byte memory map, consisting of user ROM, user RAM, Self-Check ROM, LCD RAM and I/O as shown in **Figure 2-1**.

2.1 MEMORY MAP

The MC68HC05CL48 memory map is shown in **Figure 2-1**.

2.2 I/O AND CONTROL REGISTERS

The I/O and Control Registers reside in locations $0000-$002F. The overall organization of these registers is shown in **Figure 2-1**. The bit assignments for each register are shown in **Figure 2-2**, **Figure 2-3** and **Figure 2-4**. Reading from unimplemented bits will return unknown states, and writing to unimplemented bits will be ignored.

2.2.1 Option Register ($1F)

This register can be written to only once following a Power-On-Reset (POR) but can be read at any time. This register configures the MCU options to suit the user's application.

		7	6	5	4	3	2	1	0
OPT $001F	READ	OSC_ON	COP					INTO	
	WRITE								
	POR	1	1	U	U	U	U	1	U

U = UNAFFECTED

OSC_ON

1 = Oscillator is switched on in STOP mode.
0 = Oscillator is switched off in STOP mode.

COP

Refer to **Section 8** for detailed informations on COP.

1 = Enable the COP watchdog reset.
0 = Disable the COP watchdog reset.

INTO

1 = allow negative edge-triggered interrupt on $\overline{IRQ}$ pin.

0 = allow 'low' level-triggered and negative edge-triggered interrupt on $\overline{IRQ}$ pin.

2.3 LCD RAM

The LCD RAM consists of 90 bytes at locations ($0930 thru $095C) and ($0A30 thru $0A5C).

2.4 RAM

The total RAM consists of 2.0k bytes (including the stack) at locations $0030 thru $082F. The stack begins at address $00FF and proceeds down to $00C0. The stack pointer can access 64 locations from $00FF to $00C0. Using the stack area for data storage or temporary work locations requires care to prevent it from being over written due to stacking from an interrupt or subroutine call.

2.5 ROM

There are a total of 47.5k bytes of user ROM from locations $4000 thru $FDFF for user program storage and 16 bytes for user vectors at locations $FFF0 thru $FFFF.

2.6 I/O MAPPED REGISTERS

There are 48 internal registers at locations $0000 thru $002F. These are illustrated in **Figure 2-2**, **Figure 2-3** and **Figure 2-4**.

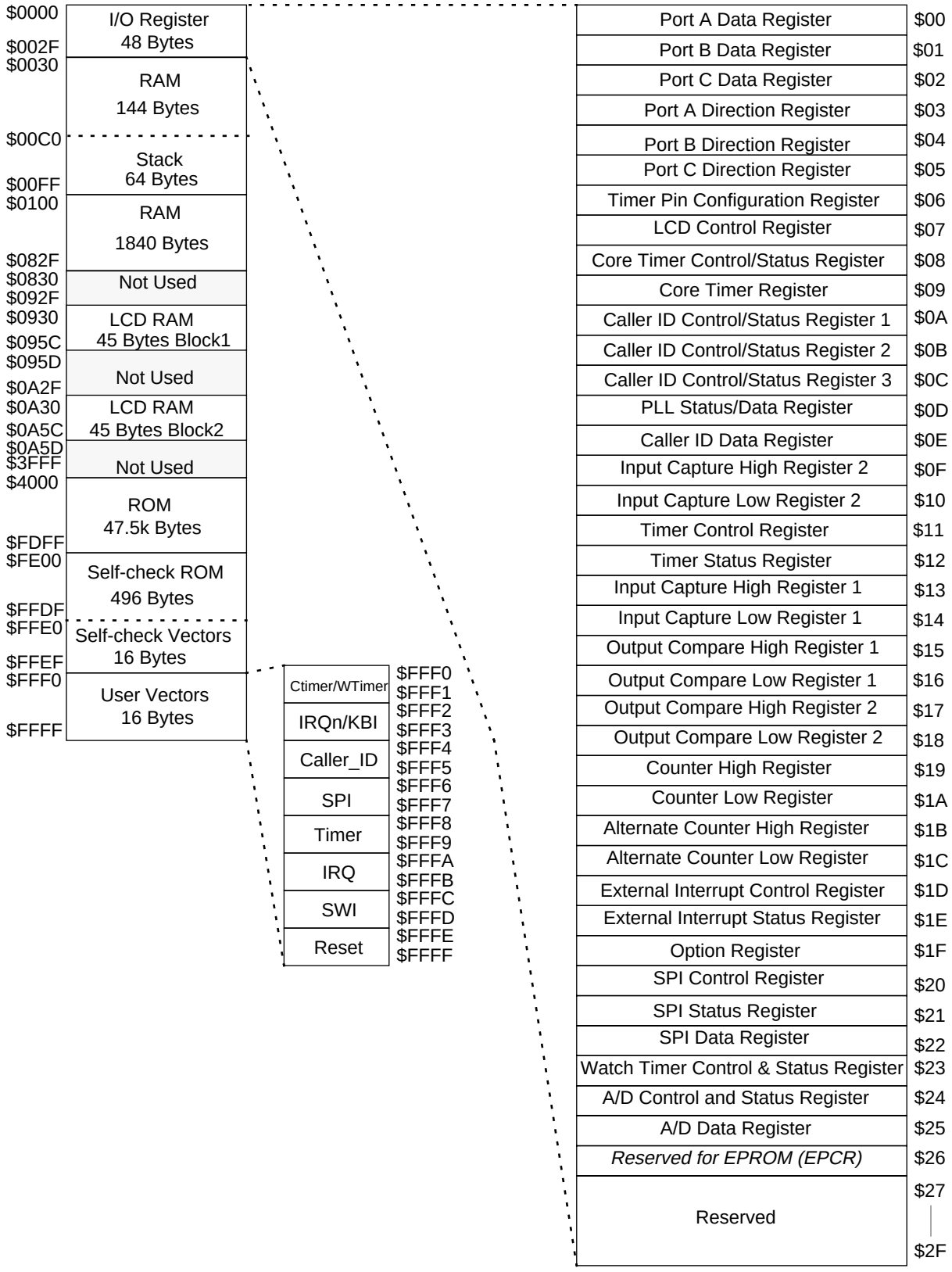


Figure 2-1. MC68HC05CL48 Memory Map

ADDR	REGISTER	ACCESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
$0000	PORTA	R	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		W								
$0001	PORTB	R	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		W								
$0002	PORTC	R	PTC7	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
		W								
$0003	DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
$0004	DDRB	R	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		W								
$0005	DDRC	R	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		W								
$0006	TIMCONF	R	CONF7	CONF6						
		W								
$0007	LCDCR	R	CC3	CC2	CC1	CC0	MX8	BIAS5		DISON
		W								
$0008	CTCSR	R	CTOF	RTIF	CTOFE	RTIE	0	0	RT1	RT0
		W								
$0009	CTCR	R	CT7	CT6	CT5	CT4	CT3	CT2	CT1	CT0
		W								
$000A	CLCSR1	R	RDIF	RDIE	CDIF	CDIE			RDO	CDO
		W								
$000B	CLCSR2	R		RDPW	CDPW			CIDSD	RD	CD
		W								
$000C	CLCSR3	R	SDSL			CDRE	DRIE	CDRF	RDOE	CDOE
		W								
$000D	PCSR	R	PON	PCLK				FREQ1	FREQ0	
		W								
$000E	CDDR	R	CDD7	CDD6	CDD5	CDD4	CDD3	CDD2	CDD1	CDD0
		W								
$000F	IC2H	R	IC2:15	IC2:14	IC2:13	IC2:12	IC2:11	IC2:10	IC2:9	IC2:8
		W								

Figure 2-2. MC68HC05CL48 I/O Registers $0000-$000F

ADDR	REGISTER	ACCESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
$0010	IC2L	R	IC2:7	IC2:6	IC2:5	IC2:4	IC2:3	IC2:2	IC2:1	IC2:0
		W								
$0011	TCR	R	IC1IE	IC2IE	OC1IE	OC2IE	TOIE		IEDG2	IEDG1
		W								
$0012	TSR	R	IC1F	IC2F	OC1F	OC2F	TOF			
		W								
$0013	IC1H	R	IC1:15	IC1:14	IC1:13	IC1:12	IC1:11	IC1:10	IC1:9	IC1:8
		W								
$0014	IC1L	R	IC1:7	IC1:6	IC1:5	IC1:4	IC1:3	IC1:2	IC1:1	IC1:0
		W								
$0015	OC1H	R	OC1:15	OC1:14	OC1:13	OC1:12	OC1:11	OC1:10	OC1:9	OC1:8
		W								
$0016	OC1L	R	OC1:7	OC1:6	OC1:5	OC1:4	OC1:3	OC1:2	OC1:1	OC1:0
		W								
$0017	OC2H	R	OC2:15	OC14	OC2:13	OC2:12	OC2:11	OC2:10	OC2:9	OC2:8
		W								
$0018	OC2L	R	OC2:7	OC2:6	OC2:5	OC4	OC2:3	OC2:2	OC2:1	OC2:0
		W								
$0019	TCH	R	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
		W								
$001A	TCL	R	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
		W	TIMER RESET							
$001B	ACH	R	AC15	AC14	AC13	AC12	AC11	AC10	AC9	AC8
		W								
$001C	ACL	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		W	TIMER RESET							
$001D	EXICR	R	KBE7	KBE6	KBE5	KBE4	KBE3	IRQE2	IRQE1	IRQE0
		W								
$001E	EXISR	R	KBIF7	KBIF6	KBIF5	KBIF4	KBIF3	IRQF2	IRQF1	IRQF0
		W								
$001F	OPT	R	OSC_ON	COP					INTO	
		W								

Figure 2-3. MC68HC05CL48 I/O Registers $0010-$001F

ADDR	REGISTER	ACCESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
$0020	SPCR	R	SPIE	SPE	-	MSTR	CPOL	CPHA	SPR1	SPR0
		W								
$0021	SPSR	R	SPIF	WCOL	0	MODF	0	0	0	0
		W								
$0022	SPDR	R	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
		W								
$0023	WTCSR	R					WTOFF	0	WTOIE	WTOF
		W						WTR		
$0024	ADSCR	R	COCO	ADRC	ADON	0		CH2	CH1	CH0
		W								
$0025	ADDR	R	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
		W								
$0026	Reserved for EPROM (EPCR)	R								
		W								
$0027	Not Used	R								
		W								
$0028	Not Used	R								
		W								
$0029	Not Used	R								
		W								
$002A	Not Used	R								
		W								
$002B	Not Used	R								
		W								
$002C	Not Used	R								
		W								
$002D	Not Used	R								
		W								
$002E	Reserved for TEST	R								
		W								
$002F	TEST	R								
		W								

Figure 2-4. MC68HC05CL48 I/O Registers $0020-$002F

SECTION 3
CPU

The MC68HC05CL48 has an 64k memory map. Therefore it uses all 16 bits of the address bus. The stack has only 64 bytes. Therefore, the stack pointer has been reduced to only 6 bits and will only decrement down to $00C0 and then wrap-around to $00FF. All other instructions and registers behave as described in this chapter.

3.1 REGISTERS

The MCU contains five registers which are hard-wired within the CPU and are not part of the memory map. These five registers are shown in **Figure 3-1** and are described in the following paragraphs.

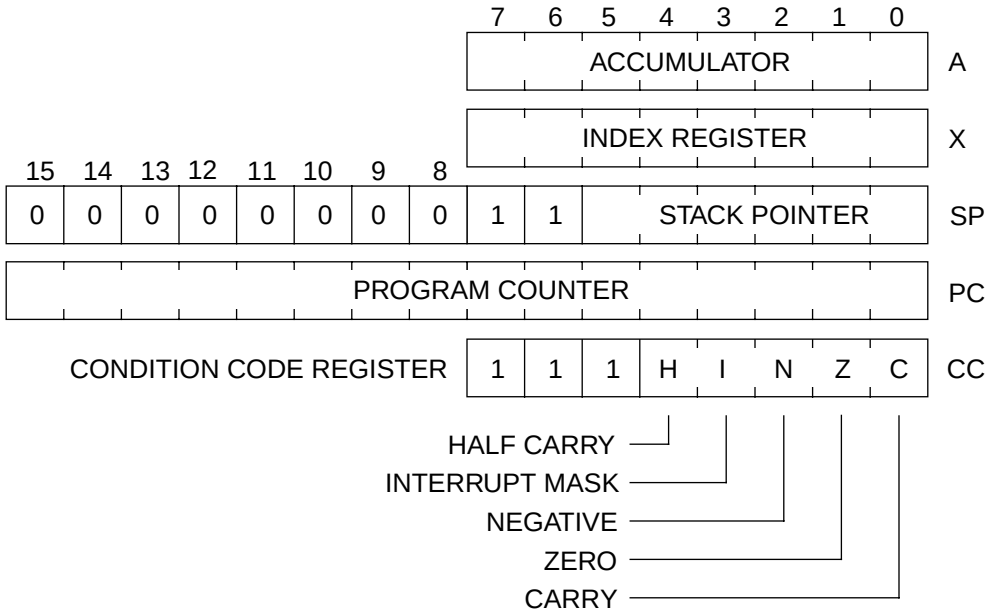


Figure 3-1. MC68HC05 Programming Model

3.1.1 Accumulator (A)

The accumulator is a general purpose 8-bit register as shown in **Figure 3-1**. The CPU uses the accumulator to hold operands and results of arithmetic calculations

or non-arithmetic operations. The accumulator is unaffected by a reset of the device.

3.1.2 Index Register (X)

The index register shown in **Figure 3-1** is an 8-bit register that can perform two functions:

- Indexed addressing
- Temporary storage

In indexed addressing with no offset, the index register contains the low byte of the operand address, and the high byte is assumed to be $00. In indexed addressing with an 8-bit offset, the CPU finds the operand address by adding the index register contents to an 8-bit immediate value. In indexed addressing with a 16-bit offset, the CPU finds the operand address by adding the index register contents to a 16-bit immediate value

The index register can also serve as an auxiliary accumulator for temporary storage.The index register is unaffected by a reset of the device.

3.1.3 Stack Pointer (SP)

The stack pointer shown in **Figure 3-1** is a 16-bit register internally. In devices with memory maps less than 64 kbytes the unimplemented upper address lines are ignored. The stack pointer contains the address of the next free location on the stack. During a reset or the reset stack pointer (RSP) instruction, the stack pointer is set to $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the ten most significant bits are permanently set to 0000000011. The six least significant register bits are appended to these ten fixed bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 ($40) locations. If 64 locations are exceeded, the stack pointer wraps around and writes over the previously stored information. A subroutine call occupies two locations on the stack, and an interrupt uses five locations.

3.1.4 Program Counter (PC)

The program counter shown in **Figure 3-1** is a 16-bit register internally. In devices with memory maps less than 64 Kbytes the unimplemented upper address lines are ignored. The program counter contains the address of the next instruction or operand to be fetched.

Normally, the address in the program counter increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

3.1.5 Condition Code Register (CCR)

The CCR shown in **Figure 3-1** is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. The fifth bit is the interrupt mask. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. The condition code register should be thought of as having three additional upper bits that are always ones. Only the interrupt mask is affected by a reset of the device. The following paragraphs explain the functions of the lower five bits of the condition code register.

3.1.5.1 Half Carry Bit (H-Bit)

When the half-carry bit is set, it means that a carry occurred between bits 3 and 4 of the accumulator during the last ADD or ADC (add with carry) operation. The half-carry bit is required for binary-coded decimal (BCD) arithmetic operations.

3.1.5.2 Interrupt Mask (I-Bit)

When the interrupt mask is set, the internal and external interrupts are disabled. Interrupts are enabled when the interrupt mask is cleared. When an interrupt occurs, the interrupt mask is automatically set after the CPU registers are saved on the stack, but before the interrupt vector is fetched. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. Normally, the interrupt is processed as soon as the interrupt mask is cleared.

A return from interrupt (RTI) instruction pulls the CPU registers from the stack, restoring the interrupt mask to its state before the interrupt was encountered. After any reset, the interrupt mask is set and can only be cleared by the Clear I-Bit (CLI), STOP, or WAIT instructions.

3.1.5.3 Negative Bit (N-Bit)

The negative bit is set when the result of the last arithmetic operation, logical operation, or data manipulation was negative. (Bit 7 of the result was a logical one.)

The negative bit can also be used to check an often-tested flag by assigning the flag to bit 7 of a register or memory location. Loading the accumulator with the contents of that register or location then sets or clears the negative bit according

to the state of the flag.

3.1.5.4 Zero Bit (Z-Bit)

The zero bit is set when the result of the last arithmetic operation, logical operation, data manipulation, or data load operation was zero.

3.1.5.5 Carry/Borrow Bit (C-Bit)

The carry/borrow bit is set when a carry out of bit 7 of the accumulator occurred during the last arithmetic operation, logical operation, or data manipulation. The carry/borrow bit is also set or cleared during bit test and branch instructions and during shifts and rotates. This bit is not set by an INC or DEC instruction.

SECTION 4
INTERRUPTS

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I-bit) to prevent additional interrupts. Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete.

If interrupts are not masked (I-bit in the CCR is clear) and the corresponding interrupt enable bit is set the processor will proceed with interrupt processing. Otherwise, the next instruction is fetched and executed. If an interrupt occurs the processor completes the current instruction, then stacks the current CPU register states, sets the I-bit to inhibit further interrupts, and finally checks the pending hardware interrupts. If more than one interrupt is pending following the stacking operation, the interrupt with the highest vector location shown in **Table 4-1** will be serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state.

Table 4-1. Vector Address for Interrupts and Reset

Interrupts	CPU Interrupt	Vector Addresses
Reset	$\overline{\text{RESET}}$	$FFFE-$FFFF
Software	SWI	$FFFC-$FFFD
External Interrupt	$\overline{\text{IRQ}}$	$FFFA-$FFFB
Timer	TCAP1,TCAP2,TCMP1,TCMP2	$FFF8-$FFF9
SPI	$\overline{\text{SPI}}$	$FFF6-$FFF7
Caller ID	CDI, RDI, CDRI, RT_L	$FFF4-$FFF5
External Interrupt/Keyboard Interrupt	$\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, IRQ2, KBI[7:3]	$FFF2-$FFF3
Core Timer/Watch Timer	Ctimer, WTimer	$FFF0-$FFF1

When an interrupt is to be processed the CPU fetches the address of the appropriate interrupt software service routine from the vector table at locations $FFF0 thru $FFFF as defined in **Table 4-1**.

An RTI instruction is used to signify when the interrupt software service routine is completed. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume at the next instruction that was to be executed when the interrupt took place. **Figure 4-1** shows the sequence of events that occur during interrupt processing.

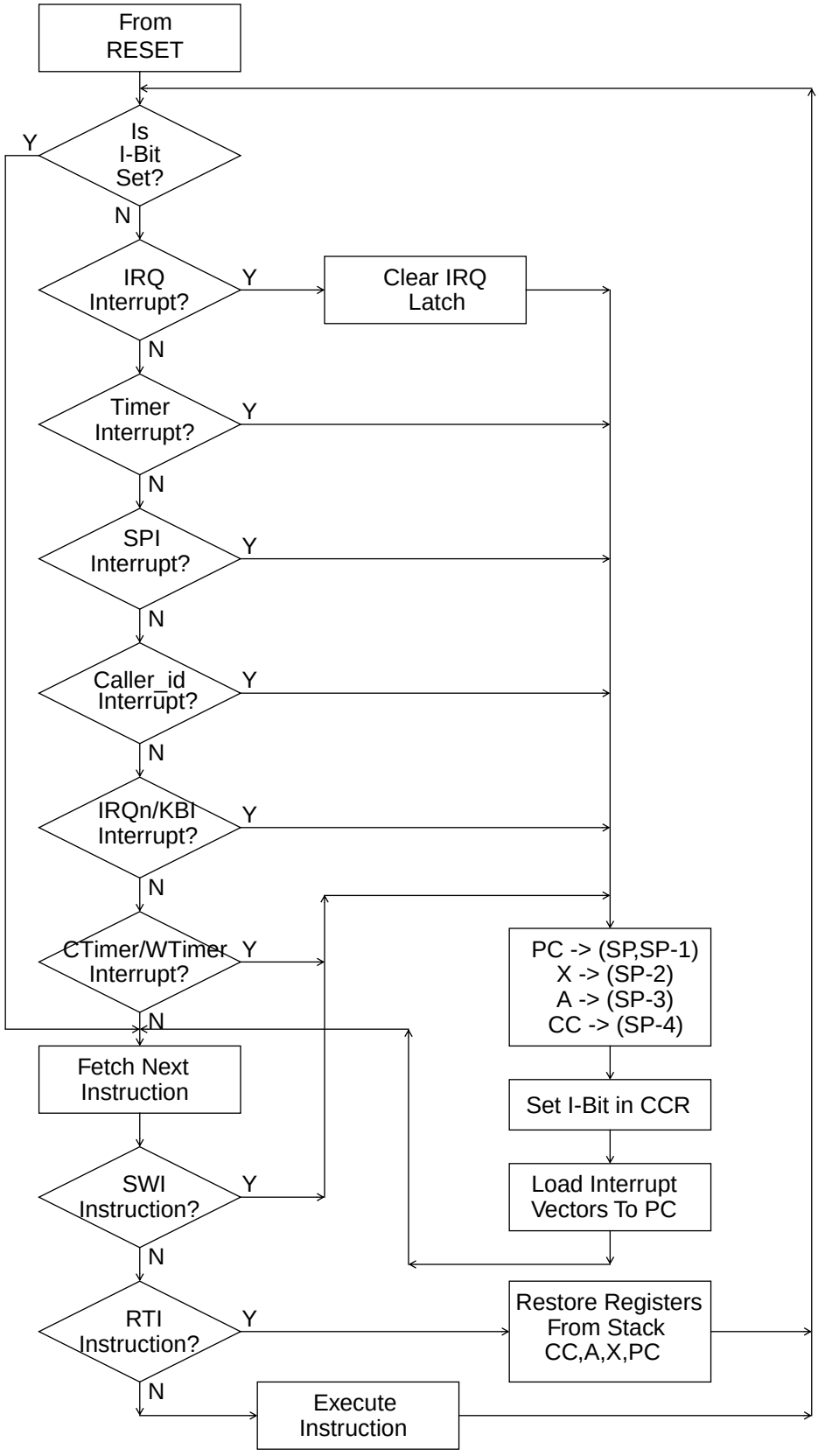


Figure 4-1. Interrupt Processing Flowchart

4.1 RESET INTERRUPT SEQUENCE

The RESET function is not in the strictest sense an interrupt; however, it is acted upon in a similar manner as shown in **Figure 4-1**. A low level input on the $\overline{\text{RESET}}$ pin or internal generated reset signal causes the program to vector to its starting address which is specified by the contents of memory locations $FFFE and $FFFF. The I-bit in the condition code register is also set. The MCU is configured to a known state during this type of reset as described in **Section 5**.

4.2 SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction and a non-maskable interrupt since it is executed regardless of the state of the I-bit in the CCR. If the I-bit is zero (interrupts enabled), the SWI instruction executes after interrupts which were pending before the SWI was fetched, or before interrupts if interrupts were generated after the SWI was fetched. The interrupt service routine address is specified by the contents of memory locations $FFFC and $FFFD.

4.3 HARDWARE INTERRUPTS

All hardware interrupts except RESET are maskable by the I-bit in the CCR. If the I-bit is set, all hardware interrupts (internal and external) are disabled. Clearing the I-bit enables the hardware interrupts. There are six types of hardware interrupts which are explained in the following sections.

4.3.1 External Interrupt ($\overline{\text{IRQ}}$)

If the IRQ option is both level and edge sensitive triggering (INTO=0), a low level or an negative edge at the $\overline{\text{IRQ}}$ pin and the interrupt mask bit of the condition code register is cleared will cause an EXTERNAL Interrupt to occur. If the MCU has finished with the interrupt service routine, but the $\overline{\text{IRQ}}$ pin is still low, the EXTERNAL Interrupt will start again. In fact, the MCU will keep on servicing the EXTERNAL Interrupt as long as the $\overline{\text{IRQ}}$ pin is low. If the $\overline{\text{IRQ}}$ pin goes low for a while and resumes to high (a negative pulse) before the interrupt mask bit is cleared, the MCU will not recognize there was an interrupt request, and no interrupt will occur after the interrupt mask bit is cleared, i.e. there is no latch for the interrupt signal for the level sensitive triggering.

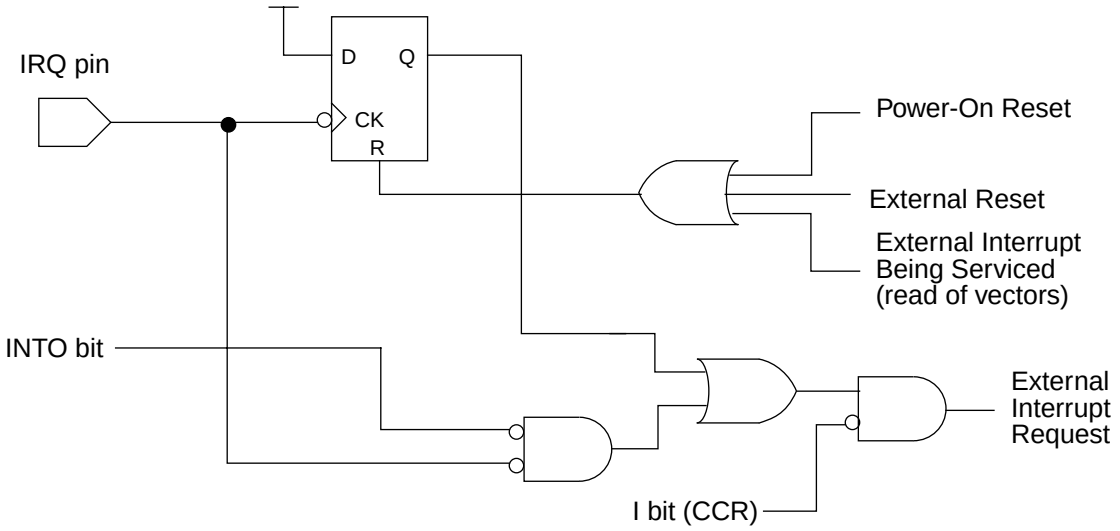
If the IRQ option is negative edge sensitive triggering (INTO=1), a negative edge occurs at the $\overline{\text{IRQ}}$ pin and the interrupt mask bit of the condition code register is cleared will cause an EXTERNAL Interrupt to occur. If the MCU has finished with the interrupt service routine, but the $\overline{\text{IRQ}}$ pin has not resumed back to high, no further interrupt will be generated. The interrupt logic recognizes negative edge transitions and pulses (special case of negative edges) only. If the negative edge occurs during the interrupt mask bit is set, the interrupt signal will be latched, an

interrupt will occur as soon as the interrupt mask bit is cleared. The latch will be cleared by RESET or cleared automatically during fetch of the EXTERNAL Interrupt vectors. Therefore, one (and only one) external interrupt edge could be latched during the interrupt mask bit is set.

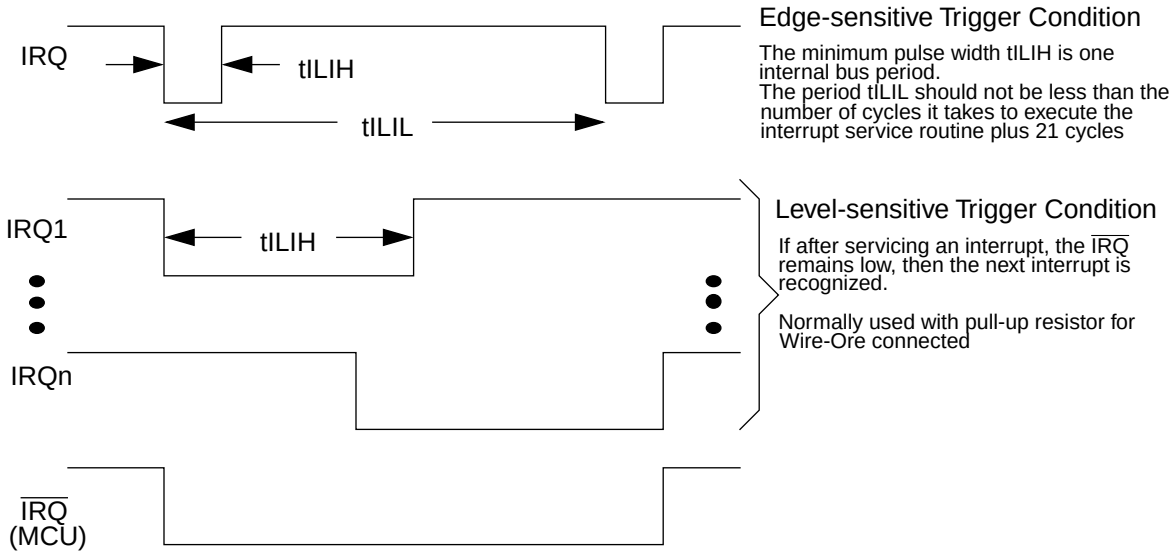
The service routine address is specified by the contents of $FFFA and $FFFB. **Figure 4-2** shows both a block diagram and the two methods for the interrupt line (IRQ) to the processor. The first method is single pulses on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the number of cycles required to execute the interrupt service routine plus 21 cycles. Once a pulse occurs, the next pulse should not occur until the MCU software has exited the routine (an RTI occurs). The second configuration shows several interrupt line “wire-ANDed” to perform the interrupts at the processor. Thus, if after servicing one interrupt and the interrupt line remains low, then the next interrupt is recognized.

