68HC05CL4
68HC705CL4

SPECIFICATION
(General Release)

Singapore Design Center
CSIC Group

TABLE OF CONTENTS

Section **Page**

SECTION 1
GENERAL DESCRIPTION

SECTION 2
MEMORY

SECTION 3
CENTRAL PROCESSING UNIT

TABLE OF CONTENTS

Section **Page**

TABLE OF CONTENTS

Section **Page**

SECTION 8
TIMERS

SECTION 9
LCD DRIVER

SECTION 10
CALLER ID

SECTION 11
INSTRUCTION SET

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

LIST OF FIGURES

Figure **Title** **Page**

LIST OF TABLES

LIST OF TABLES

Table **Title** **Page**

SECTION 1
GENERAL DESCRIPTION

The MC68HC05CL4 HCMOS Microcontroller is a member of the MC68HC05 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, Caller ID subsystem and 45 x 8 LCD driver.

1.1 FEATURES

- Industry standard 8-bit M68HC05 core
- Power saving stop, wait modes
- 6K bytes of ROM
- 1K bytes of RAM (64 bytes for stack)
- Hardware Caller ID Subsystem
- 14 bidirectional I/O lines
- 16 input lines
- Keyboard interrupt lines
- 45 x 8 (or 49 x 4) LCD Driver
- 16 bit free-running timer with 1 input capture and 1 output compare
- COP Watchdog Reset
- Low Voltage Reset (LVR)
- On-chip crystal oscillator
- Internal $\overline{\text{RESET}}$ pin pull-down for LVR, COP Watchdog and Power-On Reset
- User and Self Test Modes
- Available in 80-pin QFP

1.2 MASK OPTIONS

Two mask options are available for the MC68HC05CL4.

- Low-Voltage-Reset (disabled or enabled)
- COP Watchdog Reset (disabled or enabled)

1.3 MCU STRUCTURE

The overall block diagram of the MC68HC05CL4 is shown in **Figure 1-1**.