NOTE

INTO is located at bit-1 of the Option Register at $001F, and is set by reset. The Option Register can only be written once after reset.



(a) Interrupt Functional Block Diagram



(b) Interrupt Mode Diagram

Figure 4-2. External Interrupts

4.3.2 External Interrupts ($\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, IRQ2, KBI[3:7])

4.3.2.1 $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, IRQ2

$\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, IRQ2 interrupt function is associated with PTA[0:2]. IRQ[0:2] interrupts behave similar to $\overline{\text{IRQ}}$ except these are edge-triggered only. $\overline{\text{IRQ0}}$ and $\overline{\text{IRQ1}}$ are both negative-edge triggered only and IRQ2 can be triggered on both rising and falling edges. To enable this function, IRQE0, IRQE1 and IRQE2 bit of the external interrupt control register (EXICR $1D) should be set first. When an appropriate edge occurring at the $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$ or IRQ2 pin and the interrupt mask bit of the condition code register is cleared will cause an external interrupt to occur. If the MCU has finished with the service routine, but the external interrupt pin has not resumed backed to high, no further interrupt will be generated. If the appropriate triggering edge occurs when the interrupt mask bit is set, the interrupt signal will be latched, and interrupt will occur as soon as the interrupt mask bit is cleared. The latch for $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$ and IRQ2 are cleared by reset or cleared by writing a ‘0’ to the IRQF0, IRQF1 and IRQF2 bit in the external interrupt status register ($1E EXISR) in the service routine.

The interrupt service routine is specified by the contents of the memory locations $FFF2 and $FFF3.

To prevent unwanted interrupts generated on **$\overline{\text{RQ0}}$, $\overline{\text{IRQ1}}$** pins. A logic ‘1’ should be written to these pins before setting their corresponding interrupt enable bits. Likewise, A logic ‘1’ should be written to **IRQ2** pin prior to setting the IRQ2 interrupt enable bit.

4.3.2.2 Keyboard Interrupts

Keyboard Interrupt function is associated with PTA[3:7]. The keyboard interrupt function is enabled by setting the individual interrupt enable bits KBE[3:7] (bits[3:7] of EXICR at $001D). When the KBE bit is set, the corresponding Port A pin will be configured as an input pin, regardless of the DDR setting, and a 100k ohm pull-up resistor is connected to the pin, as shown in **Figure 4-3**. When a high to low transition is sensed on the pin, a keyboard interrupt will be generated, provided the I-bit in the CCR is cleared.

The interrupt signal is latched, and it should be cleared by writing a ‘0’ to the corresponding KBIF bit in the external interrupt status register ($1E EXISR) in the interrupt service routine. This should be cleared after the key is debounced, or unwanted keyboard interrupt signal will be generated.

The Keyboard Interrupt is negative-edge sensitive only, and the interrupt service routine is specified by the contents of the memory locations $FFF2 and $FFF3.

To prevent unwanted interrupts generated on **KBI[3:7]** pins. A logic ‘1’ should be written to these pins before setting their corresponding interrupt enable bits.

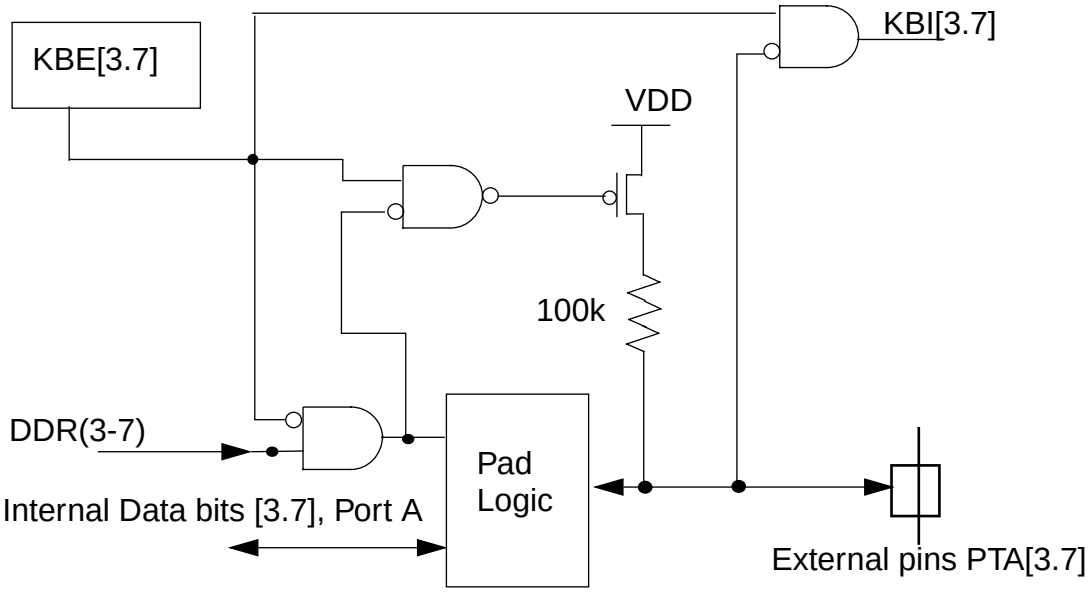


Figure 4-3. Keyboard Interrupt Circuitry

4.3.2.3 External Interrupt Control Register (EXICR $001D)

		7	6	5	4	3	2	1	0
EXICR $001D	READ / WRITE	KBE7	KBE6	KBE5	KBE4	KBE3	IRQ2E	IRQ1E	IRQ0E
	RESET	0	0	0	0	0	0	0	0

KBE[7:3]
- 1 = Enable Keyboard interrupt on PTA[7:3] respectively.
- 0 = Disable Keyboard interrupt on PTA[7:3] respectively.

IRQE[2:0]
- 1 = Enable IRQ interrupt on PTA[2:0] respectively.
- 0 = Disable IRQ interrupt PTA[2:0] respectively.

4.3.2.4 External Interrupt Status Register (EXISR)

		7	6	5	4	3	2	1	0
EXISR $001E	READ	KBIF7	KBIF6	KBIF5	KBIF4	KBIF3	IRQF2	IRQF1	IRQF0
	WRITE								
	RESET	0	0	0	0	0	0	0	0

KBIF[7:3]

The keyboard interrupt flag bit is set to ‘1’ when a negative edge has been detected at the pin and the corresponding enable bit has been set. Writing a ‘0’ to this bit will clear the corresponding interrupt flag. This bit should be cleared by software in the keyboard interrupt service routine, or the CPU will keep on serving this interrupt.

IRQF[2:0]

The external IRQ flag bit is set to ‘1’ when an appropriate edge has been detected at the pin and the corresponding enable bit has been set. Writing a “0” to this bit will clear the IRQ interrupt flag. This bit should be cleared by software in the keyboard interrupt service routine, or the CPU will keep on serving this interrupt.

4.3.3 Caller ID Interrupt (RDI/CDI,CDRI)

This interrupt is caused by the Caller ID module when a valid RT_L signal, a Ring is detected, a Carrier is detected or when the 8-bit Called ID data is ready to be read. The enable and flag bits for the Ring Detect and the Carrier detect are located in CLCSR1 register and the enable and flag bits for the Called ID data ready are located in CLCSR3 register. The RT_L is always active once it is enabled by a metal mask, see **Section 11**. All four interrupts will vector to the same interrupt service routine located at the addresses specified by the contents of memory locations $FFF4-$FFF5. The RT_L interrupt will wake up the MCU from the STOP and WAIT mode whereas the Ring Detect, Carrier Detect and Data Ready interrupts will wake up the CPU from Wait mode. See **Section 6**.

4.3.4 TIMER Interrupt

The TIMER interrupt is generated by the multi-function Timer when a timer overflow or an input capture or a output compare has occurred as described in **Section 8**. The interrupt enable bit and flag for the Timer interrupt are located in the Timer Control and Status Registers TCR and TSR located at $0011 and $0012 respectively. The I-bit in the CCR must be clear in order for the Timer interrupt to

be enabled. This interrupt will vector to the interrupt service routine located at the address specified by the contents of memory locations $FFF8 and $FFF9.

4.3.5 SPI Interrupt

The SPI interrupt is generated by serial peripheral interface as described in **Section 9**. The interrupt enable bits for the SPI interrupt is in the SPI Control Register (SPCR) at location $0020. The I-bit in the CCR must be clear in order for the SPI interrupt to be enabled. These interrupts will vector to the same interrupt service routine located at the address specified by the contents of memory locations $FFF6 and $FFF7.

4.3.6 CTIMER, WTimer Interrupt (CORE TIMER, Watch Timer)

4.3.6.1 Core Timer Interrupt

The CTIMER interrupt is generated by the Core Timer when a core timer overflow, or real time interrupt has occurred as described in **Section 8**. The interrupt enable bits and flags for the Core Timer interrupts are located in the Core Timer Control and Status Register (CTCSR) located at $0008. The I-bit in the CCR must be clear in order for the CTIMER interrupt to be enabled.

4.3.6.2 Watch Timer Interrupt

The WTimer interrupt is generated by the Watch Timer when a 1 second timer overflow interrupt has occurred as described in **Section 8**. The interrupt enable bits and flags for the Watch Timer interrupt are located in the Watch Timer Control and Status Register (WTCSR) located at $0023. The I-bit in the CCR must be clear in order for the WTIMER interrupt to be enabled.

4.3.6.3 Interrupt Vector

The above two interrupts will vector to the same interrupt service routine located at the address specified by the contents of memory locations $FFF0 and $FFF1.

SECTION 5
RESETS

The MCU can be reset in three ways:

- by the initial power-on reset function, (POR)
- by an active low input to the $\overline{\text{RESET}}$ pin, ($\overline{\text{RESET}}$)
- by a COP watchdog timer reset, (COPR)

5.1 EXTERNAL RESET ($\overline{\text{RESET}}$)

The $\overline{\text{RESET}}$ pin is the only external source of a reset. This pin is connected to a Schmitt trigger input gate to provide an upper and lower threshold voltage separated by a minimum amount of hysteresis. This external reset occurs whenever the $\overline{\text{RESET}}$ pin is pulled below the lower threshold and remains in reset until the $\overline{\text{RESET}}$ pin rises above the upper threshold. This active low input will generate the RST signal and reset the CPU and peripherals. Termination of the external RESET input can alter the operating mode of the MCU.

The $\overline{\text{RESET}}$ pin can also be pulled to a low state by an internal pulldown that is activated by the internal COP Watchdog or Power-on resets. This $\overline{\text{RESET}}$ pin pulldown device will only be activated for one cycle of the internal clock, PH2, when a COP Watchdog reset occurs; or will remain activated as long as counting the power-on reset cycles or the low voltage is detected.

5.2 POWER-ON RESET (POR)

The internal POR is generated on power-up to allow the clock oscillator to stabilize. The POR is strictly for power turn-on conditions and is not able to detect a drop in the power supply voltage (brown-out). There is an oscillator stabilization delay of 488 internal processor bus clock cycles (PH2) after the oscillator becomes active. The $\overline{\text{RESET}}$ pin will be pulled down internally during these cycles.

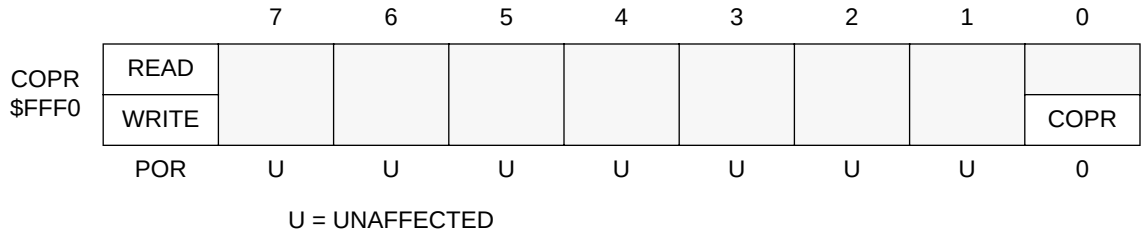
The POR will generate the RST signal which will reset the CPU. If any other reset function is active at the end of this 488 cycle delay, the RST signal will remain in the reset condition until the other reset condition(s) end.

5.3 COMPUTER OPERATING PROPERLY RESET (COPR)

The internal COPR reset is generated automatically (if enabled) by a time-out of the COP Watchdog Timer. This time-out occurs if the counter in the COP Watchdog Timer is not reset (cleared) within a specific time by a program reset sequence. The COP Watchdog Timer can be enabled by setting bit 6 of the Option Register at $001F. Refer to **Section 8** for more information on this time-out feature.

The COP Watchdog reset will activate the internal pulldown device connected to the $\overline{\text{RESET}}$ pin for one cycle of the internal clock, PH2.

The COP register shares the same address ($FFF0) with the MS byte of the Core Timer Interrupt Vector as shown below. Reading this location will return the MS byte of the Core Timer Interrupt Vector. Writing $00 to this location clears the COP watchdog timer.



SECTION 6
LOW POWER MODES

6.1 LOW-POWER MODES

The MC68HC05CL48 has two low-power operational modes. The WAIT and STOP instructions provide two modes that reduce the power required for the MCU by stopping various internal clocks and/or the on-chip oscillator. The STOP and WAIT instructions are not normally used if the COP Watchdog Timer is enabled. The flow of the STOP and WAIT modes is shown in **Figure 6-1**.

6.1.1 STOP Instruction

Execution of the STOP instruction, places the MCU in its lowest power consumption mode. In the STOP Mode the internal oscillator is turned off, halting *all* internal processing, including the COP Watchdog Timer.

When the CPU enters STOP Mode, the I-bit in the Condition Code Register will be cleared automatically. This enable the external hardware interrupt to wake up the MCU. All other registers and memory remain unaltered. All input/output lines remain unchanged.

The MCU can be brought out of the STOP Mode only by a Caller ID (RT_L) interrupt, WTIMER interrupt, a hardware interrupt “IRQ, IRQ(n), KBI(n)” or an externally generated RESET. When exiting the STOP Mode the internal oscillator will resume after a 488 internal processor clock cycle oscillator stabilization delay.

6.1.2 WAIT Instruction

The WAIT instruction places the MCU in a low-power mode, which consumes more power than the STOP Mode. In the WAIT Mode the internal processor clock is halted, suspending all processor and internal bus activity. Other Internal clocks remain active, permitting interrupts to be generated from the functional blocks, or a reset to be generated from the COP Watchdog Timer. The Timer may be used to generate a periodic exit from the WAIT Mode. Execution of the WAIT instruction automatically clears the I-bit in the Condition Code Register, so that any hardware interrupt can wake up the MCU. All other registers, memory, and input/output lines remain in their previous states.

6.2 DATA-RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltage as low as 2.0 VDC. This is called the data-retention mode where the data is held, but the device is not guaranteed to operate. The $\overline{\text{RESET}}$ pin must be held low during data-retention mode.

To put CPU into data retention mode with the lowest power:

- Clear OSC_ON bit in OPTION register.
- Run system with Oscillator clock.
- Execute STOP instruction to put CPU in STOP mode.
- Drive $\overline{\text{RESET}}$ pin to logical zero.
- Lower the VDD voltage. The $\overline{\text{RESET}}$ must remain low continuously during data retention mode.

To take the CPU out of data retention mode:

- Return VDD to normal operating level.
- Return the $\overline{\text{RESET}}$ pin to logical one.

6.3 COP WATCHDOG TIMER CONSIDERATIONS

If the COP Watchdog Timer is selected by setting the enable bit, any execution of the STOP instruction (either intentional or inadvertent due to the CPU being disturbed) will be executed as a WAIT instruction. It is because, if a STOP instruction could be executed, while the COP was enabled. The STOP instruction will cause the oscillator to halt and prevent the COP Watchdog Timer from timing out. Therefore, the STOP instruction will put the MCU into WAIT mode, instead of STOP mode, if COP is enabled.

If the COP Watchdog Timer is selected, the COP will reset the MCU when it times out. Therefore, it is recommended that the COP Watchdog should be **disabled** for a system that must have intentional uses of the WAIT Mode for periods longer than the COP time-out period.

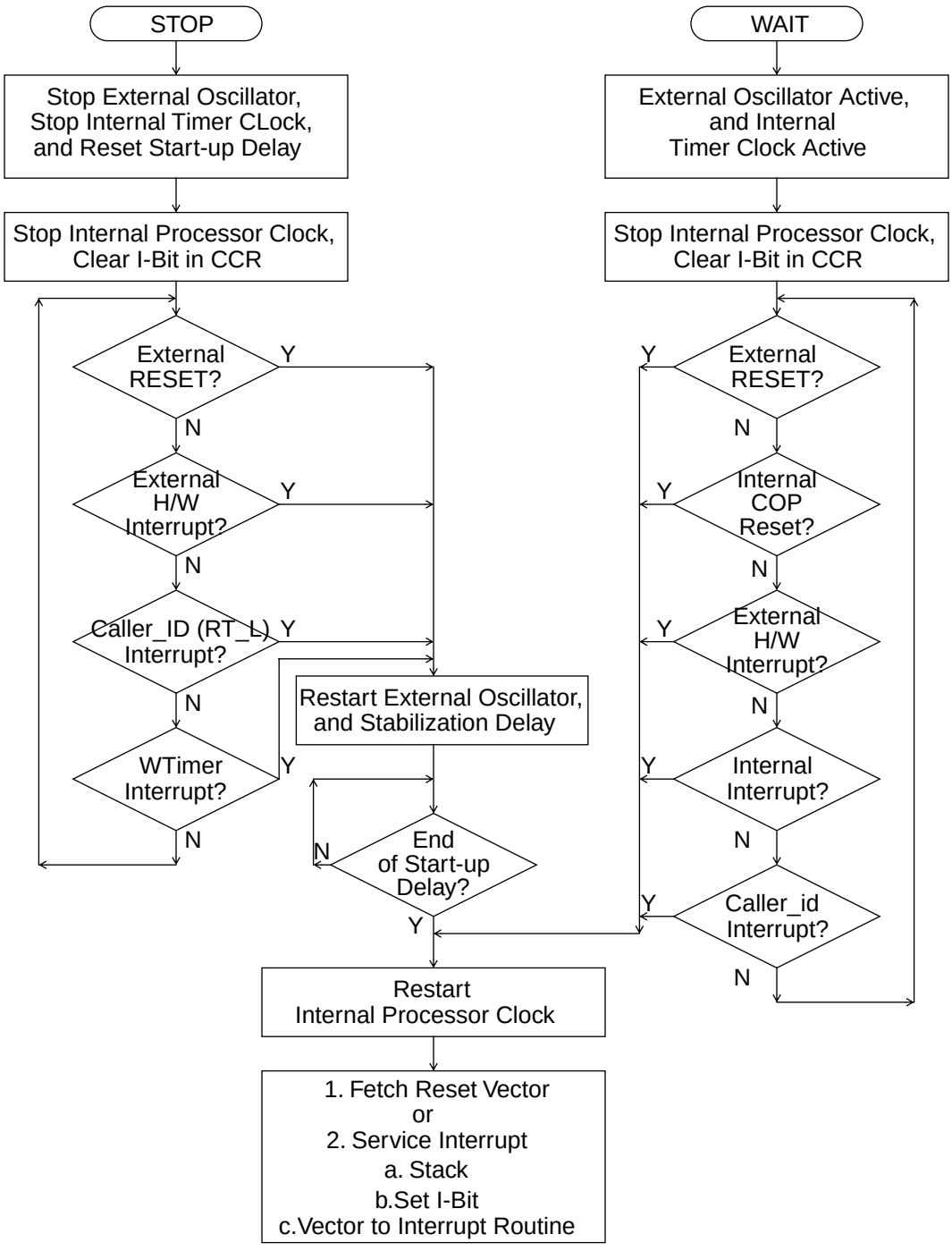


Figure 6-1. STOP/WAIT Flowchart

SECTION 7
INPUT/OUTPUT PORTS

In single-chip mode there are 24 pins arranged as 3 8-bit I/O port. The I/O ports are programmable as either inputs or outputs under software control of the data direction registers.

To avoid a glitch on the output pins, write data to the I/O Port Data Register before writing a one to the corresponding Data Direction Register.

7.1 PARALLEL PORTS

Port A and B and C are 8-bit bidirectional port. Each Port pin is controlled by the corresponding bits in a data direction register and a data register as shown in **Figure 7-1**. The functions of the I/O pins are summarized in **Table 7-1**.

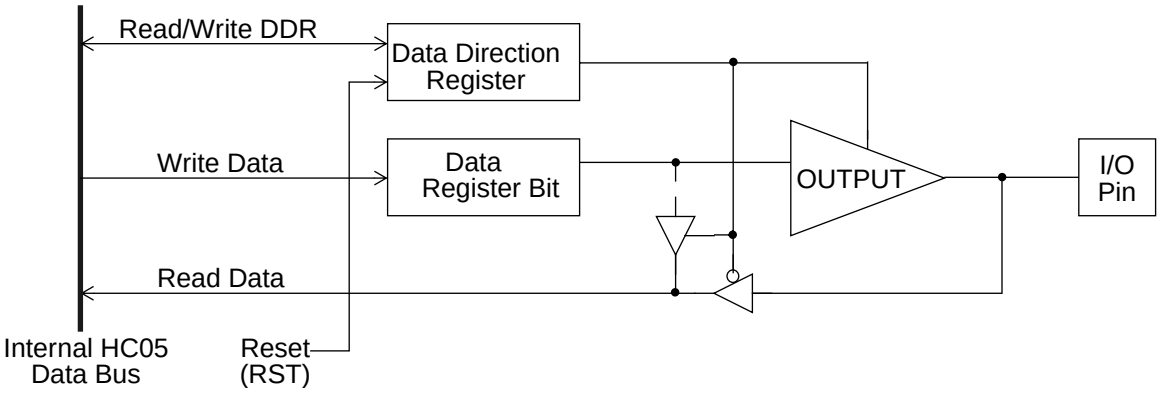


Figure 7-1. Port I/O Circuitry

Table 7-1. Port I/O Pin Functions

R/$\overline{W}$	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

7.2 PORT A

Port A is an 8-bit bidirectional port. The port A data register is at $0000 and the data direction register (DDRA) is at $0003. Reset does not affect the data register, but clears the data direction register, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode. In addition to normal I/O port function, PTA0-PTA2 are also used as additional external interrupt inputs and PTA3-PTA7 are also associated with the KEYBOARD interrupt function. See **Section 4.3.2** for detail description on External and Keyboard interrupts.

7.3 PORT B

Port B is an 8-bit bidirectional port. The port B data register is at $0001 and the data direction register (DDRB) is at $0004. Reset does not affect the data register, but clears the data direction register, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode.

7.4 PORT C

Port C is an 8-bit bidirectional port which shares its pins with subsystems A-2-D, SPI and Timer under the control of the A-2-D status and control register ADCSR, the SPI control register SPCR and the Timer pin configuration register TIMCONF. The port C data register is at $0002, the data direction register (DDRC) is at $0005. Reset does not affect the data register, but clears the data direction, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode. Writing a ‘1’ to the SPE-bit of the SPI control register SPCR configures PTC[0:3] as dedicated SPI pins. Setting CONF6-bit and CONF7-bit of the Timer pin configuration register TIMCONF to ‘1’ configures PTC6 and PTC7 as Timer input capture pins respectively.

PTC[4:5] will always be available as IO port pins even after the ADON-bit is set. When ADON-bit is set to ‘1’ and the CH[2:0] bits of the A-2-D status and control register is set to select channel-2, PTC[4] becomes an A-2-D input pin. When ADON-bit is set to ‘1’ and the CH[2:0] bits of the A-2-D status and control register is set to select channel-3, PTC[5] becomes an A-2-D input pin.

Table 7-1. summarizes the PORT C function when used as sub-system pins.

Table 7-2. PORT C I/O Configuration

PORT C	I/O Mode	Sub-System Function
PTC0	Input Output	SCK: SPI slave clock-in (MSTR = ‘0’, SPE=1). SCK: SPI master clock-out (MSTR = ‘1’, SPE=1).

Table 7-2. PORT C I/O Configuration

PORT C	I/O Mode	Sub-System Function
PTC1	Input Output	MISO: SPI slave data out (MSTR=0, SPE=1) MISO: SPI master data in (MSTR=1, SPE=1)
PTC2	Input Output	MOSI: SPI slave data in (MSTR=0, SPE=1) MOSI: SPI master data out (MSTR=1, SPE=1)
PTC3	Input	$\overline{SS}$: SPI slave select signal (SPE=1) Note: This pin is internally connected to V_{DD} when (MSTR=1), this means that PTC3 can be used as a normal port pin.
PTC4	Input	AD2: A-to-D input (ADON=1 and Channel-2 selected)
PTC5	Input	AD3: A-to-D input (ADON=1 and Channel-3 selected)
PTC6	Input	TCAP1: Timer input capture1 (CONF6=1).
PTC7	Input	TCAP2: Timer input capture2 (CONG7=1).

SECTION 8
TIMER

The MC68HC05CL48 has three timers:

- 16-bit free-running timer
- 15-stage multi-functional (core) timer
- One second timer

8.1 16-BIT FREE-RUNNING TIMER

The 16-bit free-running counter is driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to **Figure 8-1** for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE

The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

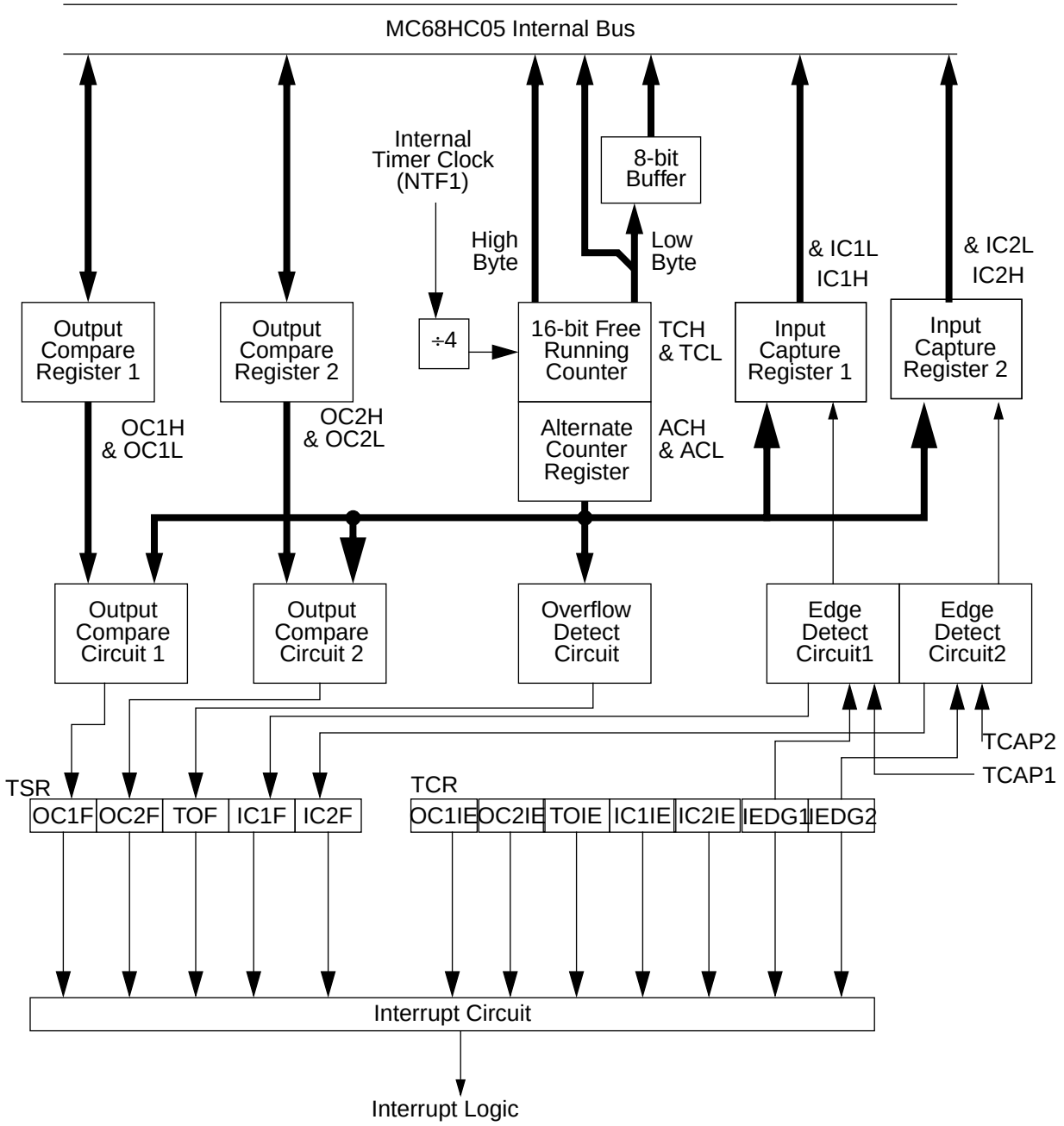


Figure 8-1. 16-Bit Free-running Timer Block Diagram

8.1.1 Counter

		7	6	5	4	3	2	1	0
TCH $0019	READ	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
	WRITE								
	RESET	1	1	1	1	1	1	1	1

		7	6	5	4	3	2	1	0
TCL $001A	READ	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
	WRITE	Counter Reset							
	RESET	1	1	1	1	1	1	0	0

		7	6	5	4	3	2	1	0
ACH $001B	READ	AC15	AC14	AC13	AC12	AT11	AC10	AC9	AC8
	WRITE								
	RESET	1	1	1	1	1	1	1	1

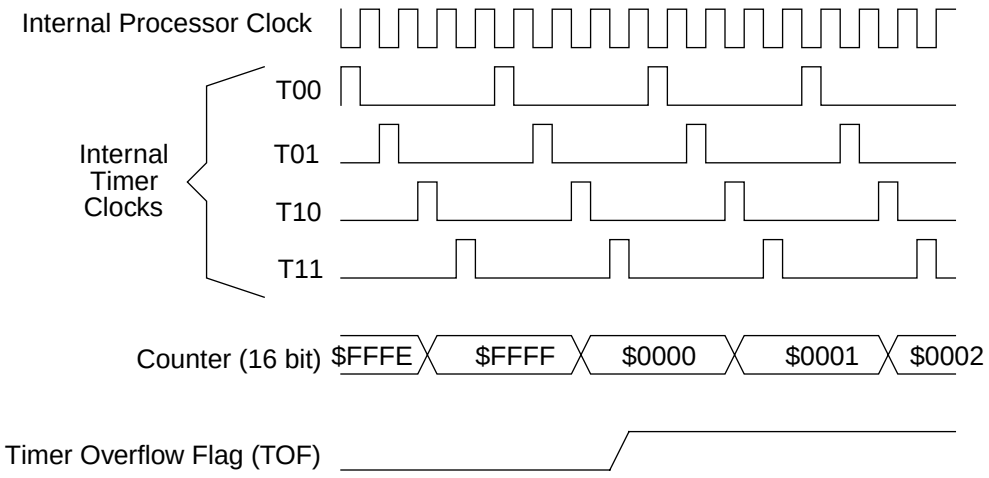
		7	6	5	4	3	2	1	0
ACL $001C	READ	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
	WRITE	Counter Reset							
	RESET	1	1	1	1	1	1	0	0

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.2 microseconds if the internal bus clock is 1.8MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, $19-$1A (counter register) or $1B-$1C (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter ($1A, $1C) receives the count value at the time of the read. If a read of the free-running

counter or counter alternate register first addresses the most significant byte (MSB) ($19, $1B), the LSB ($1A, $1C) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($1A or $1C) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register MSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.