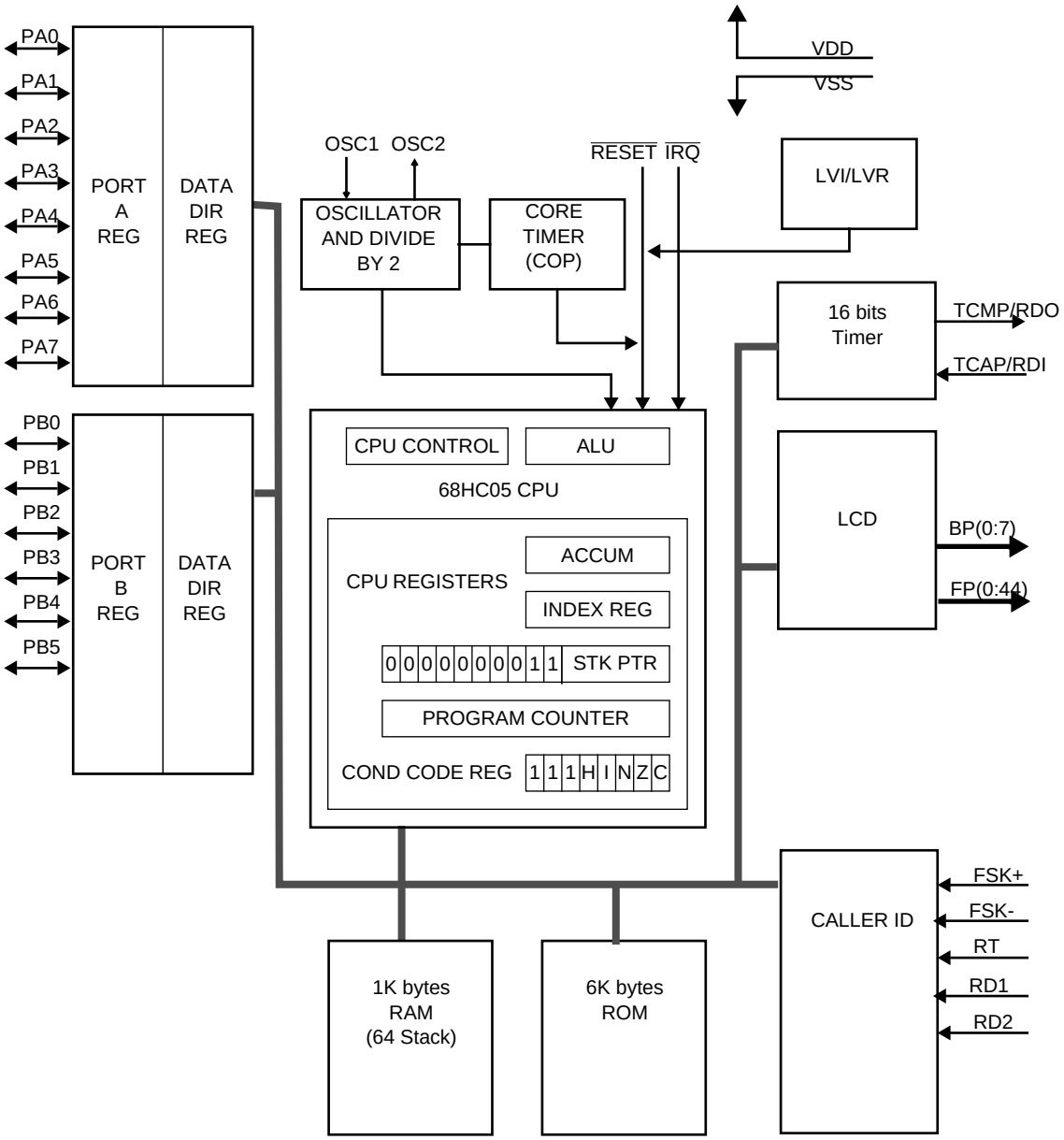


Figure 1-1. MC68HC05CL4 Block Diagram

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 12**.

1.4 PIN ASSIGNMENTS

The MC68HC05CL4 is available in the 80-pin QFP package. The pin assignments for this packages are shown in **Figure 1-2**.

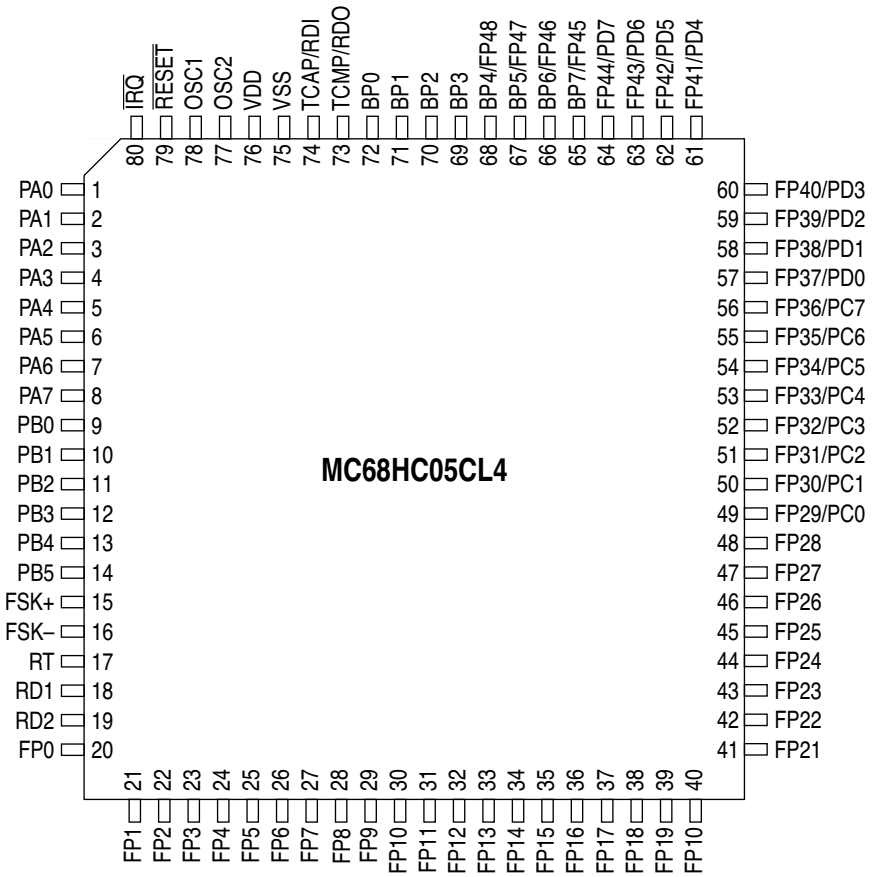


Figure 1-2. MC68HC05CL4 Pin Assignments

1.5 FUNCTIONAL PIN DESCRIPTION

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

1.5.1 VDD AND VSS

Power is supplied to the MCU through VDD and VSS. VDD is the positive supply, and VSS is ground. The MCU operates from a single power supply.

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, special care should be taken to provide good power supply bypassing at the MCU by using bypass capacitors with good high-frequency characteristics that are positioned as close to the MCU as possible.

1.5.2 OSC1, OSC2

The OSC1 and OSC2 pins are the connections for the on-chip oscillator. The OSC1, and OCS2 pins can accept the following sets of components:

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

The frequency, f_{OSC}, of the oscillator or external clock source is divided by two to produce the internal operating frequency, f_{OP}.

The circuit in **Figure 1-3(a)** shows a typical oscillator circuit for an AT-cut, parallel resonant crystal. The crystal manufacturer's recommendations should be followed, 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 capacitances. The crystal and components should be mounted as close as possible to the pins for start-up stabilization and to minimize output distortion. An external start-up resistor of approximately 2 MΩ is needed between OSC1 and OSC2 for the crystal type oscillator.

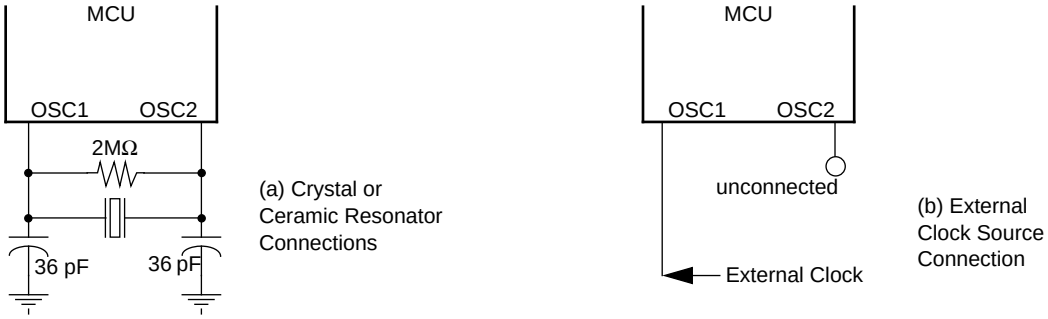


Figure 1-3. Oscillator Connections

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 1-3(b)**.

1.5.3 $\overline{\text{RESET}}$

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 RESET pin contains a steering diode to discharge any voltage on the pin to VDD, when the power is removed. The RESET pin contains an internal Schmitt trigger to improve its noise immunity as an input. The RESET pin has an internal pull-down device that pulls the RESET pin low when there is an internal COP Watchdog reset or during the power-on reset cycles. Refer to **Section 5**.

1.5.4 $\overline{IRQ}$ / V_{PP}

This pin has two choices of interrupt triggering sensitivity through the IRQ bit in the Interrupt Status and Control Register (ISCR). The choices are:

1. edge-sensitive triggering only, or
2. both edge-sensitive and level-sensitive triggering.

In addition, the IRQ pin may be selected to trigger and interrupt on either the rising or falling edge of the IRQ pin signal through the EDGE bit in the ISCR.

The MCU completes the current instruction before it responds to the interrupt request.

If the option is selected to include level-sensitive triggering, the IRQ input requires an external resistor to V_{DD} for “wire-OR” operation.

The $\overline{IRQ}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. See **Section 4**.

This pin is also used to supply the MC68HC705CL48 EPROM array with the programming voltage.

NOTE

If the voltage level applied to the $\overline{IRQ}$ pin exceeds V_{DD} it may affect the MCU’s mode of operation.

1.5.5 PA0-PA7

These eight I/O lines comprise Port A. The state of any pin is software programmable and all Port A lines are configured as inputs during power-on or reset. See **Section 7** for more details on the I/O ports. PA4 to PA7 have internal pull-up resistors and can generate key board interrupt.

1.5.6 PB0-PB5

These six I/O lines comprise Port B. The state of any pin is software programmable and all Port B lines are configured as inputs during power-on or reset. See **Section 7** for more details on the I/O ports. All the Port B pins PB0 - PB5 can sink 10mA to drive LEDs.

1.5.7 TCMP/RDO

This pin provides an output for the output compare feature (TCMP) of the on-chip programmable timer system and can be programed to as Raw Data Out (RDO) from the Caller ID module. Refer to **Section 8** for additional information.

1.5.8 TCAP/RDI

This pin is used as TCAP input to the input capture feature of the on-chip programmable timer system and can be configured as the Raw Data Input (RDI) to the MCU under software control if an external Caller ID device is used. Refer to **Section 8** for additional information.