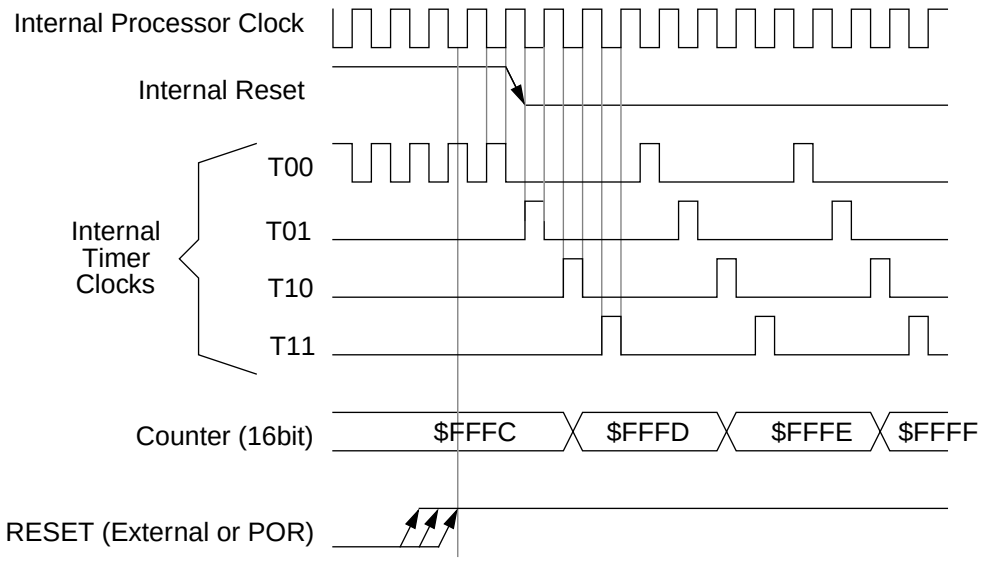
NOTE: The TOF bit is set at timer state T11 (transition of counter from $FFFF to $0000). It is cleared by read of the Timer Status Register during the internal processor clock time followed by a read of the counter low register.

Figure 8-2. Timer State Diagram For Timer Overflow

The free-running counter is configured to $FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to $FFFC and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt can also be enabled when counter roll over occurs by setting its interrupt enable bit (TOIE).

In some particular timing control applications it may be desirable to reset the 16-bit free running counter under software control. When the low byte of the counter ($1A or $1C) is written to, the counter is configured to its reset value ($FFFC).

The divide-by-4 prescaler is also reset and the counter resumes normal counting operation. All of the flags and enable bits remain unaltered by this operation. If access has previously been made to the high byte of the free running counter ($19 or $1B), then the reset counter operation terminates the access sequence.



NOTE: The Counter Register and Timer Control Register are the only ones affected by RESET.

Figure 8-3. Timer State Timing Diagram For Reset

8.1.2 Output Compare Registers

There are two output compare registers: output compare register 1 and output compare register 2. Output compare registers can be used for several purposes such as controlling an output waveform or indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

8.1.2.1 Output Compare Register 1

		7	6	5	4	3	2	1	0
OC1H $0015	READ	OC1:15	OC1:14	OC1:13	OC1:12	OC1:11	OC1:10	OC1:9	OC1:8
	WRITE								
	RESET	U	U	U	U	U	U	U	U

		7	6	5	4	3	2	1	0
OC1L $0016	READ	OC1:7	OC1:6	OC1:5	OC1:4	OC1:3	OC1:2	OC1:1	OC1:0
	WRITE								
	RESET	U	U	U	U	U	U	U	U

The 16-bit output compare register 1 is made up of two 8-bit registers at locations $15 (MSB) and $16 (LSB). The output compare register contents are compared with the contents of the free-running counter once every four internal processor clock cycles. If a match is found, the corresponding output compare flag OC1F (bit 5 of timer status register $12) is set.

The output compare register values should be changed after each successful comparison to establish a new elapsed time-out. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit OC1IE (bit 5 of timer control register $11) is set.

After a processor write cycle to the output compare register containing the MSB ($15), the output compare function is inhibited until the LSB ($16) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($16) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte.

Because the output compare flag OC1F and the output compare register 1 are undetermined at power on, and are not affected by external reset, care must be exercised when initializing the output compare function. The following procedure is recommended:

1. Write the high byte to the compare register 1 to inhibit further compares until the low byte is written.

2. Read the status register to arm the OC1F if it is already set.

3. Write the output compare register 1 low byte to enable the output compare 1 function with flag clear.

The purpose of this procedure is to prevent the OC1F bit from being set between the time it is read and the write to the corresponding output compare register.

8.1.2.2 Output Compare Register 2

		7	6	5	4	3	2	1	0
OC2H $0017	READ	OC2:15	OC2:14	OC2:13	OC2:12	OC2:11	OC2:10	OC2:9	OC2:8
	WRITE								
	RESET	U	U	U	U	U	U	U	U

		7	6	5	4	3	2	1	0
OC2L $0018	READ	OC2:7	OC2:6	OC2:5	OC2:4	OC2:3	OC2:2	OC2:1	OC2:0
	WRITE								
	RESET	U	U	U	U	U	U	U	U

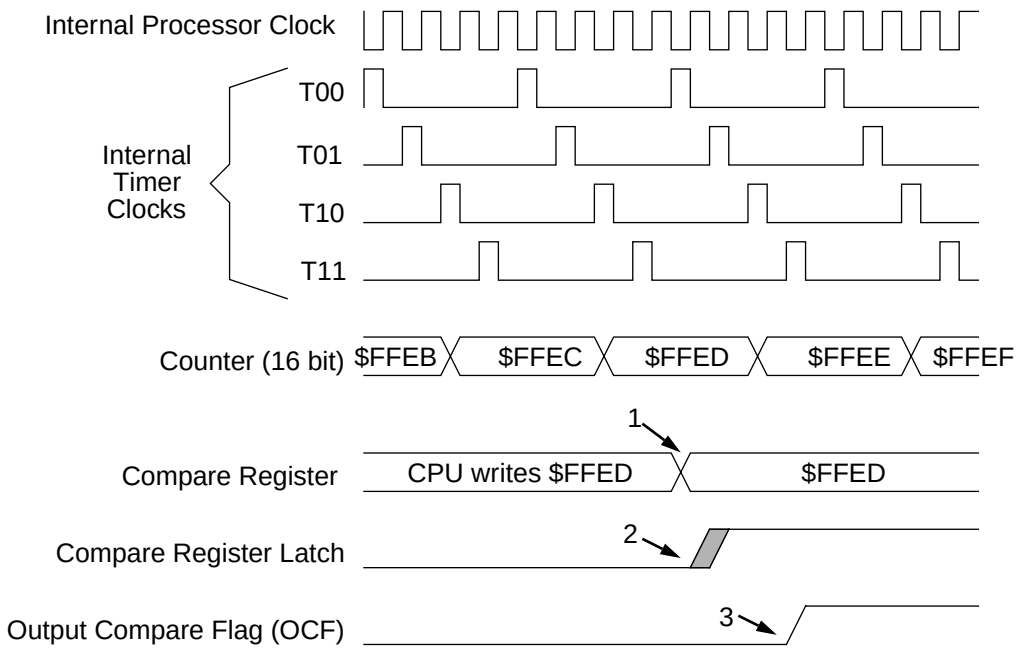
The 16-bit output compare register 2 is made up of two 8-bit registers at locations $17 (MSB) and $18 (LSB). The output compare register contents are compared with the contents of the free-running counter once every four internal processor clock cycles. If a match is found, the corresponding output compare flag OC2F (bit 4 of timer status register $12) is set.

The output compare register values should be changed after each successful comparison to establish a new elapsed time-out. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit OC2IE (bit 4 of timer control register $11) is set.

After a processor write cycle to the output compare register containing the MSB ($17), the output compare function is inhibited until the LSB ($18) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($18) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte.

Because the output compare flag OC2F and the output compare register 2 are undetermined at power on, and are not affected by external reset, care must be exercised when initializing the output compare function. A procedure as recommended for Compare Register 1 should be followed.



1. The CPU writes to the Compare Register may take place at any time, but a compare only occurs at timer state T01. Thus, a 4-cycle different may exist between the write to the Compare Register and the actual compare.
2. Internal compare takes place during timer state T01.
3. OCF is set at the timer state T11 which follows the comparison match ($FFED in this example).

Figure 8-4. Timer State Timing Diagram For Output Compare

8.1.3 Input Capture Registers

8.1.3.1 Input Capture Register 1

		7	6	5	4	3	2	1	0
IC1H $0013	READ	IC1:15	IC1:14	IC1:13	IC1:12	IC1:11	IC1:10	IC1:9	IC1:8
	WRITE								
	RESET	U	U	U	U	U	U	U	U

		7	6	5	4	3	2	1	0
IC1L $0014	READ	IC1:7	IC1:6	IC1:5	IC1:4	IC1:3	IC1:2	IC1:1	IC1:0
	WRITE								
	RESET	U	U	U	U	U	U	U	U

Two 8-bit registers, which make up the 16-bit input capture register, these are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG1). Reset does not affect the contents of the input capture register.

- IEDG1 = 0: Capture on negative edge
- IEDG1 = 1: Capture on positive edge

An interrupt can also accompany a capture provided the corresponding interrupt enable bit, ICI1E (bit 7 of the timer control register $11) is set.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (IC1F) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register ($13) MSB, the counter transfer is inhibited until the LSB ($14) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($14) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

8.1.3.2 Input Capture Register 2

		7	6	5	4	3	2	1	0
IC2H $000F	READ	IC2:15	IC2:14	IC2:13	IC2:12	IC2:11	IC2:10	IC2:9	IC2:8
	WRITE								
	RESET	U	U	U	U	U	U	U	U

		7	6	5	4	3	2	1	0
IC2L $0010	READ	IC2:7	IC2:6	IC2:5	IC2:4	IC2:3	IC2:2	IC2:1	IC2:0
	WRITE								
	RESET	U	U	U	U	U	U	U	U

Two 8-bit registers, which make up the 16-bit input capture register, these are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG2). Reset does not affect the contents of the input capture register.

- IEDG2 = 0: Capture on negative edge
- IEDG2 = 1: Capture on positive edge

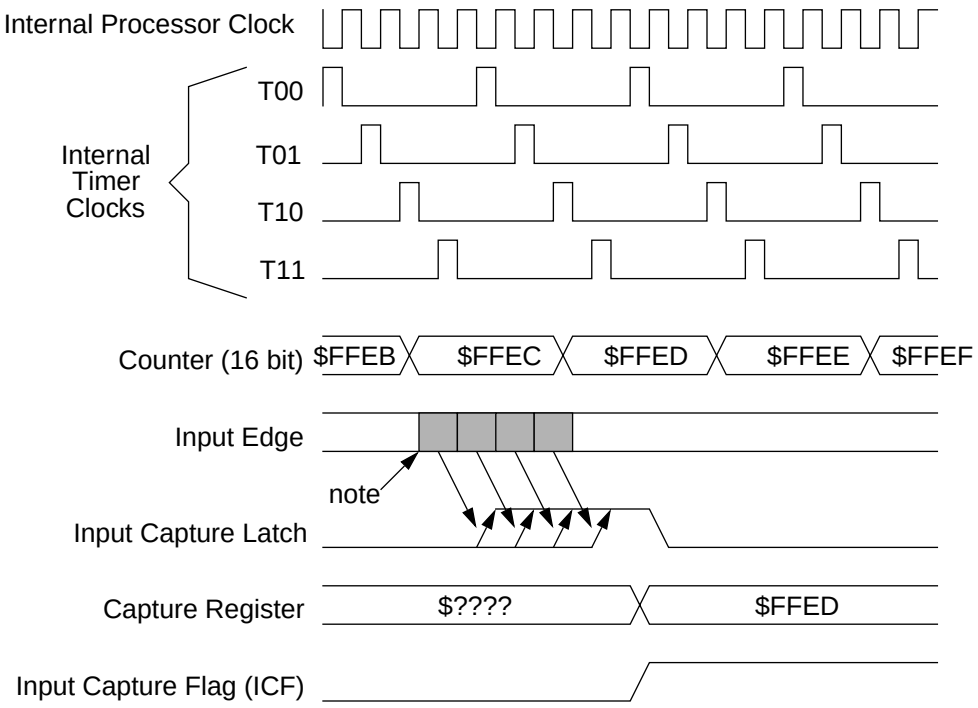
An interrupt can also accompany a capture provided the corresponding interrupt enable bit, ICI2E (bit 6 of the timer control register $11) is set.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (IC2F) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register ($0F) MSB, the counter transfer is inhibited until the LSB ($10) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($10) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.



NOTE: If the input edge occurs in the shaded area from one timer state T10 to the other timer state T10 the Input Capture Flag is set during the next state T11.

Figure 8-5. Timer State Timing Diagram For Input Capture

8.1.4 Timer Control Register (TCR)

The timer control registers TCR is a read/write register. Five bits control interrupts associated with the timer status register flags IC1F, IC2F, OC1F, OC2F and TOF. The other two bits control which edge is significant to the input capture edge detector (i.e., negative or positive). The timer control register and the free running counter are the only sections of the timer affected by reset. The timer control registers are illustrated below by a definition of each bit.

		7	6	5	4	3	2	1	0
TCR $0011	READ	IC1IE	IC2IE	OC1IE	OC2IE	TOIE		IEDG2	IEDG1
	WRITE								
	RESET	0	0	0	0	0	X	U	U

U = UNAFFECTED

IC1IE - Input Capture 1 Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

IC2IE - Input Capture 2 Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

OC1IE - Output Compare 1 Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

OC2IE - Output Compare 2 Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

TOIE - Timer Overflow Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

IEDG2 - Input Edge 2

Value of the input edge determines which level transition on TCAP2 pin will trigger a free running counter transfer to the input capture register. Reset does not affect the IEDG2 bit.
- 1 = positive edge
- 0 = negative edge

IEDG1 - Input Edge

Value of the input edge determines which level transition on TCAP1 pin will trigger a free running counter transfer to the input capture register. Reset does not affect the IEDG1 bit.
- 1 = positive edge
- 0 = negative edge

8.1.5 Timer Status Register (TSR)

The timer status register is a read-only register and is illustrated below followed by a definition of each bit. Refer to timing diagrams shown in **Figure 8-3**, **Figure 8-2** and **Figure 8-4** for timing relationship to the timer status register bits.

	7	6	5	4	3	2	1	0
TSR $0012 READ	IC1F	IC2F	OC1F	OC2F	TOF			
WRITE								
RESET	U	U	U	U	U	U	U	0

U = UNAFFECTED

IC1F - Input Capture Flag

1 = Flag set when a selected polarity edge has been sensed by the input capture edge detector.

0 = Flag cleared by reading the timer status register (with IC1F set) followed by accessing the low byte ($14) of the input capture register.

IC2F - Input Capture Flag

1 = Flag set when a selected polarity edge has been sensed by the input capture edge detector.

0 = Flag cleared by reading the timer status register (with IC2F set) followed by accessing the low byte ($10) of the input capture register.

OC1F - Output Compare 1 Flag

1 = Flag set when the output compare register 1 contents matches the contents of the free running counter.

0 = Flag cleared by reading the timer status register (with OC1F set) and then accessing the low byte ($16) of the output compare 1 register.

OC2F - Output Compare 2 Flag

1 = Flag set when the output compare register 2 contents matches the contents of the free running counter.

0 = Flag cleared by reading the timer status register (with OC2F set) and then accessing the low byte ($18) of the output compare 1 register.

TOF - Timer OVerflow Flag

1 = Flag set by transition of the free running counter from $FFFF to $0000.

0 = Flag cleared by reading the timer status register (with TOF set) followed by an access of the free running counter least significant byte ($1A).

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1. The timer status register is read or written when TOF is set, and

2. The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1B and $1C contains the same value as the free-running counter (at address $19 and $1A); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

8.1.6 Timer Pin Configuration Register (TIMCONF)

The timer pin configuration register is a read/write register, this is used to control PTC[6:7] pin function.

		7	6	5	4	3	2	1	0
TIMCONF $0006	READ	CONF7	CONF6						
	WRITE								
	RESET	0	0	U	U	U	U	U	0

U = UNAFFECTED

CONF7- Pin configuration 7

1 = Configures PTC7 as timer input capture pin.
0 = PTC7 used as normal I/O port pin.

CONF6- Pin configuration 6

1 = Configures PTC6 as timer input capture pin.
0 = PTC6 used as normal I/O port pin.

8.1.7 Operation During Low Power Mode

During the wait and stop modes, the timer stops and holds at its current state, retaining all data, and resumes operation from this point when external interrupt (IRQ), or internal interrupt is received.

8.2 CORE TIMER

The MC68HC05CL48 Core Timer (or Ctimer) for is a 15-stage multi-functional ripple counter. The features include Timer Over Flow, Power-On Reset (POR), Real Time Interrupt, and COP Watchdog Timer.

As seen in **Figure 8-6**, the Timer is driven by the internal bus clock divided by four with a fixed prescaler. This signal drives an 8-bit ripple counter. The value of this 8-bit ripple counter can be read by the CPU at any time by accessing the Ctimer Counter Register (CTCR) at address $09. A timer overflow function is implemented on the last stage of this counter, giving a possible interrupt at the rate of E/1024. One additional stages produce the POR function at E/2064. The Timer Counter Bypass circuitry (available only in Test Mode) is at this point in the

timer chain. This circuit is followed by two more stages, with the resulting clock (E/16384) driving the Real Time Interrupt circuit. The RTI circuit consists of three divider stages with a 1 of 4 selector. The output of the RTI circuit is further divided by eight to drive the optional COP Watchdog Timer circuit. The RTI rate selector bits, and the RTI and CTOF enable bits and flags are located in the Ctimer Control and Status Register(CTCSR) at location $08.

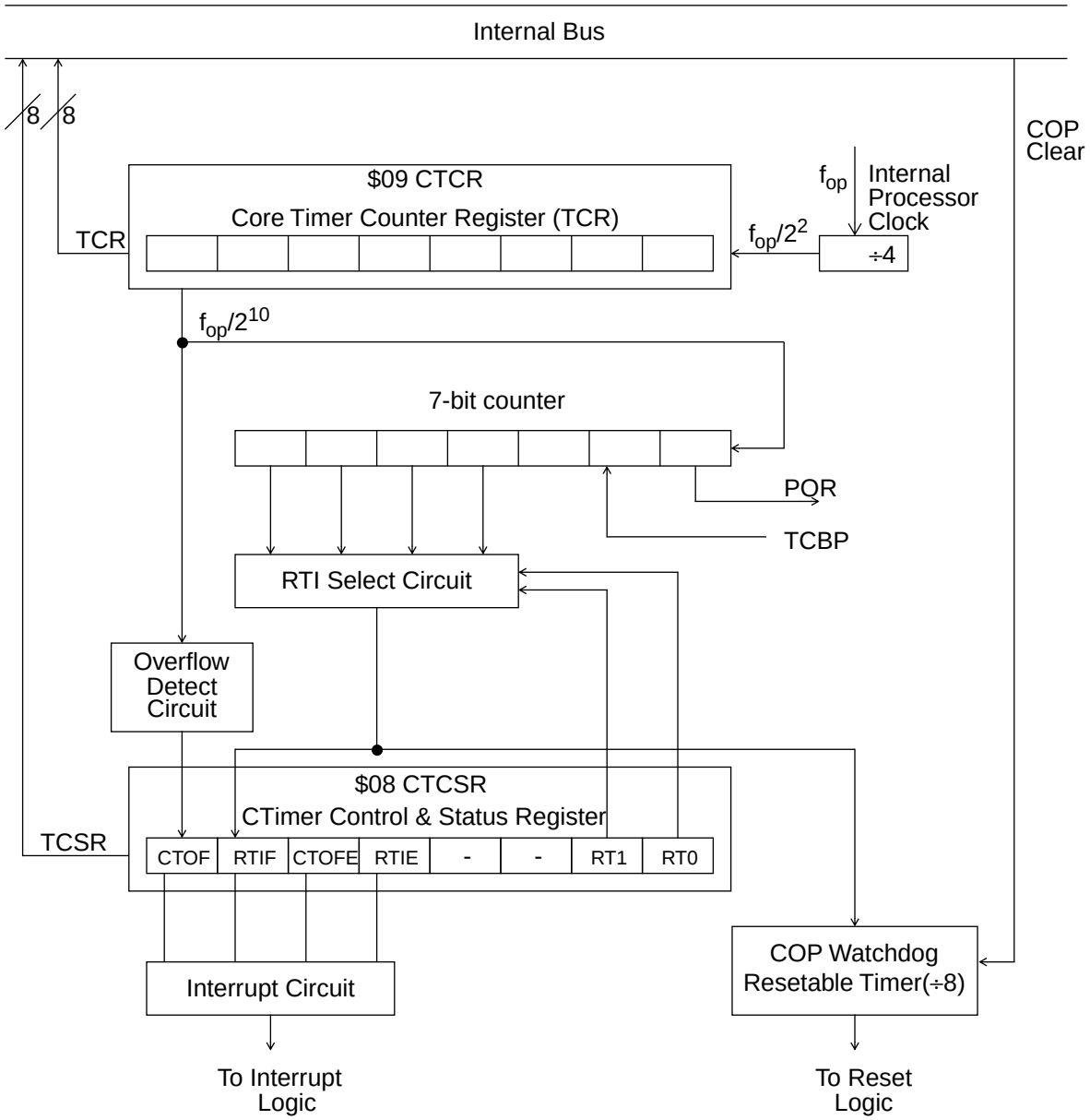


Figure 8-6. Core Timer Block Diagram

8.2.1 Computer Operating Properly (COP) Watchdog reset

The COP watchdog timer function is implemented on this device by using the output of the RTI circuit and further dividing it by eight. The minimum COP reset rates are listed in **Table 8-1**. If the COP circuit times out, an internal reset is generated and the normal reset vector is fetched.

Preventing a COP time-out is done by writing a “0” to bit 0 of address $FFF0. This location is shared with User ROM byte. And reading this location will return the User ROM data. When the COP is cleared, only the final divide by eight stage (output of the RTI) is cleared. If the COP (Computer Operating Properly) Watchdog Timer circuit times out, an internal reset is generated and the reset vector is fetched.

This function is software selectable by setting the COP bit in the Option Register. The COP bit is set after reset, i.e. the COP function is defaulted. The COP function can be disable by writing a ‘0’ to this bit.

NOTE

COP is located at bit 6 of the Option Register at $001F, which can only be written once after reset.

8.2.2 Ctimer Control and Status Register (CTCSR)

The CTCSR contains the timer interrupt flag, the timer interrupt enable bits, and the real time interrupt rate select bits.

		7	6	5	4	3	2	1	0
CTCSR $0008	READ	**CTOF**	**RTIF**	**CTOFE**	**RTIE**	**0**	**0**	**RT1**	**RT0**
	WRITE								
	RESET	0	0	0	0	0	0	1	1

CTOF - Core Timer Over Flow

This is a clearable, read-only status bit and is set when the 8-bit ripple counter rolls over from $FF to $00.

1 = No effect

0 = Clearing the TOF

RTIF - Real Time Interrupt Flag

The Real Time Interrupt circuit consists of a three stage divider and a 1 of 4 selector. The clock frequency that drives the RTI circuit is $E/2^{13}$ with three additional divider stages. This flag is a clearable, read-only status bit and is set

when the output of the chosen (1 of 4 selection) stage goes active.

1 = No effect
0 = Clearing the RTIF

CTOFE - Core Timer Overflow Enable

When this bit is set, a CPU interrupt request is generated when the TOF bit is set, provided the I bit in CCR is cleared.

1 = Interrupt enable
0 = Interrupt disable

RTIE - Real Time Interrupt Enable

When this bit is set, a CPU interrupt request is generated when the RTIF bit is set, provided the I bit in the CCR is cleared.

1 = Interrupt enable
0 = Interrupt disable

RT1:RT0 - Rate Select

These two bits select one of four taps from the Real Time Interrupt circuit. The settings for RTI is listed in **Table 8-1**. Reset sets these RT0 and RT1, selecting the lowest periodic rate and therefore the maximum time in which to alter these bits if necessary.

NOTE

Care should be taken when altering RT0 and RT1 if the time-out period is imminent or uncertain. If the selected tap is modified during a cycle in which the counter is switching, an RTIF could be missed or an additional one could be generated. To avoid problems, the COP should be cleared before changing RTI taps.

8.2.3 Ctimer Counter Register (CTCR)

The Core Timer Counter Register is a read-only register which contains the current value of the 8-bit ripple counter at the beginning of the timer chain. This counter is clocked at f_{op} divided by 4 and can be used for various functions including a software input capture. Extended time periods can be attained using the TOF function to increment a temporary RAM storage location thereby simulating a 16-bit (or more) counter.

		7	6	5	4	3	2	1	0
CTCR $0009	READ	**CT7**	**CT6**	**CT5**	**CT4**	**CT3**	**CT2**	**CT1**	**CT0**
	WRITE								
	RESET	0	0	0	0	0	0	0	0

The power-on cycle clears the entire counter chain and begins clocking the counter. After 2016 cycles, the power-on reset circuit is released which again clears the counter chain and allows the device to come out of reset. At this point, if $\overline{\text{RESET}}$ is not asserted, the timer will start counting up from zero and normal device operation will begin. When $\overline{\text{RESET}}$ is asserted any time during operation (other than POR), the counter chain will be cleared.

8.2.4 Operation During Low Power Mode

The timer is cleared when going into STOP mode. When STOP is exited by an external interrupt or an external RESET, the internal oscillator will resume, followed by 2016 cycles internal processor stabilization delay. The timer is then cleared and operation resumes.

The CPU clock halts during the WAIT mode, but the timer remains active. If the interrupts are enabled, the timer interrupt will cause the processor to exit the WAIT mode.

Table 8-1. RTI and COP Rates at 1.8MHz Bus Frequency

	BUS FREQUENCY = 1.8 MHz		
RT1:RT0	**Div. Ratio**	**RTI Rate**	**COP Rate (RTIx7)**
00	2^{14}	9.1 ms	63.7 ms
01	2^{15}	18.2ms	127.4 ms
10	2^{16}	36.4 ms	254.8 ms
11	2^{17}	72.8 ms	509.6 ms

Table 8-2. RTI and COP Rates 17.5kHz Bus Frequency

	BUS FREQUENCY = 17.5 kHz		
RT1:RT0	**Div. Ratio**	**RTI Rate**	**COP Rate (RTIx7)**
00	2^{14}	0.94 s	6.58 s
01	2^{15}	1.88 s	13.16 s
10	2^{16}	3.76 s	26.32 s
11	2^{17}	7.52 s	52.64 s

8.3 ONE SECOND WATCH TIMER

The 1 second Watch Timer is a 15-bit free running counter which runs off the 32kHz external clock directly and is therefore not affected by Stop Mode or Wait Mode. The Watch Timer is used to generate interrupts at 1 second interval to wake up the CPU when the I-bit in the CCR is cleared. See **Figure 8-7**. The Watch Timer overflow flag WTOF is set when the a transition from $7FFF to $0000 occurs and will cause an interrupt if the WTOIE-bit of the Watch Timer control and status register (WTCSR $0023) is set to ‘1’. This flag is latched and should be cleared by the Watch Timer interrupt service routine to prevent unwanted interrupts. Writing a ‘0’ to the WTOF-bit or reset clears the WTOF-bit.

Power on or external reset has no effect on the free running counter and therefore the counter state is unknown after reset. However, the Watch Timer can be reset to a known state by writing a ‘1’ to the WTR-bit in the Watch Timer control and status register. The WTOF flag is prevented from being set during software reset. Reading the WTR-bit will always return a ‘0’.

The Watch Timer can be disabled by writing a ‘1’ to the WTOFF-bit of the Watch Timer control and status register, the value of the Watch Timer is retained during the off period.

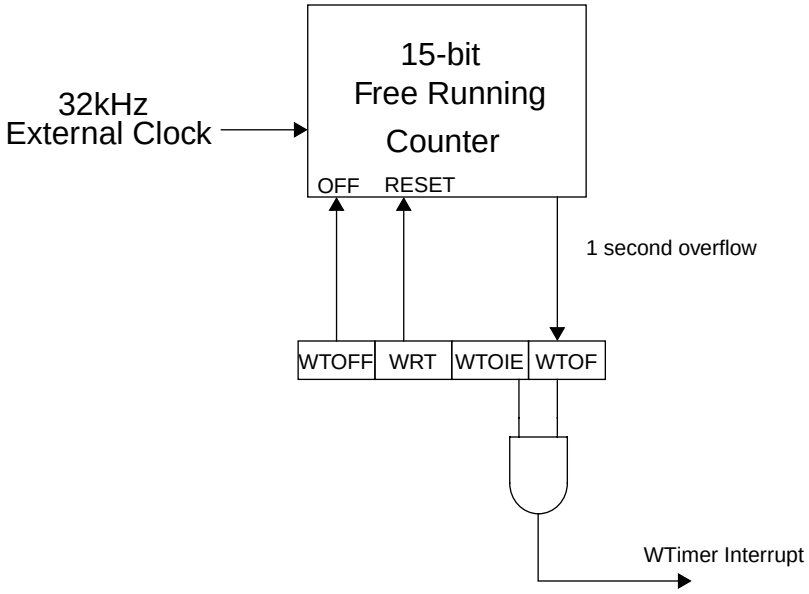


Figure 8-7. ONE Second Watch Timer

8.3.1 Watch Timer Control & Status Register (WTCSR)

	7	6	5	4	3	2	1	0
WTCSR $0023 READ					WTOFF	0	WTOIE	WTOF
WTCSR $0023 WRITE						WTR		
RESET	-	-	-	-	0	0	0	0

U = UNAFFECTED

WTOFF
- 1 = Disable Watch Timer
- 0 = Enable Watch Timer

WTR

Writing a ‘1’ to this bit will generate a reset pulse to reset the Watch Timer. Reading this bit will always return a ‘0’.

WTOIE
- 1 = Enable 1 second interrupt.
- 0 = Disable 1 second interrupt.

WTOF
- 1 = Flag set by transition of the 15-bit free running counter from $7FFF to $0000
- 0 = Flag cleared by writing a 0 to the WTOF bit.

SECTION 9
SERIAL PERIPHERAL INTERFACE

The term “serial peripheral” refers to the fact that this interface requires separate wires (signals) for data and clock. In this format, data does not contain an explicit clock. The SPI scheme may be used to interconnect microcomputers located at a short distance (usually within a single “black box” or on the same PC card). This may comprise a system of one microcomputer and several slaves or may be a system of microcomputers, each having the capability of either master or slave.

A practical system may include:

1) MISO master in slave out

2) MOSI master out slave in

3) SCK serial clock

4) $\overline{SS}$ (n) slave select(s)

9.1 SIGNAL DESCRIPTION

9.1.1 MISO Master In Slave Out

9.1.1.1 Slave Mode

MISO is the signal which is used in Slave Mode to present data from a Slave device to the bus. The MISO pin will be placed in the hi-Z state whenever a Slave device is “not” selected (SS=1) by the bus master. **Figure 9-1** shows the clock (SCK) and data relationship. Four possible timing relationships may be chosen by use of the control bits (CPOL) and (CPHA). The Slave device and a Master device must be programmed to be in similar timing modes for proper data transfer.

9.1.1.2 Master Mode

In the master mode (control bit MSTR=1), the function of MOSI and MISO are inverted within the device. Therefore, the MISO pin becomes the data input pin for a device which is in the master mode. (Also, the function of SCK switches from being an input for system clock to one of outputting the system clock.)

9.1.2 MOSI Serial Data In (Input)

9.1.2.1 Slave Mode

MOSI is the signal used to receive data from some Master device. **Figure 9-1** shows the serial clock and data timing relationship.

It should be noted that when a Master device transmits data to a second device via the MOSI line, the slave device (if it has the capability) will respond by sending data into the MISO pin of the master device. This implies full duplex transmission with both data-out and data-in synchronized to the same clock signal which is provided by the master. Moreover, the SAME shift register is used for data out and data in. Thus, the byte transmitted is replaced by the byte received, removing the need for separate status bits for XMIT EMPTY and REC FULL. A single status bit, SPIF, is used to signify IO operation complete.

9.1.2.2 Master Mode

As noted above, the function of MOSI and MISO are inverted in the master mode. The MOSI pin becomes the data output pin when the device is in the master mode. When a transfer of data is not taking place with a Slave device the Master drives the MOSI line high. The Master always allows the data onto the MOSI pin a half-cycle before the clock edge (SCK) needed for the Slave to latch the data internally.

9.1.3 SCK Serial Clock (In/Out)

9.1.3.1 Slave Mode

The serial clock is used to move data both in and out of the device through its MOSI and MISO pins. The Master and Slave device are capable of exchanging a byte of information during a sequence of eight clock pulses if wired to do so. In the slave mode, the SCK pin becomes an input for the external clock being sent from the Master device. In this case SCK is asynchronous to the Slave device's phase 1-2 clocks and read/write control, therefore synchronization must take place prior to transmission and after reception. This must be done on the byte level. The type of clock and its' relationship with the data is controlled by bits CPOL and CPHA. Reference **Figure 9-1**. The clock rate control bits SPR1 and SPR0 have no function while the part is in the Slave mode.

9.1.3.2 Master Mode

In this mode, the clock is generated within the Master device by a circuit driven

from the bus clock. The clock rate is selected by bits (SPR1,SPR0) in the control register. The SCK pin on the Master device becomes a fixed output providing the system clock to an enabled Slave or Slaves. The clock is used by the Master to latch incoming Slave data on the MISO pin and shift out data to the Slave device on the MOSI pin. The Master and Slave must be operated in the same timing mode. The type of clock and its' relationship with the data is controlled by bits CPOL and CPHA. Reference **Figure 9-1**.

9.1.4 $\overline{SS}$ SLAVE SELECT (INPUT)

9.1.4.1 Slave Mode

The slave select ($\overline{SS}$ input) is generated by the master (parallel port may be used) and used to "enable one" of several slaves to accept and/or return data or "enable several" slaves to accept data. To insure a data byte transfer, the SS signal must be low prior to occurrence of SCK and must not become high until after the 8th (last) SCK cycle. **Figure 9-1** shows the clock (SCK) and data relationship. Depending on the state of the CPHA control bit, the SS pin pulled low: (1) allows the first bit of data onto the MISO system line for transfer and (2) prevents the Slave from reading or writing the data register. A further description of the affect of the (SS) pin and (CPHA) control bit on the i/o data register is given in the description of the (WCOL) status flag. The (WCOL) flag warns the Slave if it has had a conflict between a transmission and a write of the data register. A high level on SS forces MISO to the hi-Z state. Also, SCK and MOSI are ignored by the disabled slave.

9.1.4.2 Master Mode

In this mode, Slave Select (SS) input is monitored to assure that it stays false (high). If Slave Select becomes true, the device immediately exits the master mode and becomes a slave (MSTR=0). Also, control bit (SPE) is forced to a zero causing all SPI system pins to be inputs. An interrupt flag (MODF) is set warning the device that the above events have occurred. The significance of this is that a collision has occurred; that is, two devices have both become masters. This is normally the result of software error, although some systems may allow the default master to "knock all other masters off the bus" if an erroneous bus state is detected. This is, of course, a catastrophic event and it is the responsibility of the default master to completely "clean up" the system.

9.2 SPI REGISTERS

The register addresses only show the low order address bits i.e ABL(1:0). The registers can be placed anywhere in the device memory map by generating an appropriate '**M**odule **S**elect' signal in the map logic.

9.2.1 SPCR SPI Control Register

		7	6	5	4	3	2	1	0
SPCR $0020	READ	SPIE	SPE	-	MSTR	CPOL	CPHA	SPR1	SPR0
	WRITE								
	RESET	0	0	-	0	0	1	U	U

SPIE SPI Interrupt Enable

When this bit is set to a one a hardware interrupt sequence is requested each time the SPIF or MODF status flag is set. SPI interrupts are inhibited if this bit is clear or if the I-bit in the CC-Register is one.

SPE SPI System Enable

Setting the SPE-bit to '1' will enable the SPI function and automatically configure PTC[3:0] as dedicated SPI pins.

MSTR Master/Slave Mode Select

0 = Slave mode
1 = Master mode

CPOL Clock Polarity

CPHA Clock Phase

These two bits are used to specify the clock format to be used in SPI operations. Please refer to **Figure 9-1**.

SPR1, SPR0 SPI Clock (SCK) Rate Select Bits

These bits are used to specify the SPI clock rate.

Table 9-1. SPI Clock Rates

SPR1	SPR0	E Clock divided-by	Frequency Eclk = 1.8 MHz (baud rate)
0	0	2	0.9 MHz
0	1	4	450 kHz
1	0	16	112.5 kHz
1	1	32	56.25 kHz

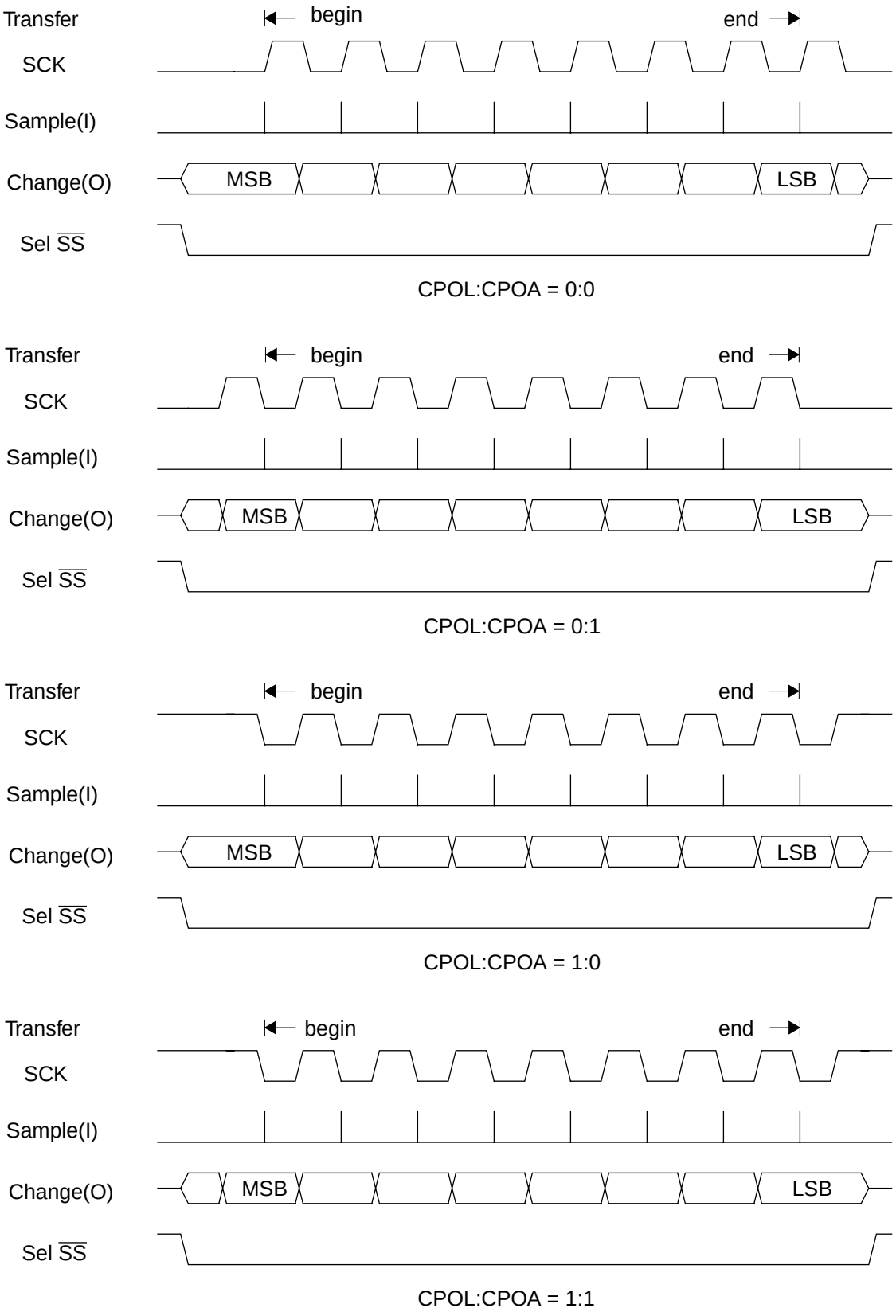


Figure 9-1. SPI Clock/Data Relationships

9.2.2 SPSR SPI Status Register

		7	6	5	4	3	2	1	0
SPSR $0021	READ	SPIF	WCOL	0	MODF	0	0	0	0
	WRITE								
	RESET	0	0	0	0	0	0	0	0

SPIF SPI Interrupt Request

SPIF is set after the eighth SCK cycle in a data transfer and it is cleared by reading the SPSR register (with SPIF set) followed by an access (read or write) to the SPI Data Register.

WCOL Write Collision Status Flag

This error status flag is used to indicate that a serial transfer was in progress when the MCU tried to write new data into the SPDR data register. The MCU write is disabled to avoid writing over the data being transmitted. No interrupt is generated because the error status flag can be read upon completion of the transfer that was in progress at the time of the error. This flag is automatically cleared by a read of the SPSR (with WCOL set) followed by an access (read or write) to the SPDR register.

MODF SPI Mode Error Interrupt Status Flag

This bit is set automatically by SPI hardware if the MSTR control bit is set to one and the Slave Select input pin becomes zero. This condition is not permitted in normal operation. This flag is automatically cleared by a read of the SPSR (with MODF set) followed by a write to the SPCR register.

9.2.3 SPDR SPI Data Register

		7	6	5	4	3	2	1	0
SPDR $0022	READ	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	WRITE								
	RESET	U	U	U	U	U	U	U	U

This 8-bit register is both the input and output register for SPI data. In the SPI system the 8-bit data register in the master and the 8-bit data register in the slave are linked by the MOSI and MISO wires to form a distributed 16-bit register. When a data transfer operation is performed, this 16-bit register is serially shifted eight bit positions by the SCK clock from the master so the data is effectively

exchanged between the master and the slave. Note that some slave devices are very simple and either accept data from the master without returning data to the master or pass data to the master without requiring data from the master.

When writing the SPDR, the data is written directly into the shift register and always shifted out MSB first.

When reading the SPDR, a Read Data Buffer is actually accessed. This buffer contains the last data byte received by the SPI, and is updated during the cycle that SPIF is set (reception complete).

SECTION 10
ANALOG TO DIGITAL CONVERTER

The MC68HC05CL48 includes a 8-channel, 8-bit, multiplexed input, successive approximation A/D converter, with four of the input channels available on external pins and another four internal channels are used for calibration purpose.

10.1 ANALOG SECTION

10.1.1 Ratiometric Conversion

The A/D is ratiometric, with two dedicated pins supplying the reference voltages (V_{RH} and $V_{RL)}$. An input voltage equal to V_{RH} converts to $FF (full scale) and an input voltage equal to V_{RL} converts to $00. An input voltage greater than V_{RH} will convert to $FF with no overflow indication. For ratiometric conversions, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}.

10.1.2 V_{RH} and V_{RL}

V_{RH} and V_{RL} are generated internally on chip to reduce pin counts.

10.1.3 Accuracy And Precision

The 8-bit conversions shall be accurate to within $\pm 1^1/_2$ LSB including quantization when averaged over four readings.

10.2 CONVERSION PROCESS

The A/D reference inputs are applied to a precision internal digital-to-analog converter. Control logic drives this D/A and the analog output is successively compared to the selected analog input which was sampled at the beginning of the conversion time. The conversion process is monotonic and has no missing codes.

10.3 DIGITAL SECTION

10.3.1 Conversion Time

Each channel of conversion takes 32 clock cycles, which must be at a frequency equal to or greater than 1 MHz.

10.3.2 Multi-Channel Operation

In User Mode a multiplexer allows the single A/D converter to select one of eight analog signals, two of which are V_{RH} and V_{RL}. The ATD3:0 are input signals to the multiplexer.

10.4 A/D STATUS AND CONTROL REGISTER (ADCSR)

The following paragraphs describe the function of the A/D Status and Control Register.

		7	6	5	4	3	2	1	0
ADSCR $0024	READ	COCO	ADRC	ADON	0		CH2	CH1	CH0
	WRITE								
	RESET	U	U	U	U	U	U	U	U

10.4.1 COCO - Conversions Complete

This read-only status bit is set when a conversion is completed, indicating that the A/D Data Register contains valid results. This bit is cleared whenever the A/D Status and Control Register is written and a new conversion automatically started, or whenever the A/D Data Register is read. Once a conversion has been started by writing to the A/D Status and Control Register, conversions of the selected channel will continue every 32 cycles until the A/D Status and Control Register is written again. In this continuous conversion mode the A/D Data Register will be filled with new data, and the COCO bit set, every 32 cycles. Data from the previous conversion will be overwritten regardless of the state of the COCO bit prior to writing.

10.4.2 ADRC - A/D RC Oscillator Control

When the RC oscillator is selected (ADRC = 1) to be the A/D clock source, it requires a time t_{ADRC} to stabilize. Results can be inaccurate during this time. If the CPU clock is running below 1 Mhz, the RC oscillator must be used.

When ADRC =0, the A/D uses the CPU clock.

10.4.3 ADON - A/D On

When the A/D is turned on (ADON = 1), it requires a time t_{ADON} for the current sources to stabilize, and results can be inaccurate during this time. This bit turns on the charge pump. If the ADRC is set, clearing this bit disables the RC oscillator to save power.

10.4.4 CH2:CH0 - Channel Select Bits

CH2, CH1, and CH0 form a 3-bit field which is used to select one of eight A/D channels. Channels 0-3 correspond to the analog input pins: ATD(3:0) in the MCU. Channels 4-7 are used for internal reference points. The following table shows the signals selected by the channel select field.

CHANNEL	SIGNAL
0	ATD0 pin
1	ATD1 pin
2	ATD2/PTC[4] pin
3	ATD3/PTC[5] pin
4	V_{RH}
5	V_{RL}
6	$(V_{RH} - V_{RL})/4$
7	$(V_{RH} - V_{RL})/2$

Table 10-1. A/D Channel Assignments

10.5 A/D DATA REGISTER (ADDR)

One 8-bit result register is provided. This register is updated each time COCO is set

		7	6	5	4	3	2	1	0
ADDR $0025	READ	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
	WRITE								
	RESET	U	U	U	U	U	U	U	U

10.6 A/D DURING WAIT MODE

The A/D continues normal operation during WAIT mode. To decrease power consumption during WAIT, it is recommended that both the ADON and ADRC bits in the A/D Status and Control Register be cleared if the A/D converter is not being used. If the A/D converter is in use and the system clock rate is above 1.0 MHz, it is recommended that the ADRC bit be cleared.

10.7 A/D DURING STOP MODE

In STOP mode the comparator and charge pump are turned off and the A/D ceases to function. Any pending conversion is aborted. When the clocks begin oscillation upon leaving the STOP mode, a finite amount of time passes before the A/D circuits stabilize enough to provide conversions to the specified accuracy. The delays built into the MC68HC05 when coming out of STOP mode are sufficient for this purpose, therefore no explicit delays need to be built into the software.

SECTION 11
CALLER-ID

The Caller ID module demodulates the Bell 202 and CCITT V.23 1200 baud FSK asynchronous data. This module consists of four major building blocks - **FSK demodulator, Carrier Detect, Ring Detect** and the **Power Management** circuit. The block diagram of this module is shown in **Figure 11-1**.

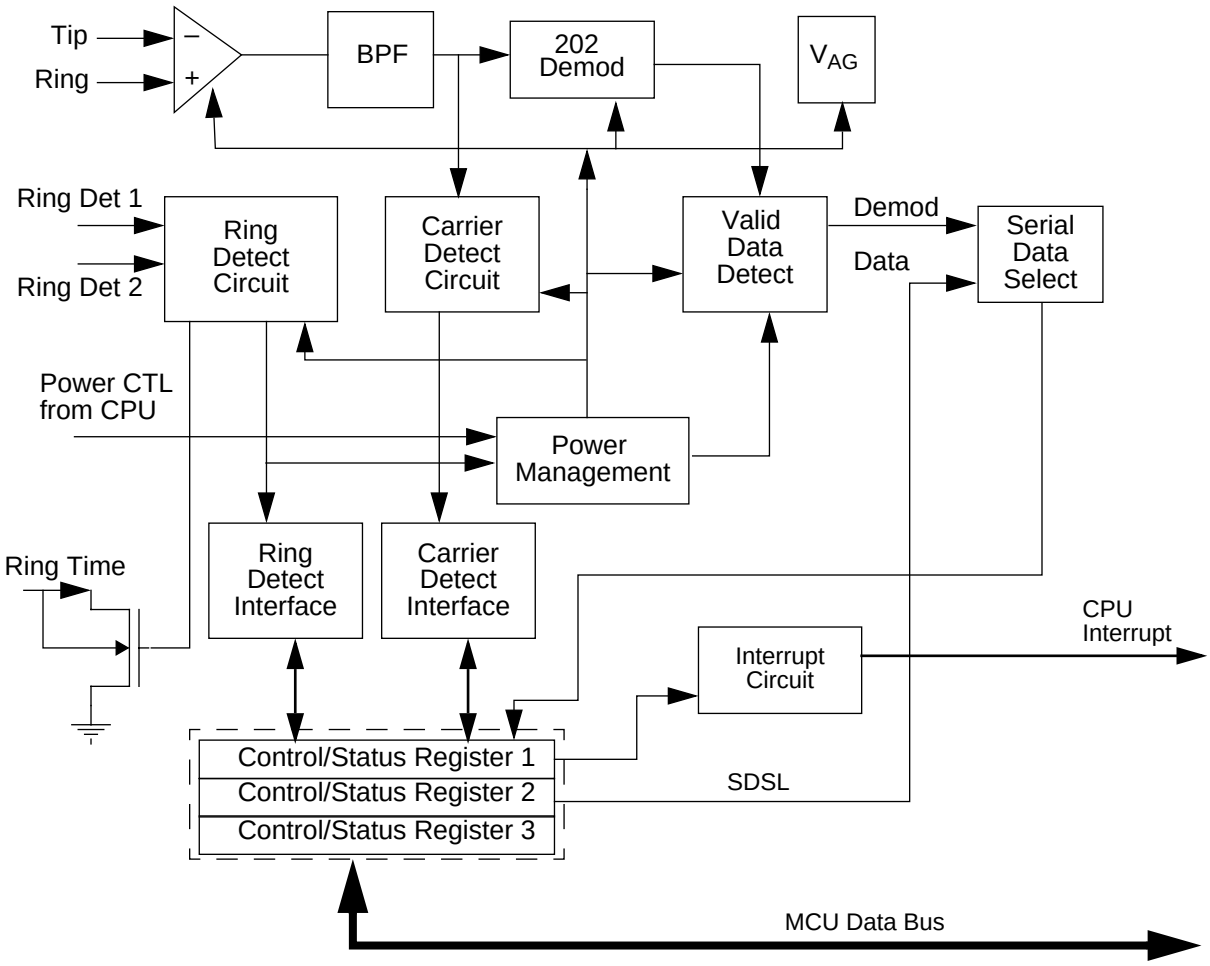


Figure 11-1. Caller ID Block Diagram

11.1 FSK DEMODULATOR

The recovered signal consists of both the channel seizure information and the message words. The original serial raw data is made available via the MCU registers. See **Section 11.7.2**.

11.2 CARRIER DETECTOR

The Carrier Detect block will validate the carrier signal from the filter section. The asynchronous carrier signal is considered valid if present for a minimum of 25ms. A carrier dropout is confirmed if it is silent for more than 8ms. The carrier detect signal will remain low until a dropout condition is detected. The carrier detect output is available in a read-only register ($\overline{CD}$) in Control/Status Register 2 (CLCSR2). It can also be overwritten by writing to CDO (Carrier Detect Override) in the Control/Status Register 1 (CLCSR1) when enabled by writing a '1' to CDOE (Carrier Detect Override Enable) in the CLCSR3 register. A valid carrier can also produce an interrupt to the CPU when enabled by the CDIE bit in CLCSR1. See **Section 11.7.1** and **Section 11.7.2**

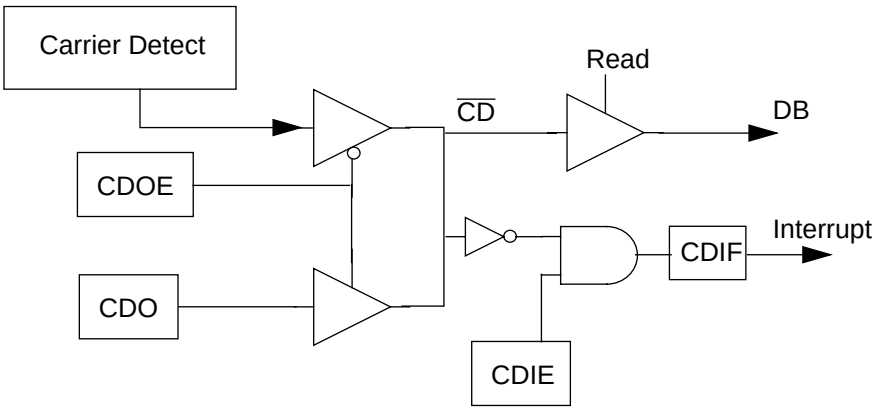


Figure 11-2. Carrier Detect Interface

11.3 RT_L INTERRUPT

Figure 11-3 shows that the RT_L pin connected to a ring time circuit. When RDI1 is 'low', RT_L is pulled high by the external pull-up resistor. When RDI1 rises above 1.0 volt, RT_L will be pulled low by the open-drain NMOS transistor and an interrupt will be generated.

Note: RT_L interrupt is permanently disabled in the current design and can only be enabled by a Metal Mask Option.

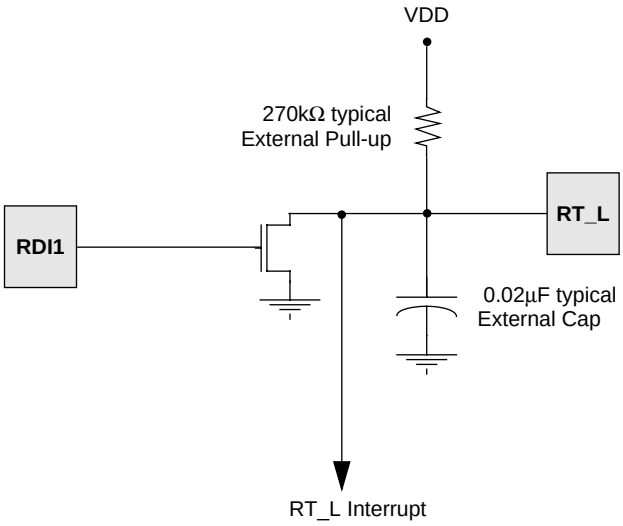


Figure 11-3. RT_L Interrupt

11.4 RING DETECTOR

The ring detect circuit validates the input ring signal (RD2) and the ring detect output is available in a read-only register ($\overline{RD}$) in Control/Status Register 2 (CLCSR2). It can also be overwritten by writing to RDO (Ring Detect Override) in the Control/Status Register 1(CLCSR1) when enabled by writing a '1' to RDOE (Ring Detect Override Enable) in the CLCSR3 register. A valid ring signal can also produce an interrupt to the CPU when enabled by the RDIE bit in CLCSR1. See **Section 11.7.1** and **Section 11.7.2**.