1.5.9 FSK+, FSK–

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

1.5.10 RT

This input is an incoming ring detect power-up control.

1.5.11 RD1 and RD2

These inputs are incoming ring qualifiers.

1.5.12 BP0 - BP7

These are the backplane drivers dedicated to the LCD subsystem. When the LCD is used in multiplex by four mode BP4 - BP7 are configured as extra frontplanes FP45 - FP48.

1.5.13 FP0 - FP44

These are the front plane drivers dedicated to the LCD subsystem. FP29 to FP44 can be configured as input only ports C and D under software control.

SECTION 2
MEMORY

When the MC68HC05CL4 is in the Single-Chip Mode the 32 bytes of I/O,1K bytes of user RAM, and 6K bytes of user ROM are all active as shown in **Figure 2-1**.

2.1 I/O AND CONTROL REGISTERS

The I/O and Control Registers reside in locations $0000-$001F. The overall organization of these registers is shown in **Figure 2-1**. The bit assignments for each register are shown in **Figure 2-2**. Reading from unimplemented bits will return unknown states, and writing to unimplemented bits will be ignored.

2.2 RAM

The total RAM consists of 1K bytes (including the stack) at locations $0050 thru $044F. The stack begins at address $00FF and proceeds down to $00C0. The stack pointer can access 64 locations from $00C0 to $00FF. Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

2.3 ROM

There are a total of 6656 bytes of user ROM on chip. This includes 6144 bytes of user ROM with locations $0800 thru $1FDF for user program storage and16 bytes for user vectors at locations $1FF0 thru $1FFF. (There are 16 bytes reserved for self-check program vectors, located at $1FE0-$1FEF.)