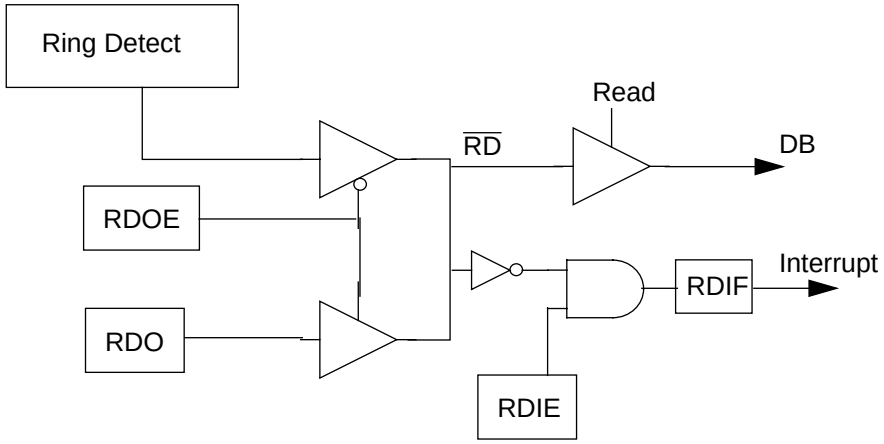


Figure 11-4. Ring Detect Interface

11.5 POWER MANAGEMENT

If the Ring Detect module is used, it should be enabled by setting the RDPW bit before executing the STOP instruction. When the RT signal is below the threshold (see **Figure 11-9**) the oscillator circuit is forced on and the Ring Signal is validated. If the RT rises above the threshold before the Ring Signal is validated the oscillator will stop and the MCU will stay in the STOP mode. However, if a valid ring is detected an interrupt is generated provided the Ring Detect Interrupt Enable bit (RDIE) is set. The interrupt will wake the MCU from the STOP mode and the clocks to all enabled modules will start. At this time if the CPU is not required the WAIT mode can be entered.

If the Carrier Detect module is enabled before entering the wait mode by CDPW bit, it will start processing the incoming data. When a valid carrier is detected an interrupt is generated if enabled by the CDIE bit. The interrupt will take the CPU out of the WAIT mode.

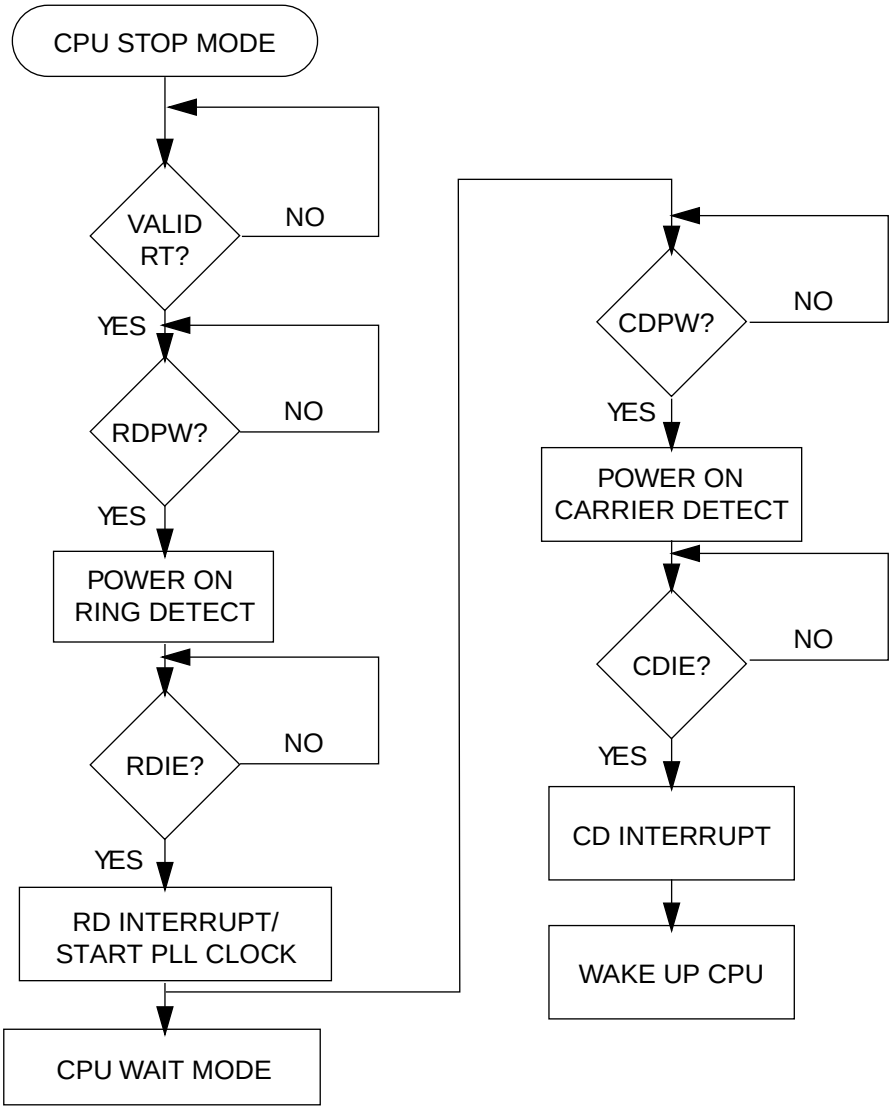


Figure 11-5. Power Up Sequence from STOP Mode

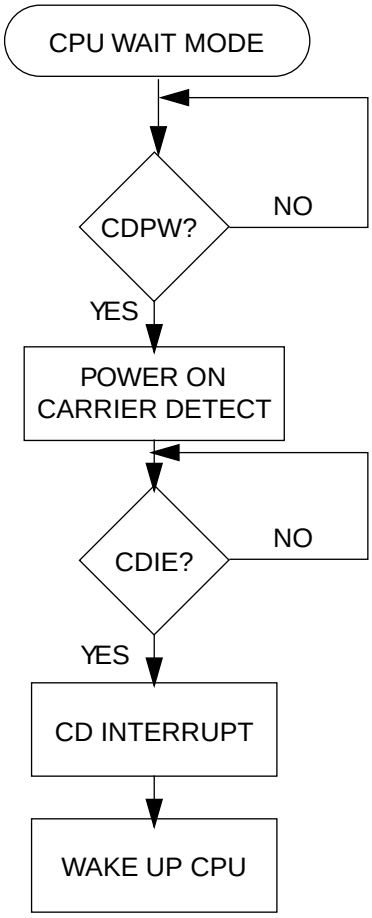


Figure 11-6. Power Up Sequence from WAIT Mode

11.6 DATA INTERFACE

The demodulated data from this module is available in bit-serial and 8-bit format. The serial data can be read from CIDSD (bit 2) of the CLCSR2 register and the 8-bit data is available from the Caller_ID data register CDDR. This data includes the alternate 0 and 1 pattern, 150 ms marking which precedes data. At all other times the demodulator output bit is high.

The Caller_ID data ready flag CDRF bit-2 of CLCSR3 will be set to indicate that the 8-bit data is ready to be read. The data ready flag will generate an CPU interrupt if the data ready interrupt enable DRIE bit-3 of CLCSR3 is set and the I-bit of the CCR is cleared. The Caller_ID data register CDDR acts as data buffer for the serial to parallel data conversion register and should be read by the CPU within 8.3msec after receiving the interrupt. Failure to do so will result data lost. Reading the Caller_ID data register CDDR clears the Caller_ID data ready flag. **Figure 11-9** shows the timing diagram for the serial to parallel data conversion.

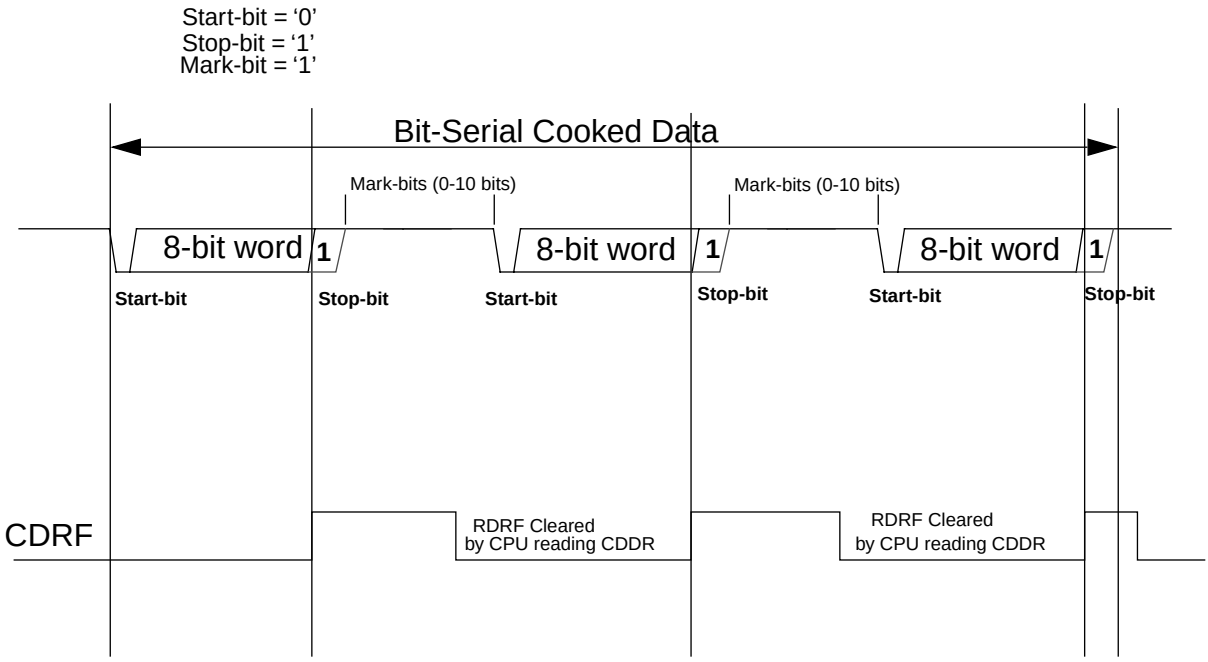


Figure 11-7. 8-bit Caller ID Data Timing Diagram

11.7 CALLER-ID REGISTERS

11.7.1 Control/Status Register1 (CLCSR1)

		7	6	5	4	3	2	1	0
CLCSR1 $000A	READ	RDIF	RDIE	CDIF	CDIE			RDO	CDO
	WRITE	0		0					
	RESET	0	0	0	0	U	U	1	1

CDO, BIT0

Carrier Detect Override, when enabled by the CDOE bit in the CLCSR3 register, the carrier detect can be forced by writing a zero to this bit. On reset this bit is set to one.

RDO, BIT1

Ring Detect Override, when enabled by RDOE in the CLCSR3 register, the ring detect can be forced by writing a zero to this bit. On reset this bit is set to one.

CDIE, Bit4 (Carrier Detect Interrupt Enable)

When set enables the carrier detect interrupt to be generated when a carrier is detected or forced by writing to CDO.

CDIF, BIT5 (Carrier Detect Interrupt Flag)

Provided the carrier detect interrupt is enabled by setting the CDIE, this bit is set when the carrier is detected. When this flag is one an interrupt is generated. This bit shares the interrupt vector address with Ring Detect interrupt. The CDIF bit must be cleared by writing a zero.

RDIE, Bit6 (Ring Detect Interrupt Enable)

When set enables the ring detect interrupt to be generated when a ring is detected or forced by writing to RDO.

RDIF, BIT7 (Ring Detect Interrupt Flag)

Provided the ring detect interrupt is enabled by setting the RDIE, this bit is set when the ring is detected. When this flag is one an interrupt is generated. This bit shares the interrupt vector address with Carrier Detect interrupt. The RDIF bit must be cleared by writing a zero.

11.7.2 Control/Status Register 2 (CLCSR2)

		7	6	5	4	3	2	1	0
CLCSR2 $000B	READ		RDPW	CDPW			CIDSD	$\overline{RD}$	$\overline{CD}$
	WRITE								
	RESET	U	0	0	U	U	U	1	1

U = UNAFFECTED

$\overline{CD}$, Bit0 (Carrier Detect)

This read only bit returns the value of the Carrier Detect signal which goes low when a valid carrier is detected and remains low while the carrier remains valid.

$\overline{RD}$, Bit1 (Ring Detect)

This read only bit returns the value of the Ring Detect signal which goes low when a valid ringing signal is detected and remains low as long as the ringing signal remains valid.

CIDSD, Bit2 (Caller ID Serial Data)

This read only bit returns the value of the Caller ID Serial Data (output of the on chip demodulator) whenever the $\overline{CD}$ is low. This data includes the alternate 0 and 1 pattern, 150 ms marking which precedes the data. At all other times the demodulator output is high.

CDPW, Bit5 (Carrier Detect Power Up)

This bit when set is used to enable carrier detection. An interrupt will be generated when a valid carrier is detected and if the CDIE bit in the CLCSR1 register is set.

RDPW, Bit6 (Ring Detect Power Up)

This bit when set is used to enable ring detection. An interrupt will be generated when a valid ring is detected and if the RDIE bit in the CLCSR1 register is set.

11.7.3 Control/Status Register 3 (CLCSR3)

		7	6	5	4	3	2	1	0
CLCSR3 $000C	READ	SDSL			CDRE	DRIE	CDRF	RDOE	CDOE
	WRITE								
	RESET	0	U	U	U	0	0	0	0

U = UNAFFECTED

CDOE, Bit-0

Carrier Detect Override Enable, when set this bit disables the carrier detect module output and allows the user to force the carrier detect signal by writing a zero to CDO, bit 0 in the CLCSR1 register.

RDOE, Bit-1

Ring Detect Override Enable, when set this bit disables the ring detect module output and allows the user to force the ring detect signal by writing a zero to RDO, bit 1 in the CLCSR1 register.

CDRF, Bit-2

Caller_ID data ready flag. This is set to ‘1’ when the 8-bit data is ready to be read. Reading CDDR ($000E) clears this bit.

DRIE, Bit-3 (Data Ready Interrupt Enable)

1 = Enable CDRF to cause an interrupt.
0 = Disable CDRF from generating an interrupt.

CDRE, Bit-4 (Caller_ID Receiver Enable)

1 = Enable Caller_ID 8-bit data receiver.
0 = Disable Caller_ID 8-bit data receiver.

SDSL, Bit-7 (Serial Data Select)

0 = Select cooked data to be routed to the Caller ID parallel data receiver.
1 = Select raw data to be routed to the Caller ID parallel data receiver.

11.7.4 Caller_ID Data Register (CDDR)

CDDR[7:0] is the Caller ID data in byte format.

		7	6	5	4	3	2	1	0
CDDR $000E	READ	CDD7	CDD6	CDD5	CDD4	CDD3	CDD2	CDD1	CDD0
	WRITE								
	RESET	U	U	U	U	0	0	0	0

U = UNAFFECTED

11.8 DESIGN PARAMETERS

The data signalling interface conforms to the recommended operating ranges of the physical layer test parameters for TYPE 1 CPE as described in Bellcore Publication SR-NWT-003004.

Table 11-1. Typical Input parameter

Parameters	Operating Range	Units
Mark Frequency	1188 ~ 1212	Hz
Space Frequency	2178 ~ 2222	Hz
Mark Level (600 ohm)	-12 ~ -32	dBm
Space Level (600 ohm)	-12 ~ -36	dBm
Carrier Frequency	1700	Hz
Twist Immunity (600 ohm)	+/- 10	dBm
Baud Rate	1188 ~ 1212	baud
Ringing Frequency	10 to 55	Hz
Noise Immunity (Signal to Noise Ratio)	-20 (for noise below 200 and above 3200 Hz) 25 (for noise between 200 and 3200 Hz)	dB
Channel Seizure Delay	250 ~ 3600	ms
Input Impedance	500	kΩ
Immunity to CS and MS	10	ms

Table 11-2. Critical Design Characteristic

Characteristics	Typ	Unit
Input Tip/Ring Sensitivity (600 ohm)	-40	dBm
Ring/Tip Common Mode Rejection Ratio	60	dB
Bandpass Filter (BPF) Frequency Response (relative to 1700 Hz @ 0 dB) 60 Hz 1000 Hz 2400 Hz >=3300 Hz	 -58 -1 -1 -34	dB
Carrier Detect Sensitivity (600 ohm)	-40 (-48 min)	dBm
Ring Detect Threshold (RDI2)	0.39*Vdd ± 0.1volt	Volts
RDI Threshold to keep RT_L='1'	< 1.0	Volts

Table 11-3. Switching Characteristics (VDD= 5V; TA=25 C)

Description	Symbol	Min	Typ	Max	Unit
PLL Clock Start-up Time	t_{DOSC}	-	-	10	ms
Carrier Detect Acquisition	t_{DAQ}	-	14	25	ms
End of Carrier Detect	t_{DCH}	8	-	-	ms

The transmission level from the terminating C.O will be -13.5 dBm +/- 1.0. The expected worst case attenuation through the loop is expected to be -20dB. The receiver therefore, should have a sensitivity of approximately -34.5 dB to handle the worst case installations.

11.9 MESSAGE FORMAT

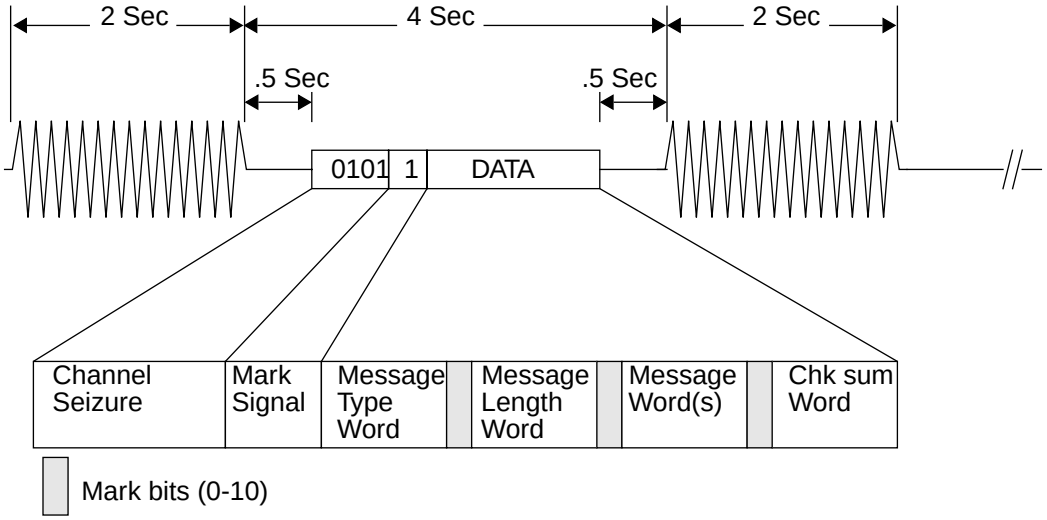


Figure 11-8. Single Message Format

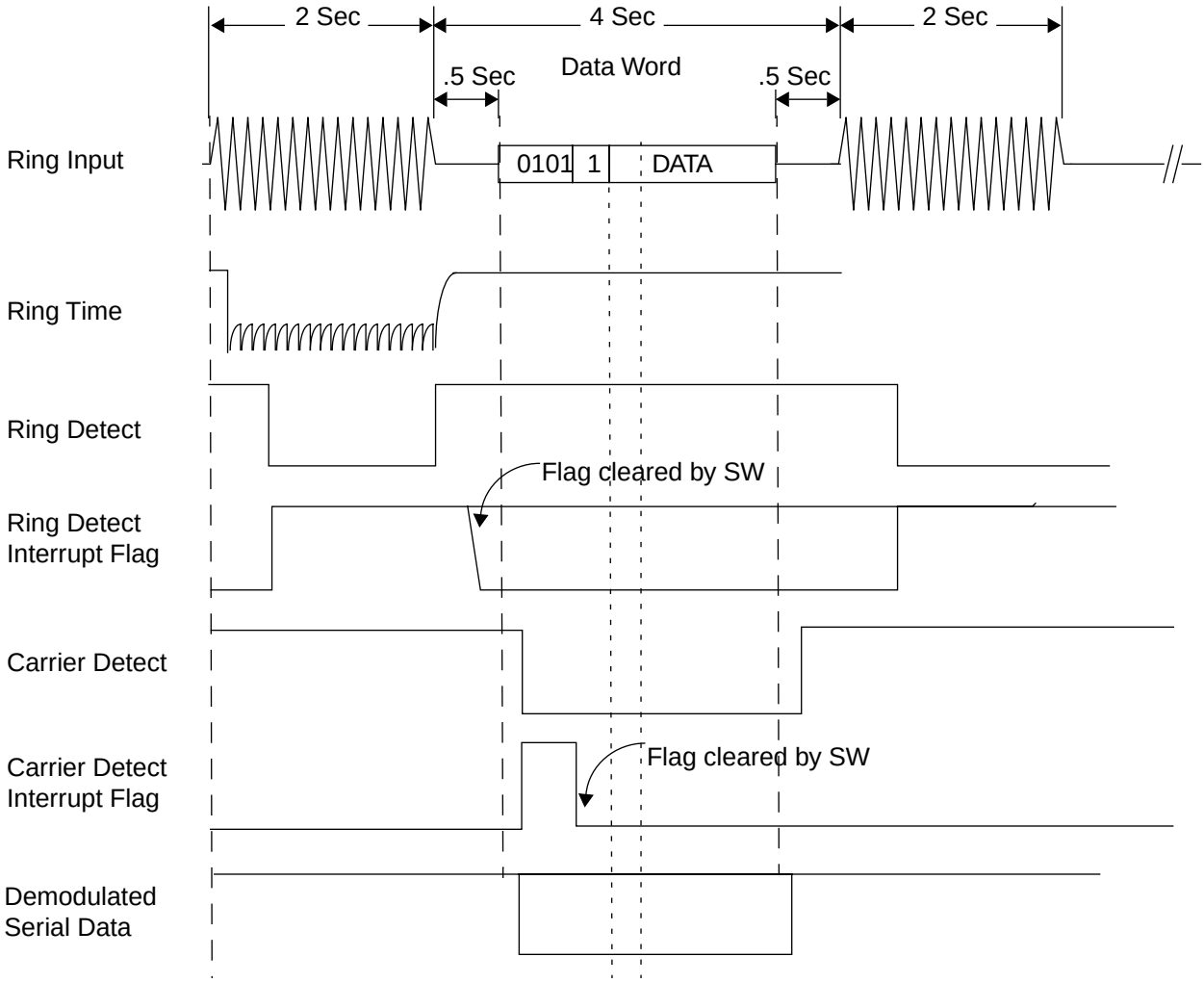


Figure 11-9. CLID Timing Diagram

SECTION 12
LCD DRIVER

The LCD driver module supports a 45 frontplanes by 16 backplanes or a 53 frontplanes by 8 backplanes display. This allows a maximum of 720 LCD segments to be driven. Each segment is controlled by a corresponding bit in the LCD RAM. On reset or on power-up, the drivers are disabled via a Display on (DISON) bit in the LCD Control (LCDCTR) register. **Figure 12-1** shows a block diagram of the LCD subsystem.

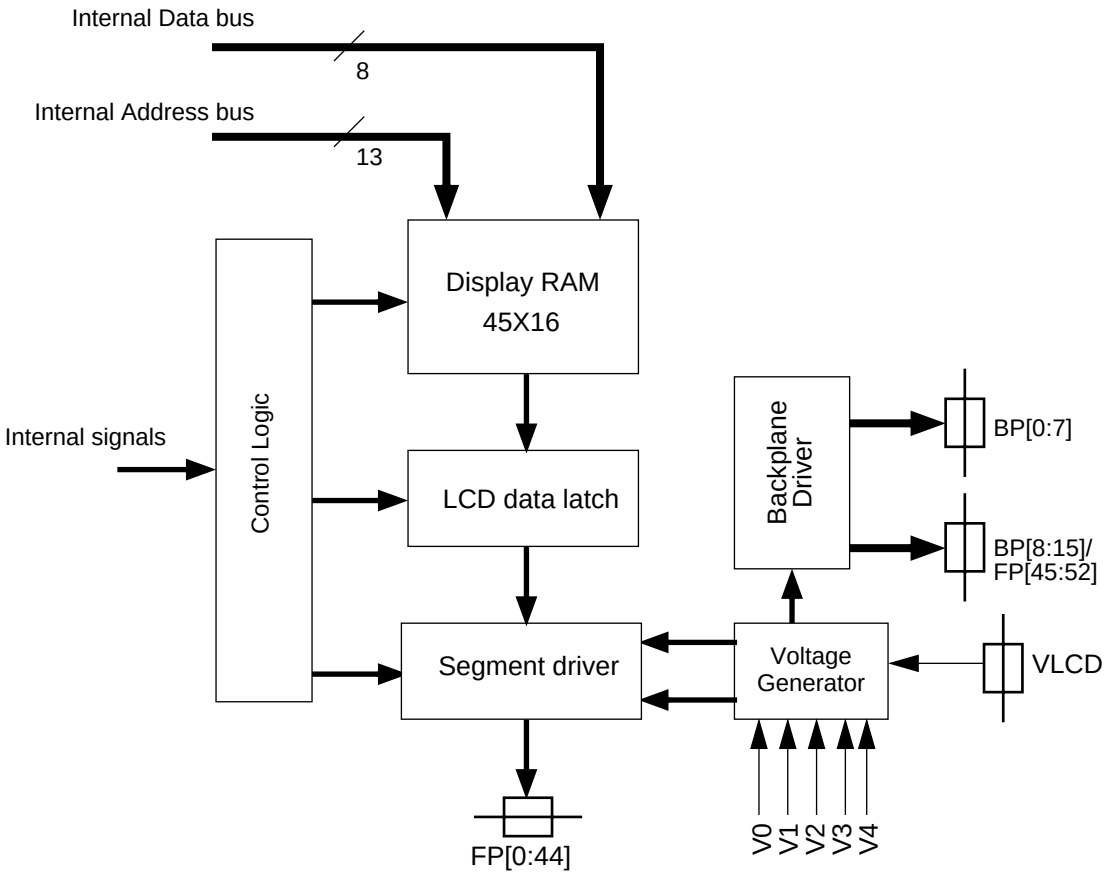


Figure 12-1. LCD Driver Block Diagram

12.1 LCD RAM.

The data to be displayed by the LCD is written to a 90 byte display RAM located at locations $0930 thru $095C and $0A30 thru $0A5C in the memory map. The bits are organized according to **Table 12-1** and **Table 12-2**, with a 1 stored in a given location resulting in the corresponding display segment being activated. The LCD RAM is a dual port RAM that interfaces with the internal address and data buses of the MCU. It is possible to read from LCD RAM locations for scrolling purposes.

When the display is disabled, the LCD RAM can be used as on-chip RAM.

Table 12-1. LCD RAM Organization

ADDR	DATA							
	0	1	2	3	4	5	6	7
$0930	FP0-BP0	FP0-BP1	FP0-BP2	FP0-BP3	FP0-BP4	FP0-BP5	FP0-BP6	FP0-BP7
$0931	FP1-BP0	FP1-BP1	FP1-BP2	FP1-BP3	FP1-BP4	FP1-BP5	FP1-BP6	FP1-BP7
$0932	FP2-BP0	FP2-BP1	FP2-BP2	FP2-BP3	FP2-BP4	FP2-BP5	FP2-BP6	FP2-BP7
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
$095A	FP42-BP0	FP42-BP1	FP42-BP2	FP42-BP3	FP42-BP4	FP42-BP5	FP42-BP6	FP42-BP7
$095B	FP43-BP0	FP43-BP1	FP43-BP2	FP43-BP3	FP43-BP4	FP43-BP5	FP43-BP6	FP43-BP7
$095C	FP44-BP0	FP44-BP1	FP44-BP2	FP44-BP3	FP44-BP4	FP44-BP5	FP44-BP6	FP44-BP7

Table 12-2. LCD RAM Organization

ADDR	DATA							
	0	1	2	3	4	5	6	7
$0A30	FP0-BP8	FP0-BP9	FP0-BP10	FP0-BP11	FP0-BP12	FP0-BP13	FP0-BP14	FP0-BP15
	FP45-BP0	FP45-BP1	FP45-BP2	FP45-BP3	FP45-BP4	FP45-BP5	FP45-BP6	FP45-BP7
$0A31	FP1-BP8	FP1-BP9	FP1-BP10	FP1-BP11	FP1-BP12	FP1-BP13	FP1-BP14	FP1-BP15
	FP46-BP0	FP46-BP1	FP46-BP2	FP46-BP3	FP46-BP4	FP46-BP5	FP46-BP6	FP46-BP7
$0A32	FP2-BP8	FP2-BP9	FP2-BP10	FP2-BP11	FP2-BP12	FP2-BP13	FP2-BP14	FP2-BP15
	FP47-BP0	FP47-BP1	FP47-BP2	FP47-BP3	FP47-BP4	FP47-BP5	FP47-BP6	FP47-BP7
$0A33	FP3-BP8	FP3-BP9	FP3-BP10	FP3-BP11	FP3-BP12	FP3-BP13	FP3-BP14	FP3-BP15
	FP48-BP0	FP48-BP1	FP48-BP2	FP48-BP3	FP48-BP4	FP48-BP5	FP48-BP6	FP48-BP7
$0A34	FP4-BP8	FP4-BP9	FP4-BP10	FP4-BP11	FP4-BP12	FP4-BP13	FP4-BP14	FP4-BP15
	FP49-BP0	FP49-BP1	FP49-BP2	FP49-BP3	FP49-BP4	FP49-BP5	FP49-BP6	FP49-BP7
$0A35	FP5-BP8	FP5-BP9	FP5-BP10	FP5-BP11	FP5-BP12	FP5-BP13	FP5-BP14	FP5-BP15
	FP50-BP0	FP50-BP1	FP50-BP2	FP50-BP3	FP50-BP4	FP50-BP5	FP50-BP6	FP50-BP7
$0A36	FP6-BP8	FP6-BP9	FP6-BP10	FP6-BP11	FP6-BP12	FP6-BP13	FP6-BP14	FP6-BP15
	FP51-BP0	FP51-BP1	FP51-BP2	FP51-BP3	FP51-BP4	FP51-BP5	FP51-BP6	FP51-BP7
$0A37	FP7-BP8	FP7-BP9	FP7-BP10	FP7-BP11	FP7-BP12	FP7-BP13	FP7-BP14	FP7-BP15
	FP52-BP0	FP52-BP1	FP52-BP2	FP52-BP3	FP52-BP4	FP52-BP5	FP52-BP6	FP52-BP7
$0A38	FP8-BP8	FP8-BP9	FP8-BP10	FP8-BP11	FP8-BP12	FP8-BP13	FP8-BP14	FP8-BP15
$0A39	FP9-BP8	FP9-BP9	FP9-BP10	FP9-BP11	FP9-BP12	FP9-BP13	FP9-BP14	FP9-BP15
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
$0A5A	FP42-BP8	FP42-BP9	FP42-BP10	FP42-BP11	FP42-BP12	FP42-BP13	FP42-BP14	FP42-BP15
$0A5B	FP43-BP8	FP43-BP9	FP43-BP10	FP43-BP11	FP43-BP12	FP43-BP13	FP43-BP14	FP43-BP15
$0A5C	FP44-BP8	FP44-BP9	FP44-BP10	FP44-BP11	FP44-BP12	FP44-BP13	FP44-BP14	FP44-BP15

12.2 LCD OPERATION

Figure 12-2 shows the backplane waveforms and some examples of frontplane waveforms which are dependent on the LCD segments to be driven as defined in the LCD RAM. Each “on” segment must have a differential driving voltage (BP-FP) applied to it once in each frame; the LCD driver module hardware uses the data in the LCD RAM to construct the frontplane waveform to meet this criterion. The backplane waveforms are continuous and repetitive (every 2 frames); they are fixed and not affected by the data in the LCD RAM. During WAIT mode the LCD drivers function as normal and will keep the display active if the DISON bit (bit0 of $07) is set.

The LCD drivers can be configured to operate with either 16 backplanes or 8 backplanes under software control.

The bias ratio can be set 1:4 or 1/5 under software control for both a 8 and 16 backplane LCD. The voltage levels required are generated internally by a resistive divider between VLCD and VSS.

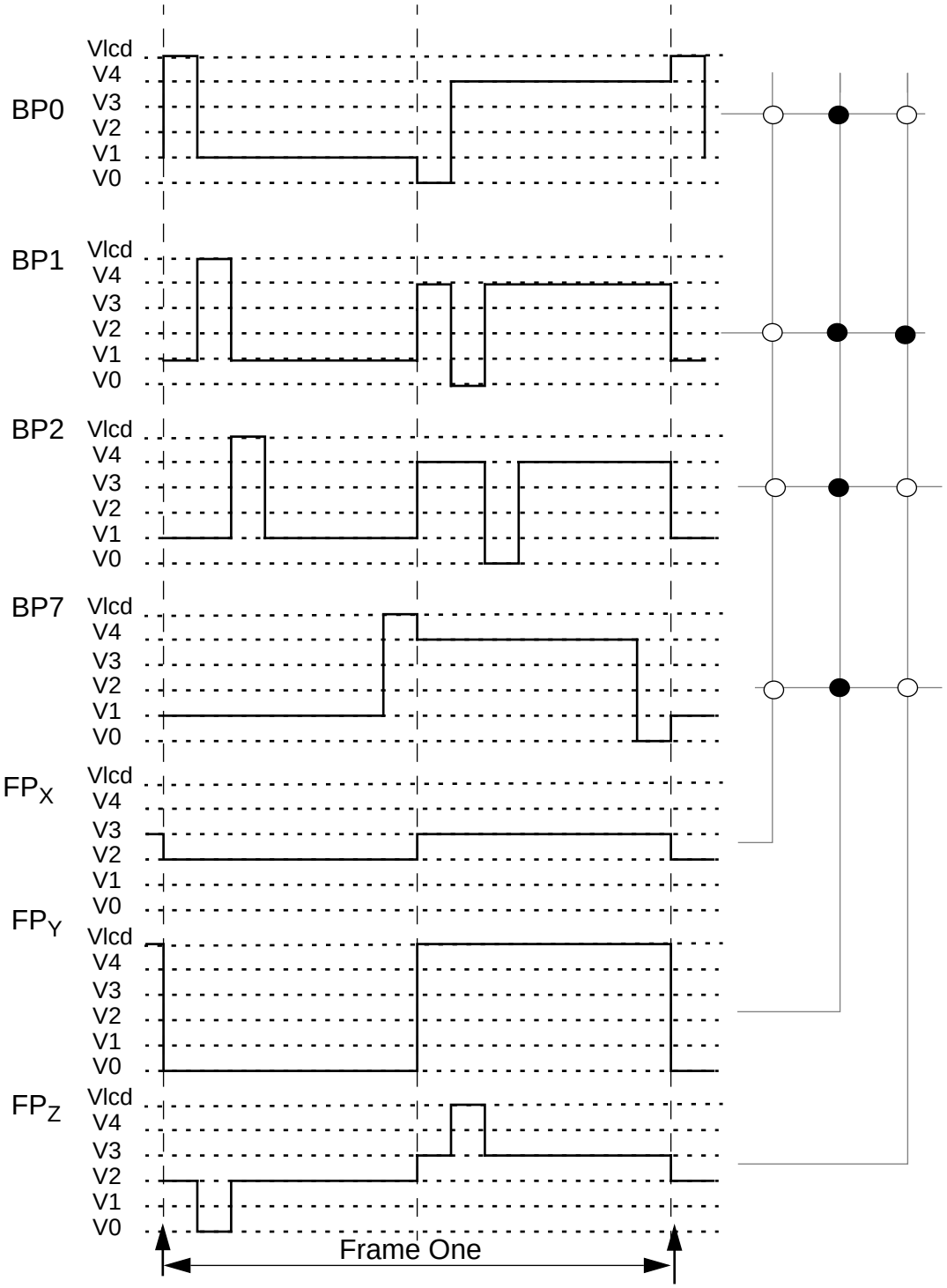


Figure 12-2. LCD 5:1 bias waveforms

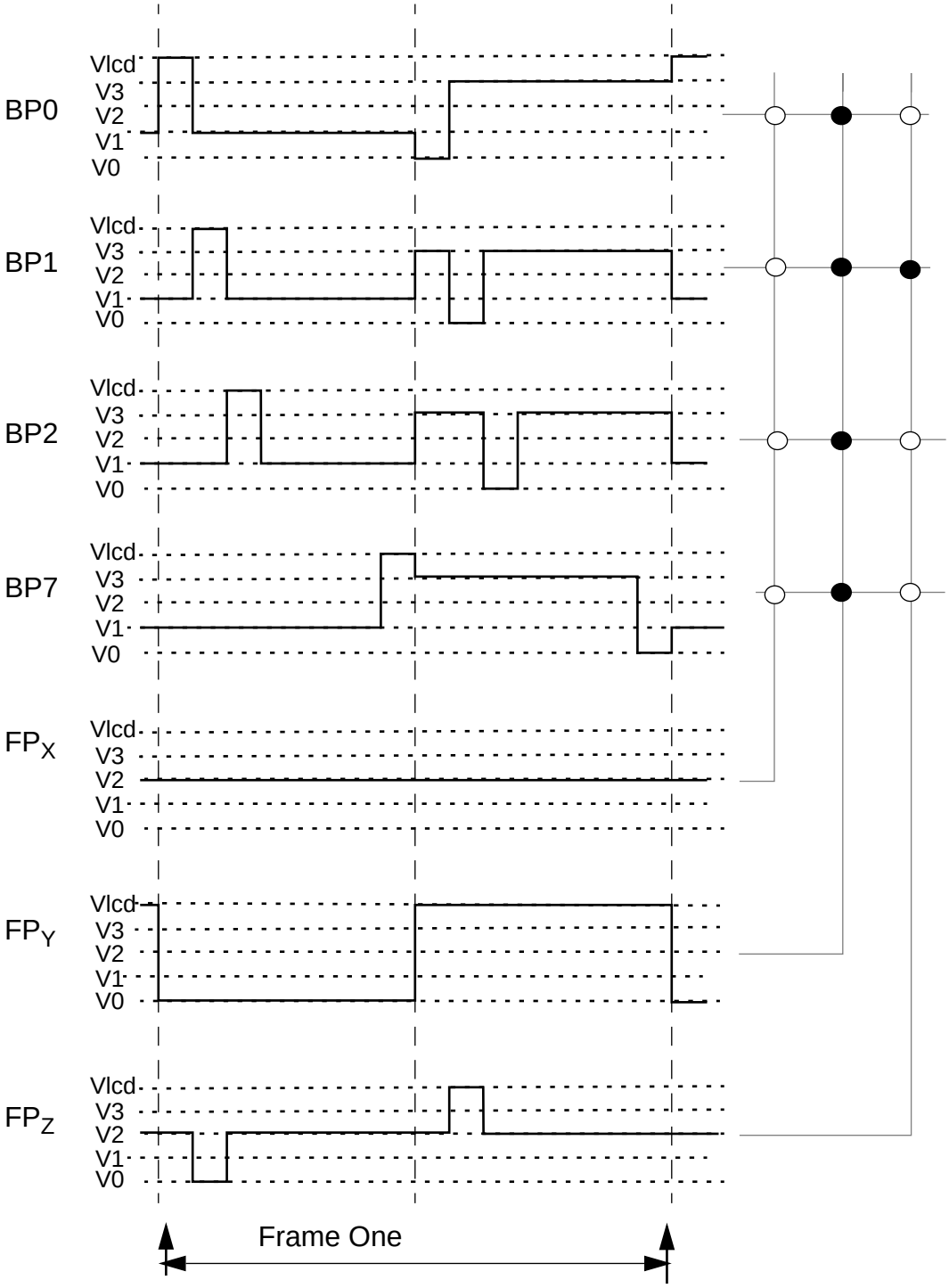


Figure 12-3. LCD 4:1 bias waveforms

12.3 LCD VOLTAGE GENERATION

Figure 12-4 shows the resistive divider chain network that is used to produce the various LCD waveforms outlined in the previous section. The LCD system can be disabled by setting the DISON bit to 0. The voltage levels of the LCD driver waveforms and hence the contrast of the LCD can be altered by selecting appropriate resistors by setting the corresponding values to the CC0 to CC3 bits in the LCDCTR register.

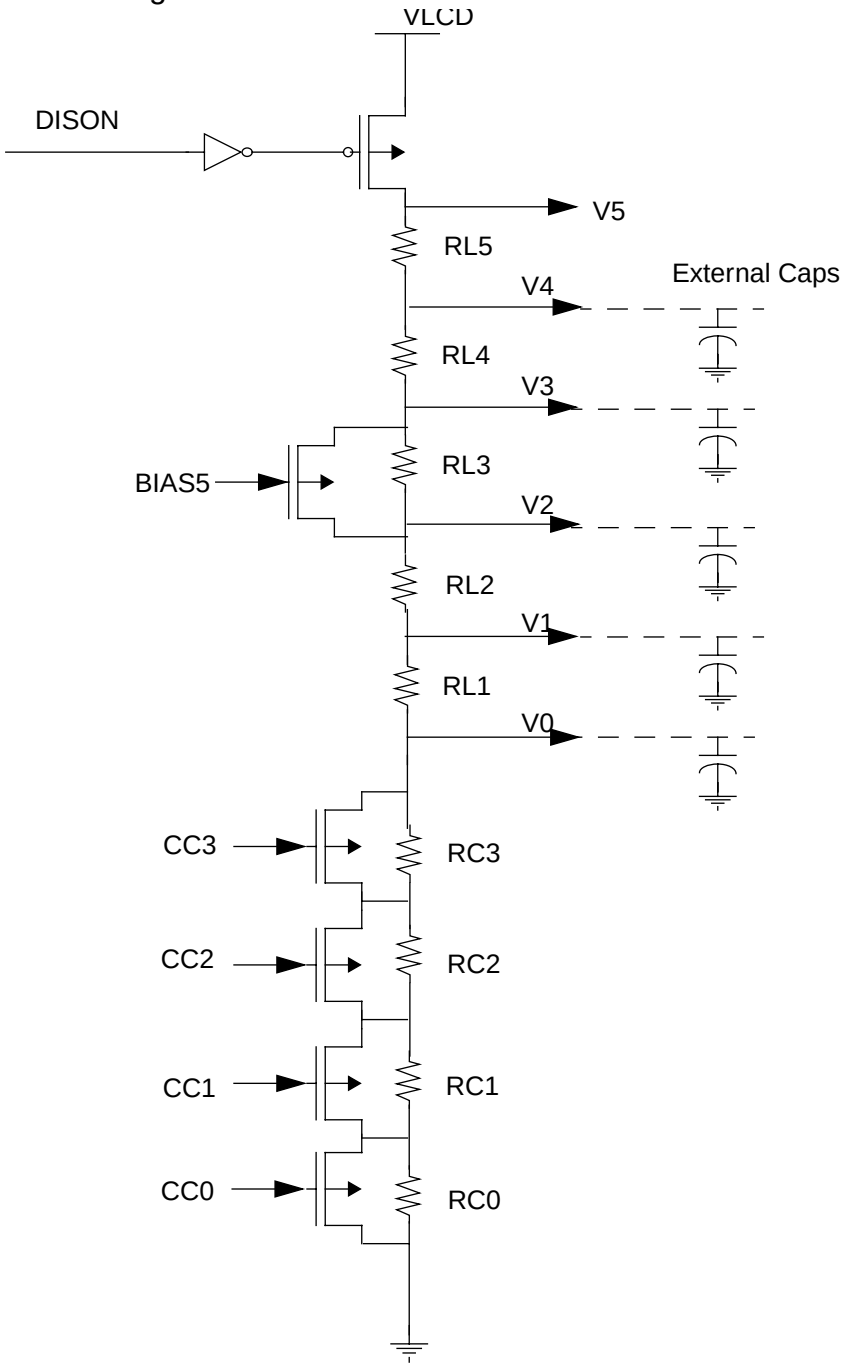


Figure 12-4. Voltage Generator

Table 12-3. LCD Voltage Characteristic

VLCD=5.0volts	4:1 bias Voltage (volts) Typical Values	5:1 bias Voltage (volts)] Typical Values
V5	5.0	5.0
V4	3.75	4.0
V3	2.5	3.0
V2	2.5	2.0
V1	1.25	1.0
V0	0.0	0.0

12.4 LCD CONTROL REGISTER (LCDCR)

		7	6	5	4	3	2	1	0
LCDCR $0007	READ / WRITE	CC3	CC2	CC1	CC0	MX8	BIAS5		DISON
	RESET	1	1	1	1	0	0	-	0

U = UNAFFECTED

DISON - BIT 0

1 = The Display is on when this bit is ’1’ and

0 = off when this bit is ’0’. Setting this bit to ’0’ also disconnects the voltage generator resistor chain from VSS, thus reducing power.

BIAS5 - Bits 2

1 = Select 5:1 bias voltage levels.

0 = Select 4:1 bias voltage levels.

MX8 - bit 3

1 = the system operates with 53 frontplanes and 8 backplanes,

0 = the system operates with 45 frontplanes and 16 backplanes.

CC0 - CC3 - bits 4,5,6,7

These bits can be used to select the values of the contrast control resistors.

1 = the corresponding resistor will be shorted. On reset these bits are set to 1.

12.5 ELECTRICAL REQUIREMENTS

Table 12-4. Ladder Voltage Values

VLCD=5.0 volts	4:1 Bias
V5-V4	1.25 ± 4%
V4-V3	1.25 ± 4%
V3-V2	0 ± 4%
V2-V1	1.25 ± 4%
V1-V0	1.25 ± 4%

Table 12-5. Contrast Resistor Values

Resistor	Resistor Values (ohm) +/- 25%
RC0	10k
RC1	20k
RC2	40k
RC3	80k
R1	100k
R2	100k
R3	100k
R4	100k
R5	100k

The contrast network will be monotonic under the normal operating conditions.

Note: Both ladder resistors and the contrast resistors are implemented using poly resistor to minimize process and temperature variation.

12.5.1 DC Requirements

The DC voltage between Front-plane to Back-plane will be less than 50mV, average over one frame. For V_{LCD} = 3 to 5 volts and temperature range of –10°C to 70°C.

SECTION 13
PHASE-LOCKED LOOP

System clock can be obtained either from the external 32kHz oscillator or from the PLL. During power on or external reset, the system is defaulted to use the 32 kHz clock from the external oscillator. This is to prevent the system from using otherwise, an unstable clock from the PLL during reset. To use the PLL clock after power on or external reset, the PON-bit should be set '1' first to switch the PLL on, a minimum of 10msec should then be allowed for the PLL clock to stabilize before setting the PCLK-bit to '1' to switch to use the PLL clock. Power on or external reset clears the PCLK-bit.

Four clocks at different frequencies are available to the system from the PLL using the frequency select bits FREQ0 and FREQ1 of the PCSR, see **Table 13-1**.

The PLL can be powered down to save power by setting the PON-bit to '0'. When the PLL is powered up after it has been powered down, again the system should allow a minimum of 10msec before switching onto the PLL clock. The PLL when it is powered up, will also provide the 1.8MHz clock for the Caller ID module. Before switching the PLL off, the PCLK-bit must be cleared thus selecting the 32kHz crystal clock as the system clock. A minimum of 2*OSC cycles should be allowed for the system to switch from the PLL clock to the oscillator clock.

Table 13-1. System Clock Frequency Selection

FREQ1	FREQ0	System Clock
0	0	3.6MHz
0	1	1.8MHz
1	0	900kHz
1	1	450kHz

13.1 PLL CONTROL AND STATUS REGISTER (PCSR $000D)

		7	6	5	4	3	2	1	0
PCSR $000D	READ	PON	PCLK				FREQ1	FREQ0	
	WRITE								
	RESET	0	0	X	X	X	0	0	X

PON - PLL power on/off bit.
- 1 = PLL is switched on.
- 0 = PLL is switched off.

PCLK - PLL Clock select bit.
- 1 = Use PLL clock.
- 0 = Use external clock.

FREQ1 - Frequency select bit 1.

FREQ0 - Frequency select bit 0.

13.2 PLL OPERATIONS

The PLL consists of an on chip VCO, a phase comparator and a divider. A filter is required to filter the phase comparator output to provide a DC signal to control the VCO frequency **Figure 13-1**.

13.3 PLL LOCK TIME

The PLL will lock in less than 10msec after the PON-bit is set. The PCLK should not be set before the PLL is locked.

13.4 PLL CLOCK FREQUENCY

When the PLL is in lock, the VCO output frequency is given by:

$F_{vco} = 114 * F_{osc}$ where F_{osc} = oscillator frequency

Example: F_{osc} = 32.768kHz, F_{vco} = 3.604MHz.

The VCO output frequency is further divided by a frequency divider circuit to give three other frequencies as shown in **Figure 13-1**.

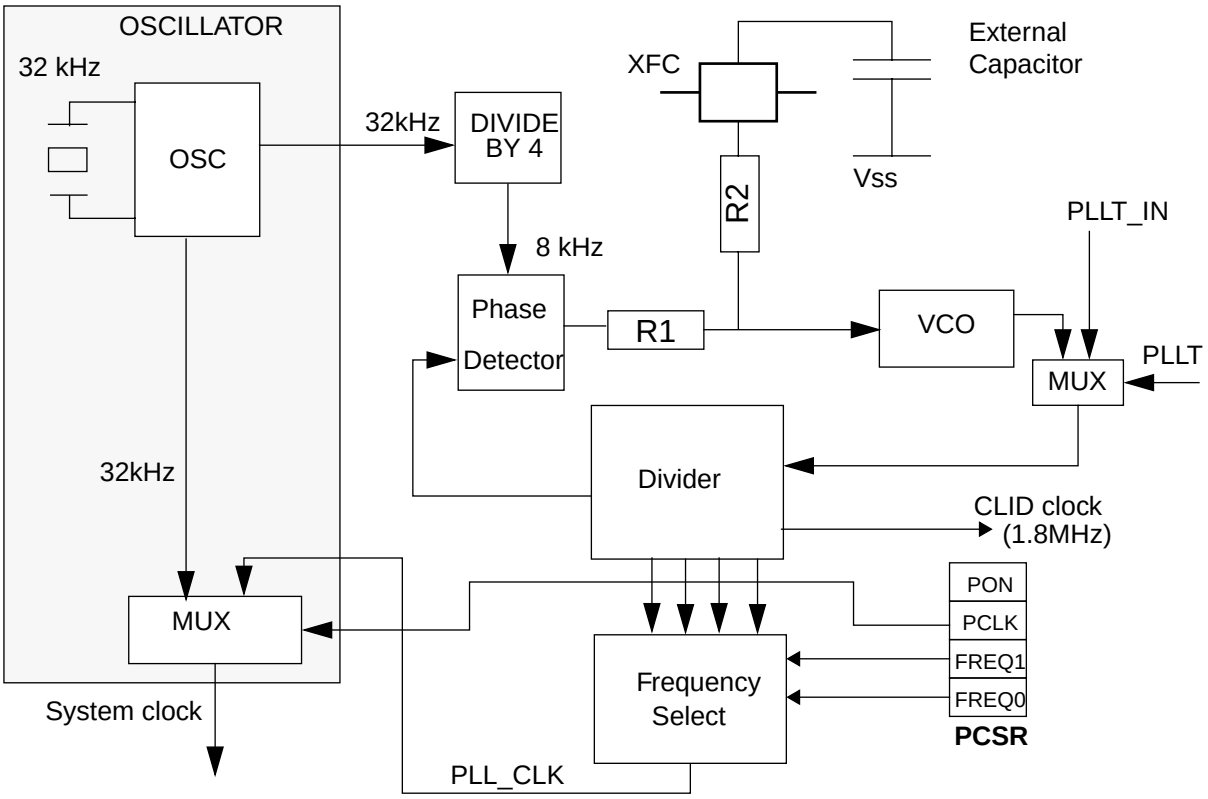


Figure 13-1. Phase Lock Loop Block Diagram

The phase comparator compares the rising edge of a 8kHz reference signal derived from the 32kHz crystal clock to the rising edge of the VCO clock after being divided by the divider. When there is a phase difference between the two signals, the phase comparator output will adjust the DC level input to the VCO to change the VCO frequency, see **Figure 13-2**.

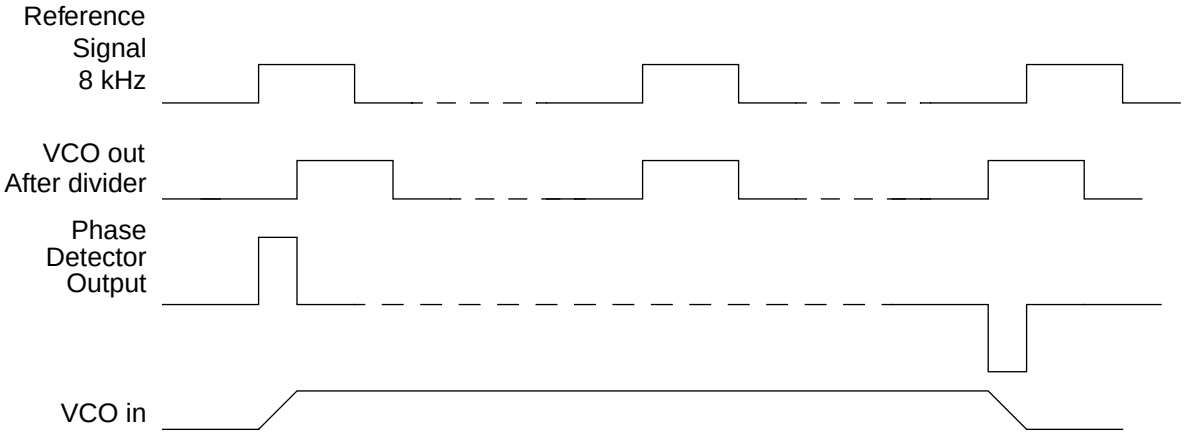


Figure 13-2. Typical Waveform for PLL

SECTION 14
INSTRUCTION SET

This section describes the addressing modes and instruction types.

14.1 ADDRESSING MODES

The CPU uses eight addressing modes for flexibility in accessing data. The addressing modes define the manner in which the CPU finds the data required to execute an instruction. The eight addressing modes are the following:

- Inherent
- Immediate
- Direct
- Extended
- Indexed, No Offset
- Indexed, 8-Bit Offset
- Indexed, 16-Bit Offset
- Relative

14.1.1 Inherent

Inherent instructions are those that have no operand, such as return from interrupt (RTI) and stop (STOP). Some of the inherent instructions act on data in the CPU registers, such as set carry flag (SEC) and increment accumulator (INCA). Inherent instructions require no memory address and are one byte long.

14.1.2 Immediate

Immediate instructions are those that contain a value to be used in an operation with the value in the accumulator or index register. Immediate instructions require no memory address and are two bytes long. The opcode is the first byte, and the immediate data value is the second byte.

14.1.3 Direct

Direct instructions can access any of the first 256 memory addresses with two bytes. The first byte is the opcode, and the second is the low byte of the operand address. In direct addressing, the CPU automatically uses $00 as the high byte of the operand address. BRSET and BRCLR are three-byte instructions that use direct addressing to access the operand and relative addressing to specify a branch destination.

14.1.4 Extended

Extended instructions use only three bytes to access any address in memory. The first byte is the opcode; the second and third bytes are the high and low bytes of the operand address.

When using the Motorola assembler, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.

14.1.5 Indexed, No Offset

Indexed instructions with no offset are one-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the conditional address of the operand. The CPU automatically uses $00 as the high byte, so these instructions can address locations $0000–$00FF.

Indexed, no offset instructions are often used to move a pointer through a table or to hold the address of a frequently used RAM or I/O location.

14.1.6 Indexed, 8-Bit Offset

Indexed, 8-bit offset instructions are two-byte instructions that can access data with variable addresses within the first 511 memory locations. The CPU adds the unsigned byte in the index register to the unsigned byte following the opcode. The sum is the conditional address of the operand. These instructions can access locations $0000–$01FE.

Indexed 8-bit offset instructions are useful for selecting the kth element in an n-element table. The table can begin anywhere within the first 256 memory locations and could extend as far as location 510 ($01FE). The k value is typically in the index register, and the address of the beginning of the table is in the byte following the opcode.

14.1.7 Indexed, 16-Bit Offset

Indexed, 16-bit offset instructions are three-byte instructions that can access data with variable addresses at any location in memory. The CPU adds the unsigned byte in the index register to the two unsigned bytes following the opcode. The sum is the conditional address of the operand. The first byte after the opcode is the high byte of the 16-bit offset; the second byte is the low byte of the offset. These instructions can address any location in memory.

Indexed, 16-bit offset instructions are useful for selecting the kth element in an n-element table anywhere in memory.

As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

14.1.8 Relative

Relative addressing is only for branch instructions. If the branch condition is true, the CPU finds the conditional branch destination by adding the signed byte following the opcode to the contents of the program counter. If the branch condition is not true, the CPU goes to the next instruction. The offset is a signed, two's complement byte that gives a branching range of –128 to +127 bytes from the address of the next location after the branch instruction.

When using the Motorola assembler, the programmer does not need to calculate the offset, because the assembler determines the proper offset and verifies that it is within the span of the branch.

14.1.9 Instruction Types

The MCU instructions fall into the following five categories:

- Register/Memory Instructions
- Read-Modify-Write Instructions
- Jump/Branch Instructions
- Bit Manipulation Instructions
- Control Instructions

14.1.10 Register/Memory Instructions

Most of these instructions use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory. **Table 14-1** lists the register/memory instructions.

Table 14-1. Register/Memory Instructions

Instruction	Mnemonic
Add Memory Byte and Carry Bit to Accumulator	ADC
Add Memory Byte to Accumulator	ADD
AND Memory Byte with Accumulator	AND
Bit Test Accumulator	BIT
Compare Accumulator	CMP
Compare Index Register with Memory Byte	CPX
EXCLUSIVE OR Accumulator with Memory Byte	EOR
Load Accumulator with Memory Byte	LDA
Load Index Register with Memory Byte	LDX
Multiply	MUL
OR Accumulator with Memory Byte	ORA
Subtract Memory Byte and Carry Bit from Accumulator	SBC
Store Accumulator in Memory	STA
Store Index Register in Memory	STX
Subtract Memory Byte from Accumulator	SUB

14.1.11 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to the memory location or to the register. The test for negative or zero instruction (TST) is an exception to the read-modify-write sequence because it does not write a replacement value. **Table 14-2** lists the read-modify-write instructions.

Table 14-2. Read-Modify-Write Instructions

Instruction	Mnemonic
Arithmetic Shift Left	ASL
Arithmetic Shift Right	ASR
Clear Bit in Memory	BCLR
Set Bit in Memory	BSET
Clear	CLR
Complement (One's Complement)	COM
Decrement	DEC
Increment	INC
Logical Shift Left	LSL
Logical Shift Right	LSR
Negate (Two's Complement)	NEG
Rotate Left through Carry Bit	ROL
Rotate Right through Carry Bit	ROR
Test for Negative or Zero	TST

14.1.12 Jump/Branch Instructions

Jump instructions allow the CPU to interrupt the normal sequence of the program counter. The unconditional jump instruction (JMP) and the jump to subroutine instruction (JSR) have no register operand. Branch instructions allow the CPU to interrupt the normal sequence of the program counter when a test condition is met. If the test condition is not met, the branch is not performed. All branch instructions use relative addressing.