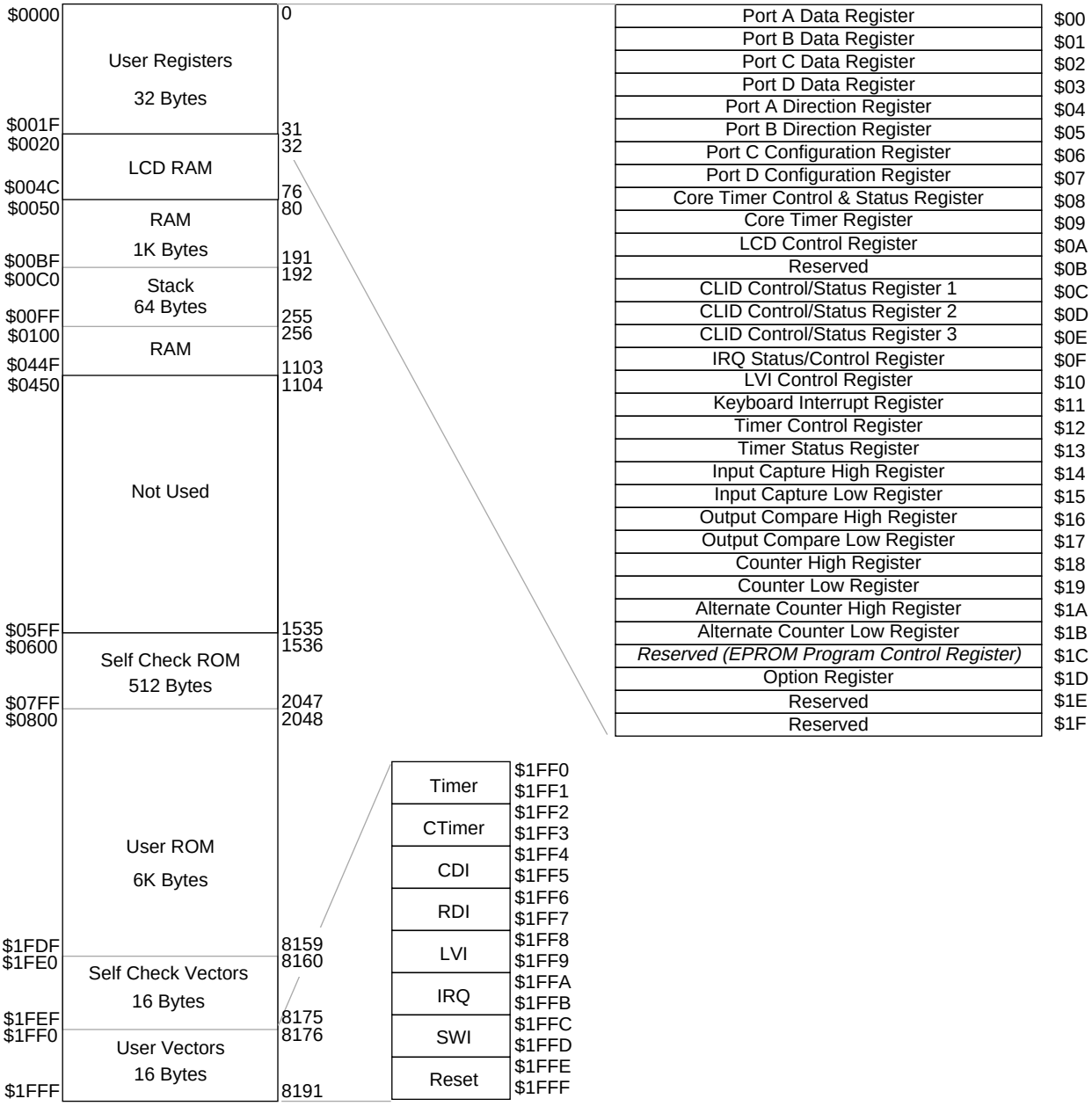


Figure 2-1. Memory Map

ADDR	REGISTER	R/W	BIT 7	BIT 6	BIT 5	BIT 4	BIT3	BIT2	BIT1	BIT 0
$0000	Port A Data PORTA	READ	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		WRITE								
$0001	Port B Data PORTB	READ	0	0	PB5	PB4	PB3	PB2	PB1	PB0
		WRITE								
$0002	Port C Data PORTC	READ	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		WRITE								
$0003	Port D Data PORTD	READ	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
		WRITE								
$0004	Port A Data Direction DDRA	READ	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		WRITE								
$0005	Port B Data Direction DDRB	READ	0	0	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		WRITE								
$0006	Port C Configuration CFGC	READ	CFGC7	CFGC6	CFGC5	CFGC4	CFGC3	CFGC2	CFGC1	CFGC0
		WRITE								
$0007	Port D Configuration CFGD	READ	CFGD7	CFGD6	CFGD5	CFGD4	CFGD3	CFGD2	CFGD1	CFGD0
		WRITE								
$0008	Core Timer Control CTCSR	READ	CTOF	RTIF	CTOFE	RTIE	0	0	RT1	RT0
		WRITE								
$0009	Core Timer Register CTR	READ	CT7	CT6	CT5	CT4	CT3	CT2	CT1	CT0
		WRITE								
$000A	LCD Control LCDCTR	READ	CC3	CC2	CC1		MX4	FC	LC	DISON
		WRITE								
$000B	RESERVED	READ	R	R	R	R	R	R	R	R
		WRITE								
$000C	CLRID Control Status1 CLCSR1	READ	RDIF	RDIE	CDIF	CDIE			RDO	CDO
		WRITE	0		0					
$000D	CLRID Control Status2 CLCSR2	READ		CDPW	RDPW			CIDSD	$\overline{RD}$	$\overline{CD}$
		WRITE								
$000E	CLRID Control Status3 CLCSR3	READ	SDSL				RDEDG	CDEDG	RDOE	CDOE
		WRITE								
$000F	IRQ Status/Control ISCR	READ	IRQM	IRQS	EDGE	0	REQ	0	0	0
		WRITE							ACK	

= UNIMPLEMENTED R = RESERVED

Figure 2-2. I/O Registers

ADDR	REGISTER	R/W	BIT 7	BIT 6	BIT 5	BIT 4	BIT3	BIT2	BIT1	BIT 0
$0010	LVI Control LVSCR	READ						LVIS	LVIF	LVIE
		WRITE							0	
$0011	Keyboard Interrupt KBIR	READ	KBIE7	KBIE6	KBIE5	KBIE4	0	KEDGE	KBIF	0
		WRITE								KBIC
$0012	Timer Control TCR	READ	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
		WRITE								
$0013	Timer Status TSR	READ	ICF	OCF	TOF	0	0	0	0	0
		WRITE								
$0014	Input Capture H ICH	READ	IC15	IC14	IC13	IC12	IC11	IC10	IC9	IC8
		WRITE								
$0015	Input Capture L ICL	READ	IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0
		WRITE								
$0016	Output Compare H OCH	READ	OC15	OC14	OC13	OC12	OC11	OC10	OC9	OC8
		WRITE								
$0017	Output Compare L OCL	READ	OC7	OC6	OC5	OC4	OC3	OC2	OC1	OC0
		WRITE								
$0018	Timer Counter H TCH	READ	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
		WRITE								
$0019	Timer Counter L TCL	READ	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
		WRITE								
$001A	Alternate Counter H ACH	READ	AC15	AC14	AC13	AC12	AC11	AC10	AC9	AC8
		WRITE								
$001B	Alternate Counter L ACL	READ	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		WRITE								
$001C	*RESERVED FOR EPROM CONTROL*	*READ*	*R*	*R*	*R*	*R*	*R*	*R*	*ELAT*	*PGM*
		WRITE								
$001D	*RESERVED FOR OPTION*	*READ*		*COP*				*LVRE*		
		WRITE								
$001E	RESERVED	READ	R	R	R	R	R	R	R	R
		WRITE								
$001f	RESERVED	READ	R	R	R	R	R	R	R	R
		WRITE								