Bit test and branch instructions cause a branch based on the state of any readable bit in the first 256 memory locations. These three-byte instructions use a combination of direct addressing and relative addressing. The direct address of the byte to be tested is in the byte following the opcode. The third byte is the signed offset byte. The CPU finds the conditional branch destination by adding the third byte to the program counter if the specified bit tests true. The bit to be tested

and its condition (set or clear) is part of the opcode. The span of branching is from –128 to +127 from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register. **Table 14-3** lists the jump and branch instructions.

Table 14-3. Jump and Branch Instructions

Instruction	Mnemonic
Branch if Carry Bit Clear	BCC
Branch if Carry Bit Set	BCS
Branch if Equal	BEQ
Branch if Half-Carry Bit Clear	BHCC
Branch if Half-Carry Bit Set	BHCS
Branch if Higher	BHI
Branch if Higher or Same	BHS
Branch if $\overline{\text{IRQ}}$ Pin High	BIH
Branch if $\overline{\text{IRQ}}$ Pin Low	BIL
Branch if Lower	BLO
Branch if Lower or Same	BLS
Branch if Interrupt Mask Clear	BMC
Branch if Minus	BMI
Branch if Interrupt Mask Set	BMS
Branch if Not Equal	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit Clear	BRCLR
Branch Never	BRN
Branch if Bit Set	BRSET
Branch to Subroutine	BSR
Unconditional Jump	JMP
Jump to Subroutine	JSR

14.1.13Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory. Port registers, port data direction registers, timer registers, and on-chip RAM locations are in the first 256 bytes of memory. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations. Bit manipulation instructions use direct addressing. **Table 14-4** lists these instructions.

Table 14-4. Bit Manipulation Instructions

Instruction	Mnemonic
Clear Bit	BCLR
Branch if Bit Clear	BRCLR
Branch if Bit Set	BRSET
Set Bit	BSET

14.1.14Control Instructions

These register reference instructions control CPU operation during program execution. Control instructions, listed in **Table 14-5**, use inherent addressing.

Table 14-5. Control Instructions

Instruction	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
No Operation	NOP
Reset Stack Pointer	RSP
Return from Interrupt	RTI
Return from Subroutine	RTS
Set Carry Bit	SEC
Set Interrupt Mask	SEI
Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin	STOP
Software Interrupt	SWI
Transfer Accumulator to Index Register	TAX
Transfer Index Register to Accumulator	TXA
Stop CPU Clock and Enable Interrupts	WAIT

14.1.15Instruction Set Summary

Table 14-6 is an alphabetical list of all M68HC05 instructions and shows the effect of each instruction on the condition code register.

Table 14-6. Instruction Set Summary

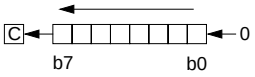
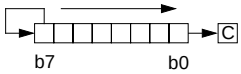
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	◊	—	◊	◊	◊	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X	Add without Carry	A ← (A) + (M)	◊	—	◊	◊	◊	IMM DIR EXT IX2 IX1 IX	AB BB CB DB EB FB	ii dd hh ll ee ff ff	2 3 4 5 4 3
AND #*opr* AND *opr* AN*D opr* AND *opr*,X AND *opr*,X AND ,X	Logical AND	A ← (A) ∧ (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	A4 B4 C4 D4 E4 F4	ii dd hh ll ee ff ff	2 3 4 5 4 3
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X	Arithmetic Shift Left (Same as LSL)		—	—	◊	◊	◊	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR ,X	Arithmetic Shift Right		—	—	◊	◊	◊	DIR INH INH IX1 IX	37 47 57 67 77	dd ff 	5 3 3 6 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BCLR *n opr*	Clear Bit n	Mn ← 0	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? Z = 1	—	—	—	—	—	REL	27	rr	3
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3

Table 14-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BIH *rel*	Branch if $\overline{\text{IRQ}}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if $\overline{\text{IRQ}}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 0	—	—	—	—	—	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X	Bit Test Accumulator with Memory Byte	(A) ∧ (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	A5 B5 C5 D5 E5 F5	ii dd hh ll ee ff ff p	2 3 4 5 4 3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? C ∨ Z = 1	—	—	—	—	—	REL	23	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? I = 0	—	—	—	—	—	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? N = 1	—	—	—	—	—	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? I = 1	—	—	—	—	—	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? Z = 0	—	—	—	—	—	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? N = 0	—	—	—	—	—	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel* ? 1 = 1	—	—	—	—	—	REL	20	rr	3
BRCLR *n opr rel*	Branch if bit n clear	PC ← (PC) + 2 + *rel* ? Mn = 0	—	—	—	—	◊	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRSET *n opr rel*	Branch if Bit n Set	PC ← (PC) + 2 + *rel* ? Mn = 1	—	—	—	—	◊	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3

Table 14-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BSET *n opr*	Set Bit n	Mn ← 1	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	—	—	—	—	—	REL	AD	rr	6
CLC	Clear Carry Bit	C ← 0	—	—	—	—	0	INH	98		2
CLI	Clear Interrupt Mask	I ← 0	—	0	—	—	—	INH	9A		2
CLR *opr* CLRA CLRX CLR *opr*,X CLR ,X	Clear Byte	M ← $00 A ← $00 X ← $00 M ← $00 M ← $00	—	—	0	1	—	DIR INH INH IX1 IX	3F 4F 5F 6F 7F	dd ff 	5 3 3 6 5
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X	Compare Accumulator with Memory Byte	(A) – (M)	—	—	◊	◊	◊	IMM DIR EXT IX2 IX1 IX	A1 B1 C1 D1 E1 F1	ii dd hh ll ee ff ff 	2 3 4 5 4 3
COM *opr* COMA COMX COM *opr*,X COM ,X	Complement Byte (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (M) X ← ($\overline{X}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	—	—	◊	◊	1	DIR INH INH IX1 IX	33 43 53 63 73	dd ff 	5 3 3 6 5
CPX #*opr* CPX *opr* CPX *opr* CPX *opr*,X CPX *opr*,X CPX ,X	Compare Index Register with Memory Byte	(X) – (M)	—	—	◊	◊	1	IMM DIR EXT IX2 IX1 IX	A3 B3 C3 D3 E3 F3	ii dd hh ll ee ff ff 	2 3 4 5 4 3
DEC *opr* DECA DECX DEC *opr*,X DEC ,X	Decrement Byte	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1	—	—	◊	◊	—	DIR INH INH IX1 IX	3A 4A 5A 6A 7A	dd ff 	5 3 3 6 5
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X	EXCLUSIVE OR Accumulator with Memory Byte	A ← (A) ⊕ (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	A8 B8 C8 D8 E8 F8	ii dd hh ll ee ff ff 	2 3 4 5 4 3

Table 14-6. Instruction Set Summary (Continued)

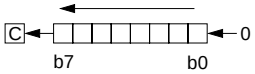
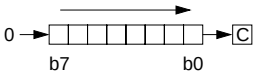
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
INC *opr* INCA INCX INC *opr*,X INC ,X	Increment Byte	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1	—	—	◊	◊	—	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff 	5 3 3 6 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Unconditional 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 ← Conditional Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff 	5 6 7 6 5
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X	Load Accumulator with Memory Byte	A ← (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff 	2 3 4 5 4 3
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X	Load Index Register with Memory Byte	X ← (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff 	2 3 4 5 4 3
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X	Logical Shift Left (Same as ASL)		—	—	◊	◊	◊	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
LSR *opr* LSRA LSRX LSR *opr*,X LSR ,X	Logical Shift Right		—	—	0	◊	◊	DIR INH INH IX1 IX	34 44 54 64 74	dd ff 	5 3 3 6 5
MUL	Unsigned Multiply	X : A ← (X) × (A)	0	—	—	—	0	INH	42		11
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	—	—	◊	◊	◊	DIR INH INH IX1 IX	30 40 50 60 70	ii ff 	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2

Table 14-6. Instruction Set Summary (Continued)

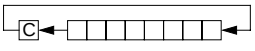
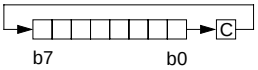
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	◊	◊	◊	DIR INH INH IX1 IX	39 49 59 69 79	dd ff	5 3 3 6 5
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	◊	◊	◊	DIR INH INH IX1 IX	36 46 56 66 76	dd ff	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	◊	◊	◊	◊	◊	INH	80		6
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)						INH			
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	◊	◊	◊	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X	Store Accumulator in Memory	M ← (A)	—	—	◊	◊	—	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff	4 5 6 5 4
STOP	Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin		—	0	—	—	—	INH	8E		2
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X	Store Index Register In Memory	M ← (X)	—	—	◊	◊	—	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff	4 5 6 5 4

Table 14-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X	Subtract Memory Byte from Accumulator	A ← (A) – (M)	—	—	◊	◊	◊	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff	2 3 4 5 4 3
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		10
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2
TST *opr* TSTA TSTX TST *opr*,X TST ,X	Test Memory Byte for Negative or Zero	(M) – $00	—	—	—	—	—	DIR INH INH IX1 IX	3D 4D 5D 6D 7D	dd ff	4 3 3 5 4
TXA	Transfer Index Register to Accumulator	A ← (X)	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	◊	—	—	—	INH	8F		2

A	Accumulator	*opr*	Operand (one or two bytes)
C	Carry/borrow flag	PC	Program counter
CCR	Condition code register	PCH	Program counter high byte
dd	Direct address of operand	PCL	Program counter low byte
dd rr	Direct address of operand and relative offset of branch instruction	REL	Relative addressing mode
DIR	Direct addressing mode	*rel*	Relative program counter offset byte
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	rr	Relative program counter offset byte
EXT	Extended addressing mode	SP	Stack pointer
ff	Offset byte in indexed, 8-bit offset addressing	X	Index register
H	Half-carry flag	Z	Zero flag
hh ll	High and low bytes of operand address in extended addressing	#	Immediate value
I	Interrupt mask	∧	Logical AND
ii	Immediate operand byte	∨	Logical OR
IMM	Immediate addressing mode	⊕	Logical EXCLUSIVE OR
INH	Inherent addressing mode	( )	Contents of
IX	Indexed, no offset addressing mode	–( )	Negation (two's complement)
IX1	Indexed, 8-bit offset addressing mode	←	Loaded with
IX2	Indexed, 16-bit offset addressing mode	?	If
M	Memory location	:	Concatenated with
N	Negative flag	↕	Set or cleared
n	Any bit	—	Not affected

Table 14-7. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	DIR	DIR	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
MSB / LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	MSB / LSB
0	BRSET0 5 3 DIR	BSET0 5 2 DIR	BRA 3 2 REL	NEG 5 2 DIR	NEGA 3 1 INH	NEGX 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0
1	BRCLR0 5 3 DIR	BCLR0 5 2 DIR	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1
2	BRSET1 5 3 DIR	BSET1 5 2 DIR	BHI 3 2 REL		MUL 11 1 INH						SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2
3	BRCLR1 5 3 DIR	BCLR1 5 2 DIR	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3
4	BRSET2 5 3 DIR	BSET2 5 2 DIR	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4
5	BRCLR2 5 3 DIR	BCLR2 5 2 DIR	BCS/BLO 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5
6	BRSET3 5 3 DIR	BSET3 5 2 DIR	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6
7	BRCLR3 5 3 DIR	BCLR3 5 2 DIR	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7
8	BRSET4 5 3 DIR	BSET4 5 2 DIR	BHCC 3 2 REL	ASL/LSL 5 2 DIR	ASLA/LSLA 3 1 INH	ASLX/LSLX 3 1 INH	ASL/LSL 6 2 IX1	ASL/LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8
9	BRCLR4 5 3 DIR	BCLR4 5 2 DIR	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9
A	BRSET5 5 3 DIR	BSET5 5 2 DIR	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A
B	BRCLR5 5 3 DIR	BCLR5 5 2 DIR	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B
C	BRSET6 5 3 DIR	BSET6 5 2 DIR	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C
D	BRCLR6 5 3 DIR	BCLR6 5 2 DIR	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D
E	BRSET7 5 3 DIR	BSET7 5 2 DIR	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E
F	BRCLR7 5 3 DIR	BCLR7 5 2 DIR	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F

INH = Inherent
IMM = Immediate
DIR = Direct
EXT = Extended
REL = Relative
IX = Indexed, No Offset
IX1 = Indexed, 8-Bit Offset
IX2 = Indexed, 16-Bit Offset

LSB of Opcode in Hexadecimal

MSB / LSB	0
0	BRSET0 5 3 DIR

MSB of Opcode in Hexadecimal
Number of Cycles
Opcode Mnemonic
Number of Bytes/Addressing Mode

SECTION 15
ELECTRICAL SPECIFICATIONS

15.1 MAXIMUM RATINGS

(Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	–0.3 to +7.0	V
LCD Supply Voltage	V_{LCD}	V_{DD} to +6.0	V
Input Voltage	V_{IN}	V_{SS} – 0.3 to V_{DD} + 0.3	V
Self-Check Mode ($\overline{IRQ}$/IRQ Pin only)	V_{IN}	V_{SS} – 0.3 to 2xV_{DD} +0.3	V
Current Drain per pin excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range	T_A	–10 to 70	°C
Storage Temperature Range	T_{STG}	–65 to 150	°C

NOTE

Maximum current drain per pin is for one pin at a time, limited by an external resistor.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{IN} and V_{OUT} be constrained to the range $V_{SS} \le (V_{IN}$ or $V_{OUT}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

15.2 THERMAL CHARACTERISTICS

Rating	Symbol	Value	Unit
Thermal resistance (Plastic)	θ_{JA}	60	°C/W

15.3 DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 3.3 V_{DC} ±10%, V_{SS} = 0 V_{DC}, T_A =–10°C to +70 °C, unless otherwise noted)

CHARACTERISTICS	Symbol	Min	Typ	Max	Unit
Output Voltage I_{LOAD} = –10µA I_{LOAD} = 10µA	 V_{OL} V_{OH}	 — V_{DD}–0.1	 — —	 0.1 —	 V V
Output High Voltage I_{LOAD} = 0.8mA PTA0-7,PTB0-7, PTC0-7	V_{OH}	V_{DD}–0.8	—	—	V
Output Low Voltage I_{LOAD} = 1.6mA PTA0-7, PTB0-7,PTC0-7	V_{OL}	—	—	0.4	V
Input High Voltage PTA0-7,PTB0-7,PTC0-7,$\overline{RESET}$, $\overline{IRQ}$, OSC1,	V_{IH}	0.8xV_{DD}	—	V_{DD}	V
Input Low Voltage PTA0-7, PTB0-7, PTC0-7,$\overline{RESET}$, $\overline{IRQ}$, OSC1, TCAP1,TCAP2	V_{IL}	V_{SS}	—	0.2xV_{DD}	V
Output Sink Current (from 5V) PTA0-7,PTB0-7, PTC0-7	I_{SINK}	—	—	3.0	mA
Output Source Current (into 2.5V) PTA0-7,PTB0-7, PTC0-7	I_{SINK}	—	3.0	—	mA
I/O Ports High-Z leakage PTA0-7,PTB0-7,PTC0-7 Measured with input at (V_{DD} - 0.1) volts and (V_{SS} + 0.1) volts.	I_{OZ}	—	—	±100	nA
Input Current $\overline{RESET}$, $\overline{IRQ}$, $\overline{IRQ0}$, $\overline{IRQ1}$, IRQ2, OSC1, TCAP1 TCAP2	I_{OZ}	—	—	±10	µA
Capacitance All input or output.	C_{IN} C_{OUT}	—	—	12 12	pF pF

Notes:

(1) All values shown reflect average measurements.

(2) Typical values at midpoint of voltage range, 25°C only.

(3) Wait I_{DD} : Only timer system active.

(4) Run (Operating) I_{DD} , Wait I_{DD} : Measured with all inputs 0.2 V from rail; no D.C. loads, less than 50pF on all outputs, C_L = 20 pF on OSC2.

(5) Wait, Stop I_{DD} : All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD}–0.2 V.

(6) Stop I_{DD} measured with OSC1 = V_{SS}.

(7) Wait I_{DD} is affected linearly by the OSC2 capacitance.

15.4 POWER DISSIPATION

(V_{DD} = 3.3Vdc, V_{SS} = 0 Vdc, T_A=–10°C to +70 °C, unless otherwise noted)

Mode	Conditions	Symbol	Max (HC705CL48)	Max (HC05CL48)	Unit
RUNNING (a)	System Clock = 3.6MHz PLL = active Core Timer = active WTimer = active LCD = active ATD = active SPI = active CLID = active	I_{DD}	6.34	6.5	mA
RUNNING (b)	System Clock = 3.6MHz PLL = active Core Timer = active WTimer = active LCD = active	I_{DD}	3.82	2.8	mA
RUNNING (c)	System Clock = 3.6MHz PLL = active Core Timer = active WTimer = active ATD = active LCD = active	I_{DD}	4.13	3.0	mA
WAIT (d)	System Clock = 3.6MHz PLL = active Core Timer = active WTimer = active ATD = active LCD = active	I_{DD}	2.67	1.0	mA
RUNNING (e)	System Clock = 3.6MHz PLL = active Core Timer = active WTimer = active ATD = active CLID = active LCD = active	I_{DD}	6.36	6.5	mA
WAIT (f)	System Clock = 3.6MHz PLL = active Core Timer = active WTimer = active ATD = active CLID = active LCD = active	I_{DD}	4.90	2.5	mA
STOP (g)	System Clock = 32kHz LCD = active WTimer = active	I_{DD}	80	80	µA
STOP (h)	No Oscillator All modules switched off.	I_{DD}	7	7	µA

Mode	Conditions	Symbol	Max (HC705CL48)	Max (HC05CL48)	Unit
Data Retention (i)	V_{DD}=2.0, RESETB = "0" No Oscillator	I_{DD}	5	5	µA
Start Up (j)	V_{DD}= 0 to 3V, RESETB = "0"	I_{DD}	3.0	3.0	mA

Note: The above I_{DD} measurements include the 32kHz oscillator I_{DD} configured as in **Figure 1-4** (a) except for case (h) and (i).

15.5 CONTROL TIMING

(V_{DD} = 3.3 V_{DC} ±10%, V_{SS} = 0 V_{DC}, T_A = –10°C to +70°C, unless otherwise noted)

CHARACTERISTIC	Symbol	Min	Max	Unit
Frequency of operation Crystal option External clock	 f_{OSC} f_{IN}	 32.768 —	 32.768 —	 kHz MHz
Internal operating frequency Crystal option External clock	 f_{OP} f_{OP}	 0.016384 —	 1.802 —	 MHz MHz
Cycle time	t_{CYC}	61035	555	ns
Crystal Oscillator Start-up time. (Refer to **Figure 1-4** (a)) "From supply=Vdd to the end of POR cycles."	t_{OXOV}	—	3	s
Crystal Oscillator Start-up Voltage	t_{OXOV}	2.7	5.0	Volts
Crystal Oscillator Small Signal Voltage Gain	A_V	10	25	
Stop recovery Start-up time (crystal oscillator + POR cycles)	t_{ILCH}	—	70	ms
PLL lock time (after asserting PON-bit)	t_{PLL}	—	10	ms
$\overline{\text{RESET}}$ pulse width	t_{RL}	1.5	I_{OUT}	t_{CYC}
Interrupt pulse width (Edge triggered)	t_{ILIH}	119	I_{IN}	ns
Interrupt pulse period	t_{ILIL}	See note1	—	t_{CYC}
OSC1 Pulse Width	t	270	280	ns

Notes:

(1) The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{CYC}.

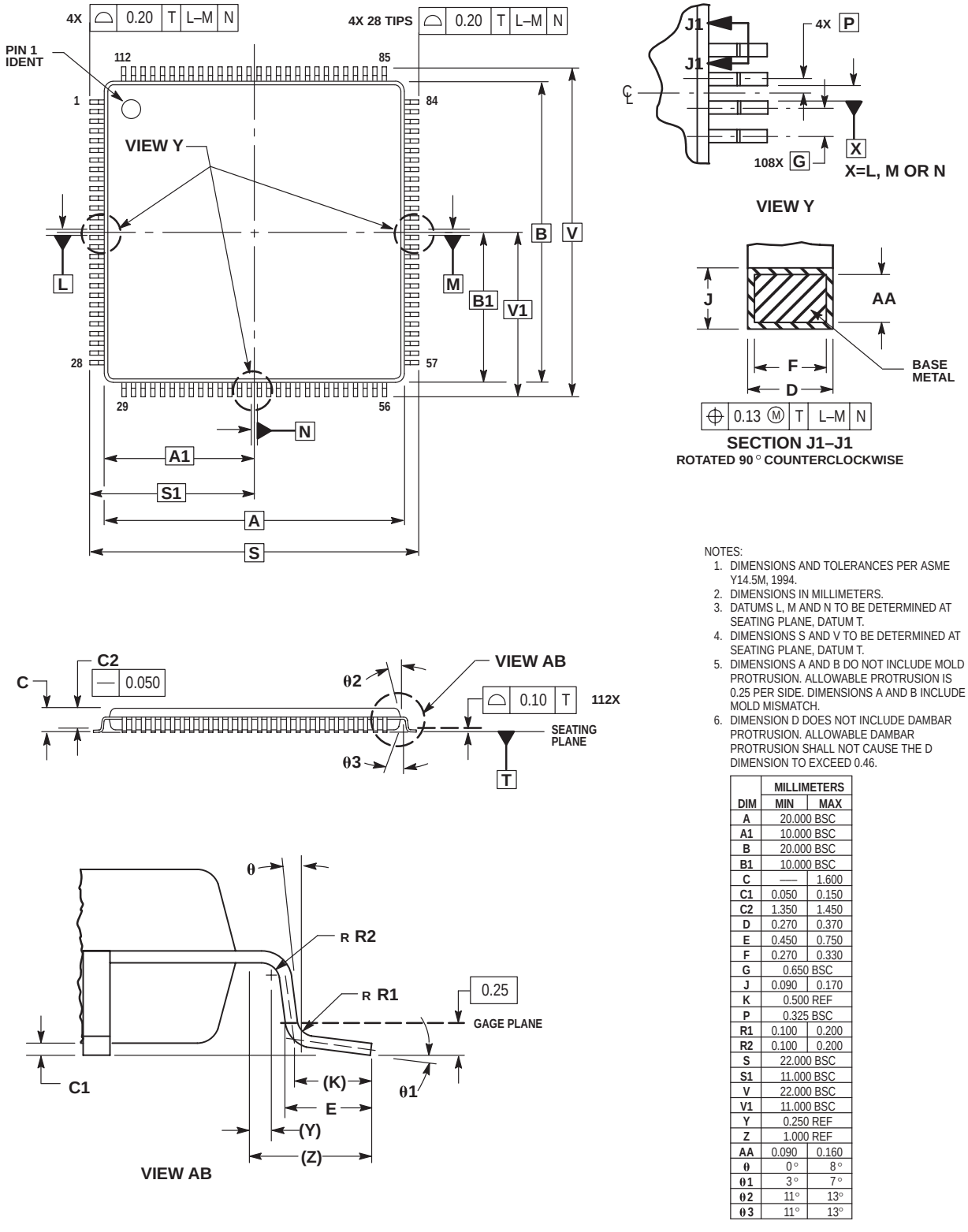
15.6 ESD PROTECTION

All pads are designed to withstand 2kV of ESD.

SECTION 16
MECHANICAL SPECIFICATIONS

This section outlines the mechanical dimensions of the 112-pin TQFP and 100-pin TQFP packages.

16.1 112-PIN THIN-QUAD-FLAT-PACKAGE (CASE 987-01)

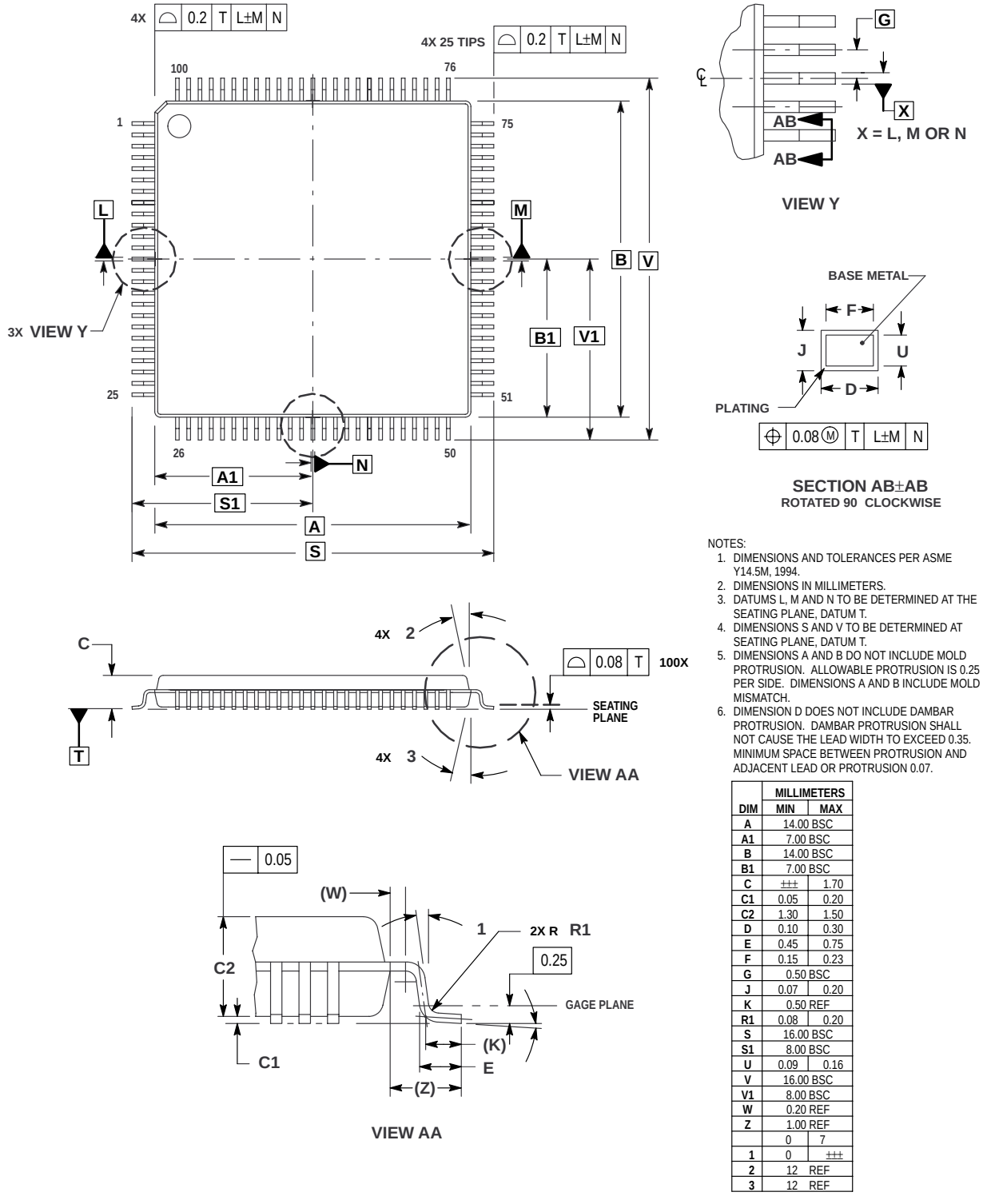


NOTES:
1. DIMENSIONS AND TOLERANCES PER ASME Y14.5M, 1994.
2. DIMENSIONS IN MILLIMETERS.
3. DATUMS L, M AND N TO BE DETERMINED AT SEATING PLANE, DATUM T.
4. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE, DATUM T.
5. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 PER SIDE. DIMENSIONS A AND B INCLUDE MOLD MISMATCH.
6. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE D DIMENSION TO EXCEED 0.46.

	MILLIMETERS	
DIM	MIN	MAX
A	20.000 BSC	
A1	10.000 BSC	
B	20.000 BSC	
B1	10.000 BSC	
C	—	1.600
C1	0.050	0.150
C2	1.350	1.450
D	0.270	0.370
E	0.450	0.750
F	0.270	0.330
G	0.650 BSC	
J	0.090	0.170
K	0.500 REF	
P	0.325 BSC	
R1	0.100	0.200
R2	0.100	0.200
S	22.000 BSC	
S1	11.000 BSC	
V	22.000 BSC	
V1	11.000 BSC	
Y	0.250 REF	
Z	1.000 REF	
AA	0.090	0.160
θ	0°	8°
θ1	3°	7°
θ2	11°	13°
θ3	11°	13°

Figure 16-1. 112-Pin TQFP Mechanical Dimensions

16.2 100-PIN THIN-QUAD-FLAT-PACKAGE (CASE 983-02)



NOTES:
1. DIMENSIONS AND TOLERANCES PER ASME Y14.5M, 1994.
2. DIMENSIONS IN MILLIMETERS.
3. DATUMS L, M AND N TO BE DETERMINED AT THE SEATING PLANE, DATUM T.
4. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE, DATUM T.
5. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 PER SIDE. DIMENSIONS A AND B INCLUDE MOLD MISMATCH.
6. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED 0.35. MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD OR PROTRUSION 0.07.

	MILLIMETERS	
DIM	MIN	MAX
A	14.00 BSC	
A1	7.00 BSC	
B	14.00 BSC	
B1	7.00 BSC	
C	---	1.70
C1	0.05	0.20
C2	1.30	1.50
D	0.10	0.30
E	0.45	0.75
F	0.15	0.23
G	0.50 BSC	
J	0.07	0.20
K	0.50 REF	
R1	0.08	0.20
S	16.00 BSC	
S1	8.00 BSC	
U	0.09	0.16
V	16.00 BSC	
V1	8.00 BSC	
W	0.20 REF	
Z	1.00 REF	
	0	7
1	0	---
2	12 REF	
3	12 REF	

Figure 16-2. 100-Pin TQFP Mechanical Dimensions

APPENDIX A
MC68HC705CL48

This appendix describes the differences between the MC68HC705CL48 and MC68HC05CL48.

This part is for user system evaluation and debugging only, and is not recommended for mass production.

A.1 INTRODUCTION

The MC68HC705CL48 is an EPROM version of the MC68HC05CL48, and is available for user system evaluation and debugging only. The MC68HC705CL48 is functionally identical to the MC68HC05CL48 with the exception of the 47.5k-byte user ROM replaced by 47.5k-byte user EPROM and, the self-check routine is replaced by a bootstrap routine.

A.2 MEMORY

The MC68HC705CL48 memory map is shown in **Figure A-1**.

A.3 BOOTLOADER MODE

Bootloader mode is entered upon the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}/V_{PP}$ pin is at V_{TST} and the PTB7 pin is at logic one. The Bootloader program and vectors are masked in the ROM area from $FE00 to $FFEF. This program handles copying of user code from an external EPROM into the on-chip EPROM. The bootload function has to be done from an external EPROM. The bootloader performs one programming pass at 1ms per byte then does a verify pass.

The user code must be a one-to-one correspondence with the internal EPROM addresses.

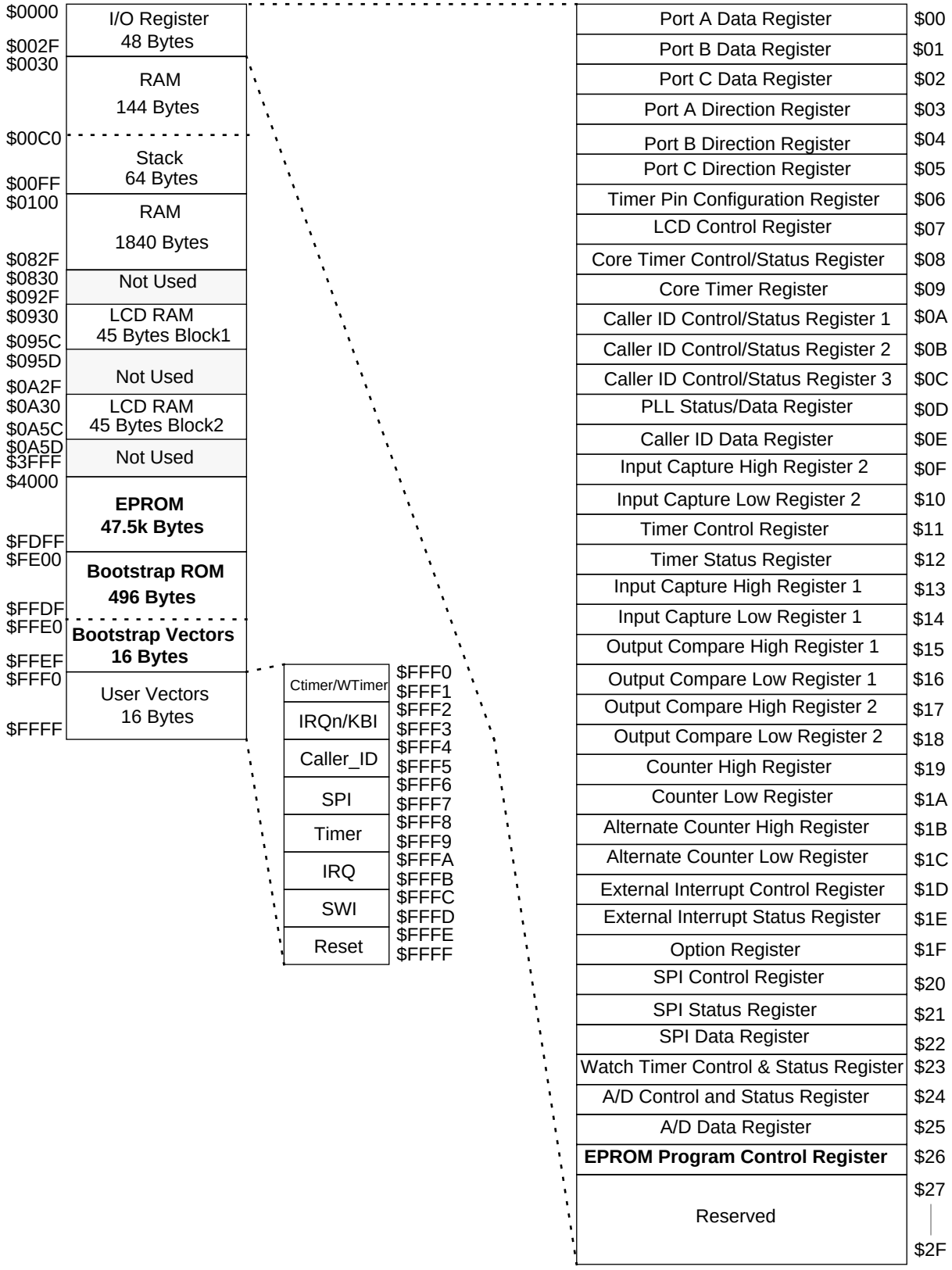


Figure A-1. MC68HC705CL48 Memory Map

A.4 EPROM PROGRAMMING

Programming the on-chip EPROM is achieved by using the Program Control Register located at address $26.

Please contact Motorola for programming board availability.

A.4.1 EPROM Program Control Register (EPCR $26)

This register is provided for programming the on-chip EPROM in the MC68HC705CL48.

		bit-7	bit-6	bit-5	bit4	bit-3	bit-2	bit1	bit-0
EPCR $0026	**Read**	RESERVED						ELAT	PGM
	Write								
	Reset	0	0	0	0	0	0	0	0

ELAT—EPROM LATch control

0 = EPROM address and data bus configured for normal reads

1 = EPROM address and data bus configured for programming (writes to EPROM cause address and data to be latched). EPROM is in programming mode and cannot be read if ELAT is 1. This bit should not be set when no programming voltage is applied to the V_{pp} pin.

PGM—EPROM ProGraM command

0 = Programming power is switched OFF from EPROM array.

1 = Programming power is switched ON to EPROM array. If ELAT ≠ 1, then PGM = 0.

A.4.2 Programming Sequence

The EPROM programming sequence is:

1. Set the ELAT bit
2. Write the data to the address to be programmed
3. Set the PGM bit
4. Delay for a time t_{PGMR}
5. Clear the PGM bit
6. Clear the ELAT bit

The last two steps must be performed with separate CPU writes.

CAUTION

It is important to remember that an external programming voltage must be applied to the V_{PP} pin while programming, but it should be equal to V_{DD} during normal operations.

Figure A-2 shows the flow required to successfully program the EPROM.

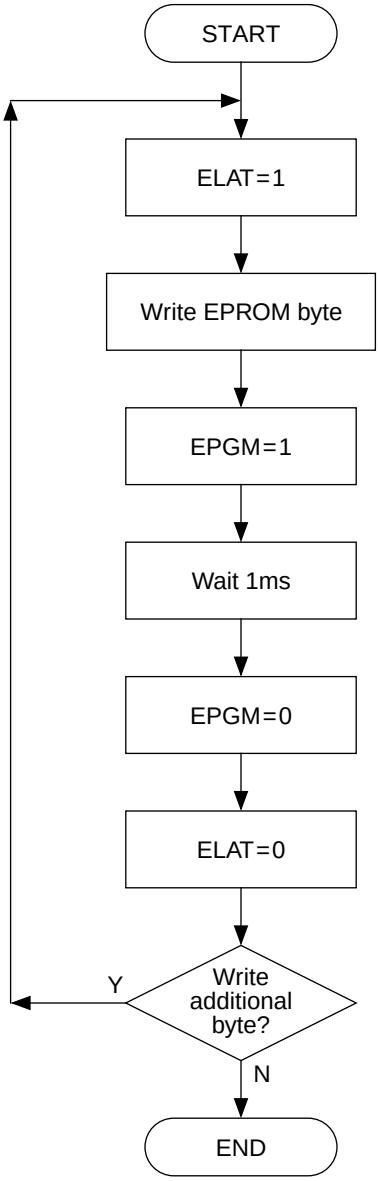


Figure A-2. EPROM Programming Sequence

APPENDIX B
MC68HC05CL16

This appendix describes the differences between the MC68HC05CL16 and MC68HC05CL48. The entire MC68HC05CL48 data sheet applies to the MC68HC05CL16, with exceptions outlined in this appendix.

B.1 INTRODUCTION

The MC68HC05CL16 is functionally identical to the MC68HC05CL48, except the following:

Table B-1. MC68HC05CL16 and MC68HC05CL48 Differences

MC68HC05CL16	MC68HC05CL48
16k-bytes user ROM	47.5k-bytes user ROM
—	LCD driver output voltages: V0, V1, V2, V3, V4
100-pin TQFP	112-pin TQFP

Figure B-1 shows a block diagram of the MC68HC05CL16.

B.2 SIGNAL DESCRIPTION

The LCD driver voltages from the resistor ladder, V0, V1, V2, V3, and V4 on the MC68HC05CL48 are not available on the MC68HC05CL16.

B.3 MEMORY

The MC68HC05CL16 has 16k-bytes of user ROM.

The MC68HC05CL16 memory map is shown in **Figure B-2**.

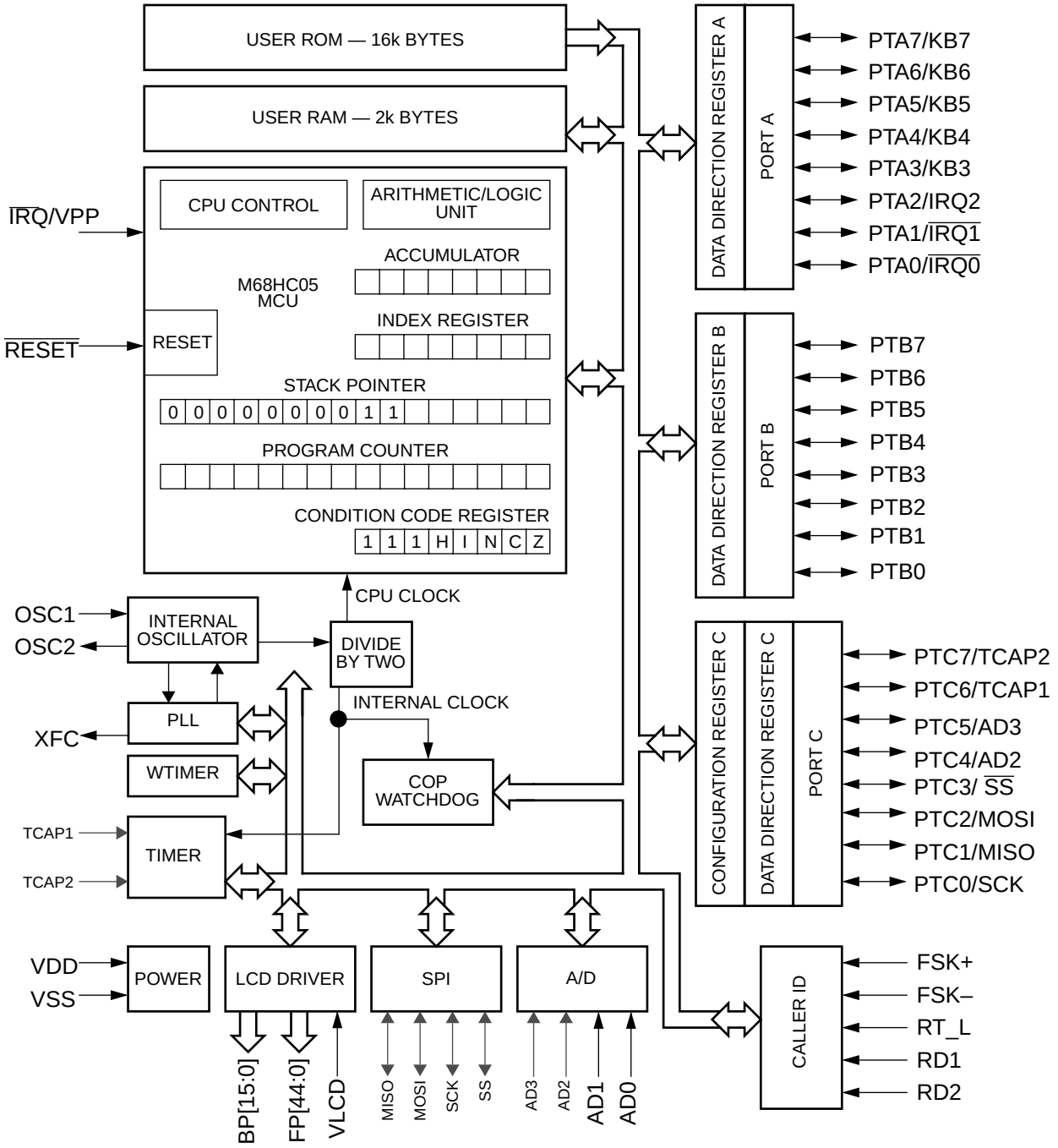


Figure B-1. MC68HC05CL16 Block Diagram

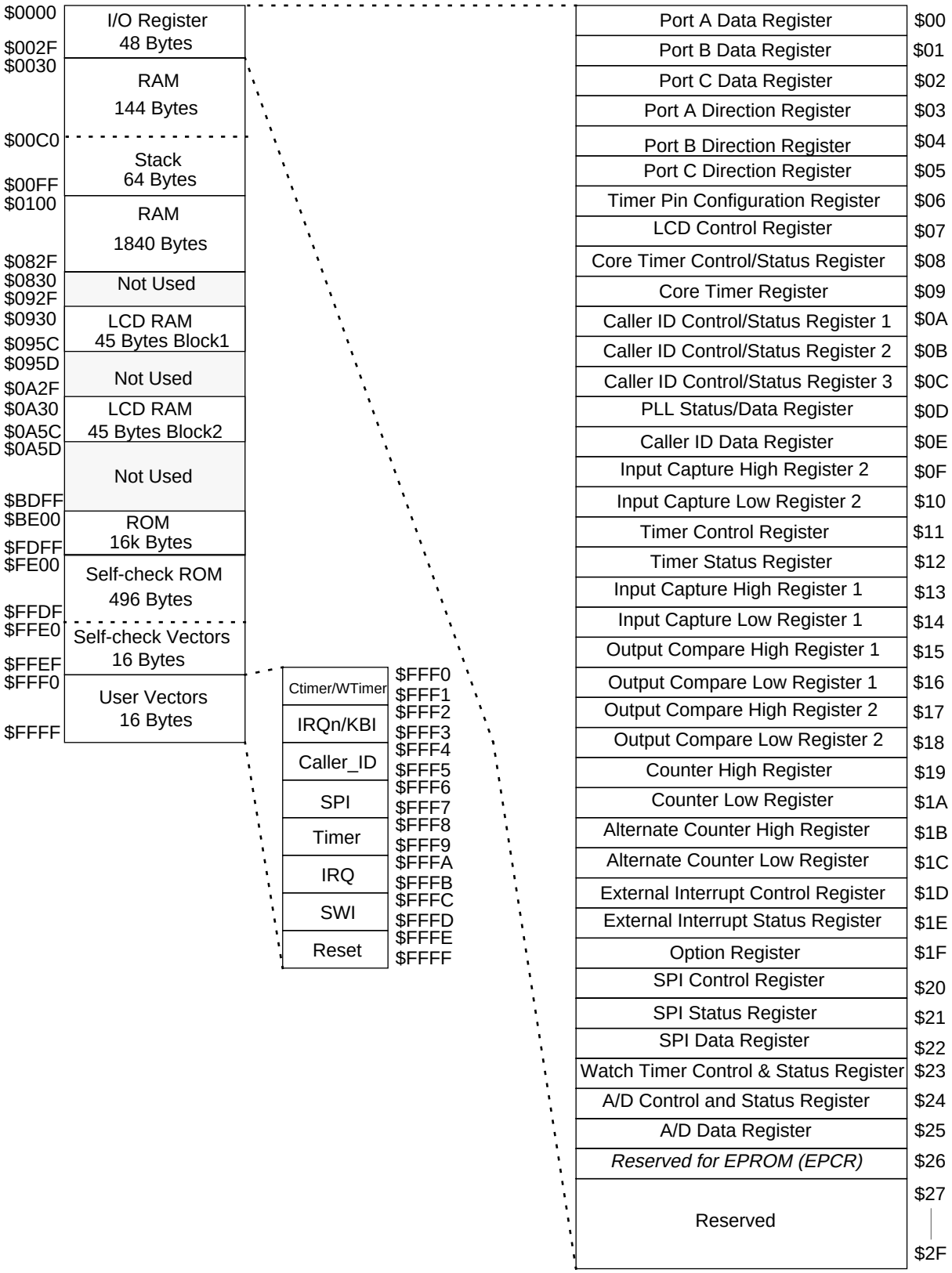
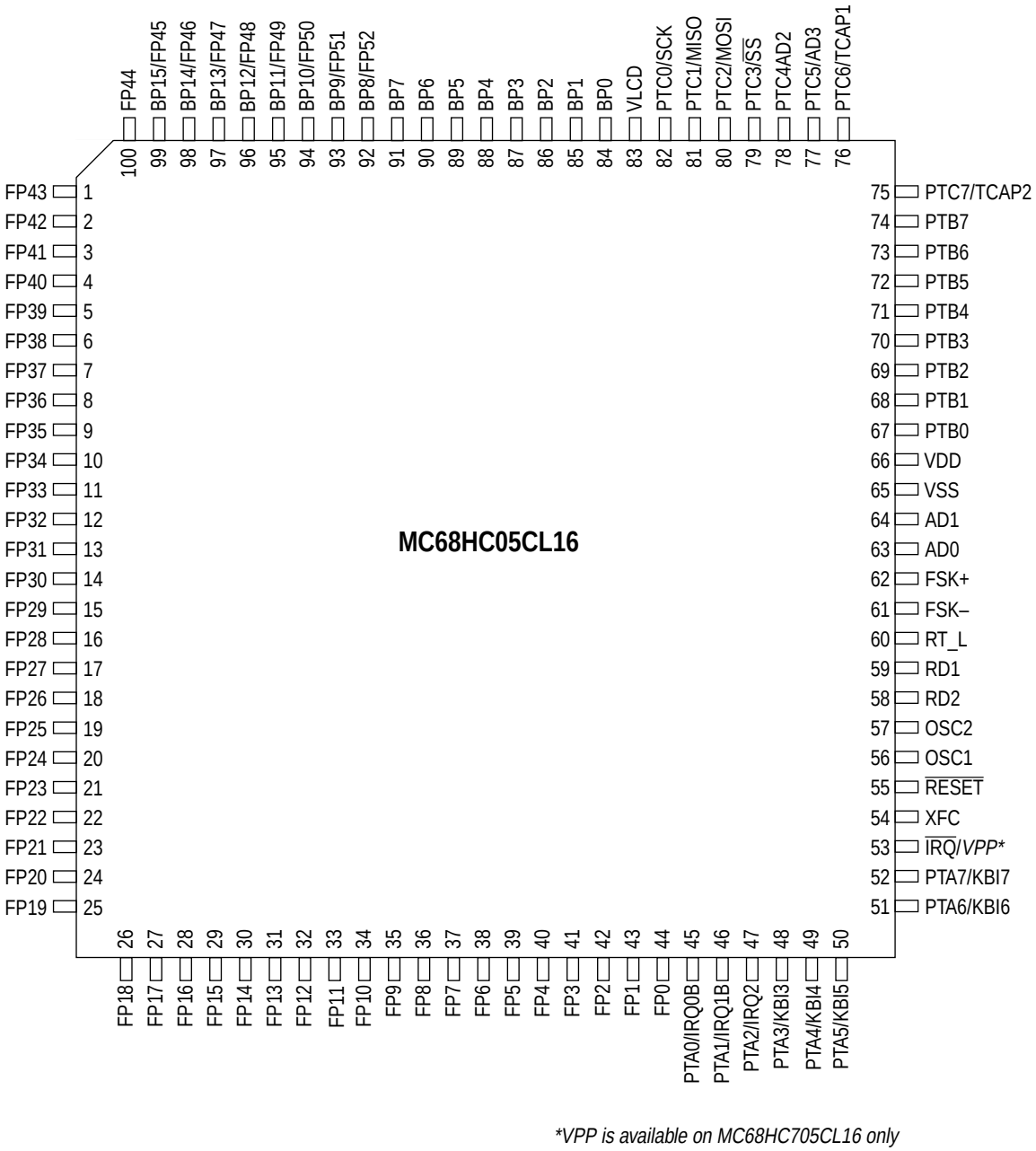


Figure B-2. MC68HC05CL16 Memory Map

B.4 PIN ASSIGNMENT

The MC68HC05CL16 pin assignment for the 100-pin TQFP package is shown in **Figure B-3**.



**VPP is available on MC68HC705CL16 only*

Figure B-3. MC68HC05CL16 Pin Assignment

APPENDIX C
MC68HC705CL16

This appendix describes the differences between the MC68HC705CL16 and MC68HC05CL16.

This part is for user system evaluation and debugging only, and is not recommended for mass production.

C.1 INTRODUCTION

The MC68HC705CL16 is an EPROM version of the MC68HC05CL16, and is available for user system evaluation and debugging only. The MC68HC705CL16 is functionally identical to the MC68HC05CL16 with the exception of the 16k-byte user ROM replaced by 16k-byte user EPROM and, the self-check routine is replaced by a bootstrap routine.

C.2 MEMORY

The MC68HC705CL16 memory map is shown in **Figure C-1**.

C.3 BOOTLOADER MODE

Bootloader mode is entered upon the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}/V_{PP}$ pin is at V_{TST} and the PTB7 pin is at logic one. The Bootloader program and vectors are masked in the ROM area from $FE00 to $FFEF. This program handles copying of user code from an external EPROM into the on-chip EPROM. The bootload function has to be done from an external EPROM. The bootloader performs one programming pass at 1ms per byte then does a verify pass.

The user code must be a one-to-one correspondence with the internal EPROM addresses.

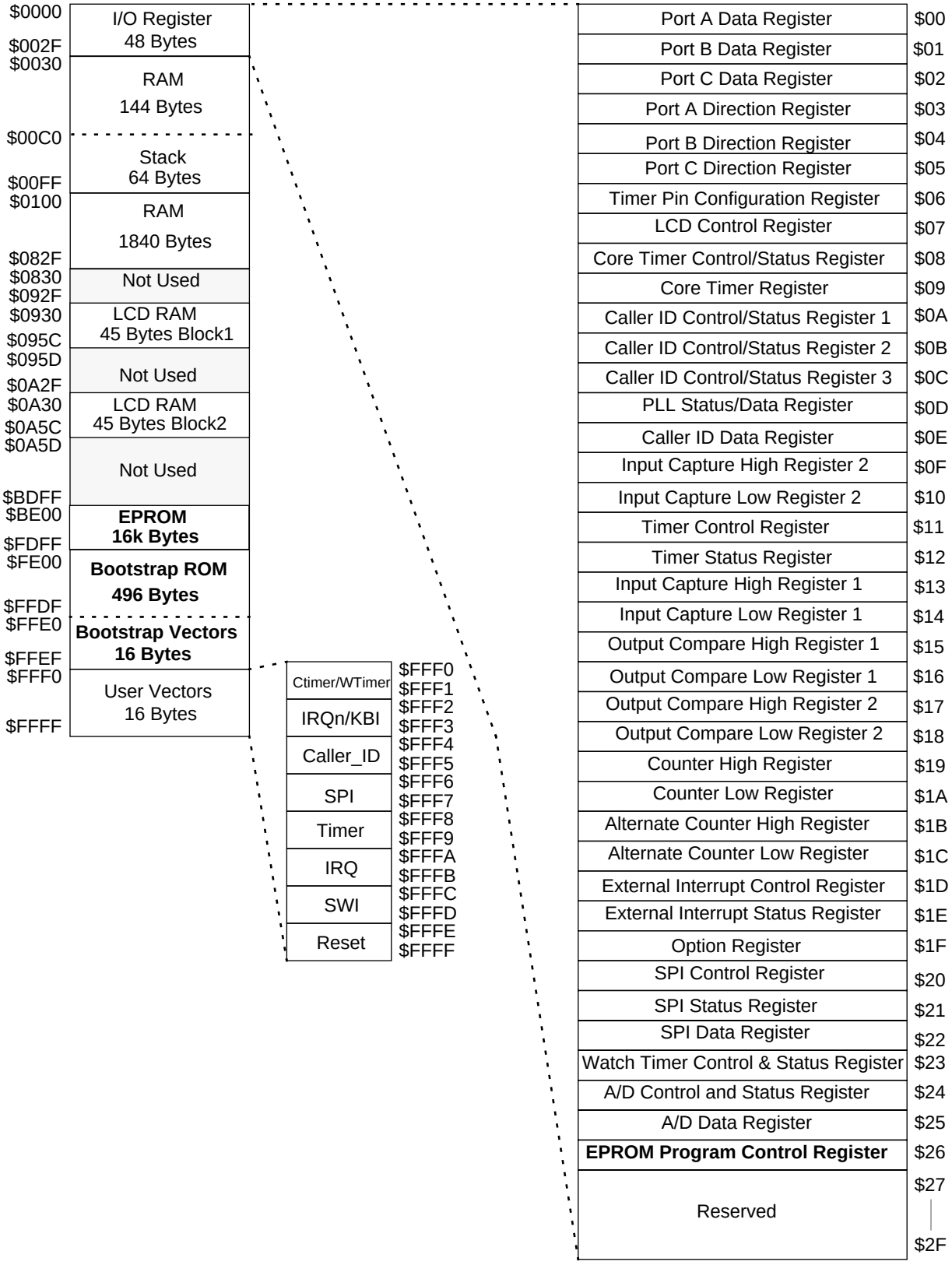


Figure C-1. MC68HC705CL16 Memory Map

C.4 EPROM PROGRAMMING

Programming the on-chip EPROM is achieved by using the Program Control Register located at address $26.

Please contact Motorola for programming board availability.

C.4.1 EPROM Program Control Register (EPCR $26)

This register is provided for programming the on-chip EPROM in the MC68HC705CL16.

		bit-7	bit-6	bit-5	bit4	bit-3	bit-2	bit1	bit-0
EPCR $0026	**Read**	RESERVED						ELAT	PGM
	Write								
	Reset	0	0	0	0	0	0	0	0

ELAT—EPROM LATch control

0 = EPROM address and data bus configured for normal reads

1 = EPROM address and data bus configured for programming (writes to EPROM cause address and data to be latched). EPROM is in programming mode and cannot be read if ELAT is 1. This bit should not be set when no programming voltage is applied to the V_{pp} pin.

PGM—EPROM ProGraM command

0 = Programming power is switched OFF from EPROM array.

1 = Programming power is switched ON to EPROM array. If ELAT ≠ 1, then PGM = 0.

C.4.2 Programming Sequence

The EPROM programming sequence is:

1. Set the ELAT bit
2. Write the data to the address to be programmed
3. Set the PGM bit
4. Delay for a time t_{PGMR}
5. Clear the PGM bit
6. Clear the ELAT bit

The last two steps must be performed with separate CPU writes.

CAUTION

It is important to remember that an external programming voltage must be applied to the V_{PP} pin while programming, but it should be equal to V_{DD} during normal operations.

Figure C-2 shows the flow required to successfully program the EPROM.

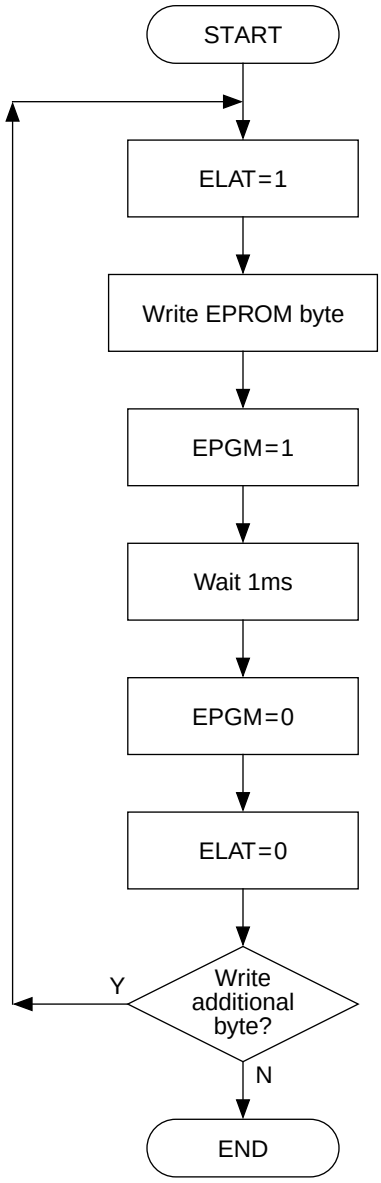


Figure C-2. EPROM Programming Sequence

How to reach us:

MFAX: RMFAX0@email.sps.mot.com – TOUCHTONE 602-244-6609

INTERNET: http://www.mot-sps.com/csic

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado 80217. 303-675-2140 or 1-800-441-2447

JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, 6F Seibu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 03-81-3521-8315

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298