= UNIMPLEMENTED R = RESERVED

Figure 2-2. I/O Registers

SECTION 3
CENTRAL PROCESSING UNIT

The MC68HC05CL4 has an 8K memory map. Therefore it uses only the lower 13 bits of the address bus. In the following discussion the upper 3 bits of the address bus can be ignored. 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.

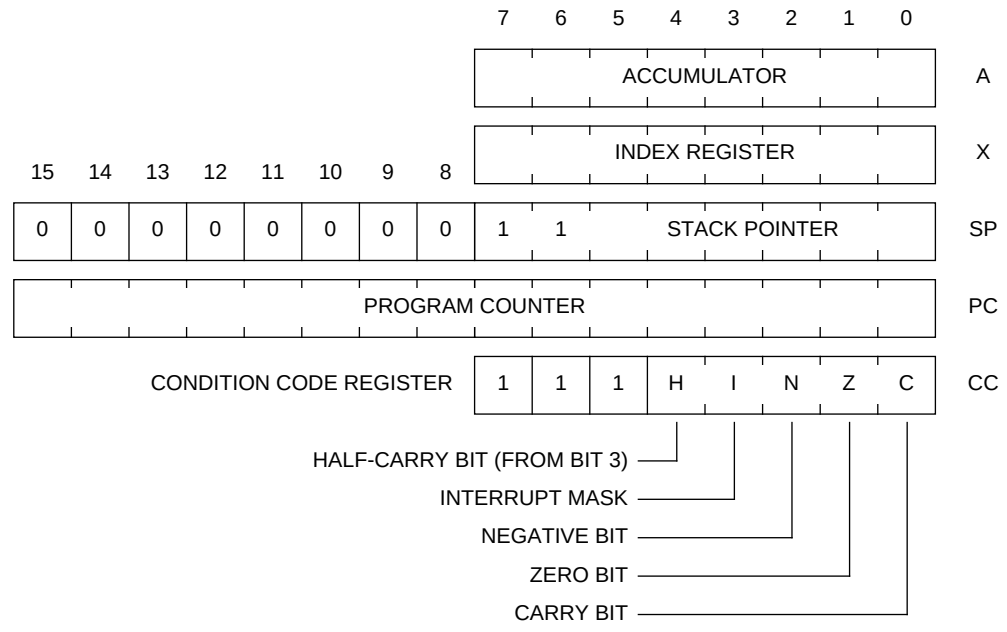


Figure 3-1. MC68HC05 Programming Model

3.2 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.

3.3 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.

3.4 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 $00C0 to $00FF. 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.

3.5 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.

3.6 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.

3.6.1 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.

3.6.2 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.

3.6.3 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.

3.6.4 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.

3.6.5 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

This section describes the interrupt structure of the MC68HC05CL4.

4.1 CPU INTERRUPT PROCESSING

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.

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 $1FF0 thru $1FFF as defined in **Table 4-1**.

Table 4-1. Vector Address for Interrupts and Reset

Register	Flag	Interrupt	CPU Int	Vector Address
N/A	N/A	Reset	RESET	$1FFE - $1FFF
N/A	N/A	Software	SWI	$1FFC - $1FFD
N/A	N/A	External Interrupt/ Key board interrupt	IRQ/KBI	$1FFA - $1FFB
N/A	N/A	Low Voltage	LVI	$1FF8 - $1FF9
CLCSR1	RDF	Ring Detect	RDI	$1FF6 - $1FF7
CLCSR1	CDF	Carrier Detect	CDI	$1FF4 - $1FF5
CTCSR	CTOF RTIF	Core Timer	CTIMER	$1FF2 - $1FF3
TSR	TOF OCF ICF	Timer	TIMER	$1FF0 - $1FF1

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.

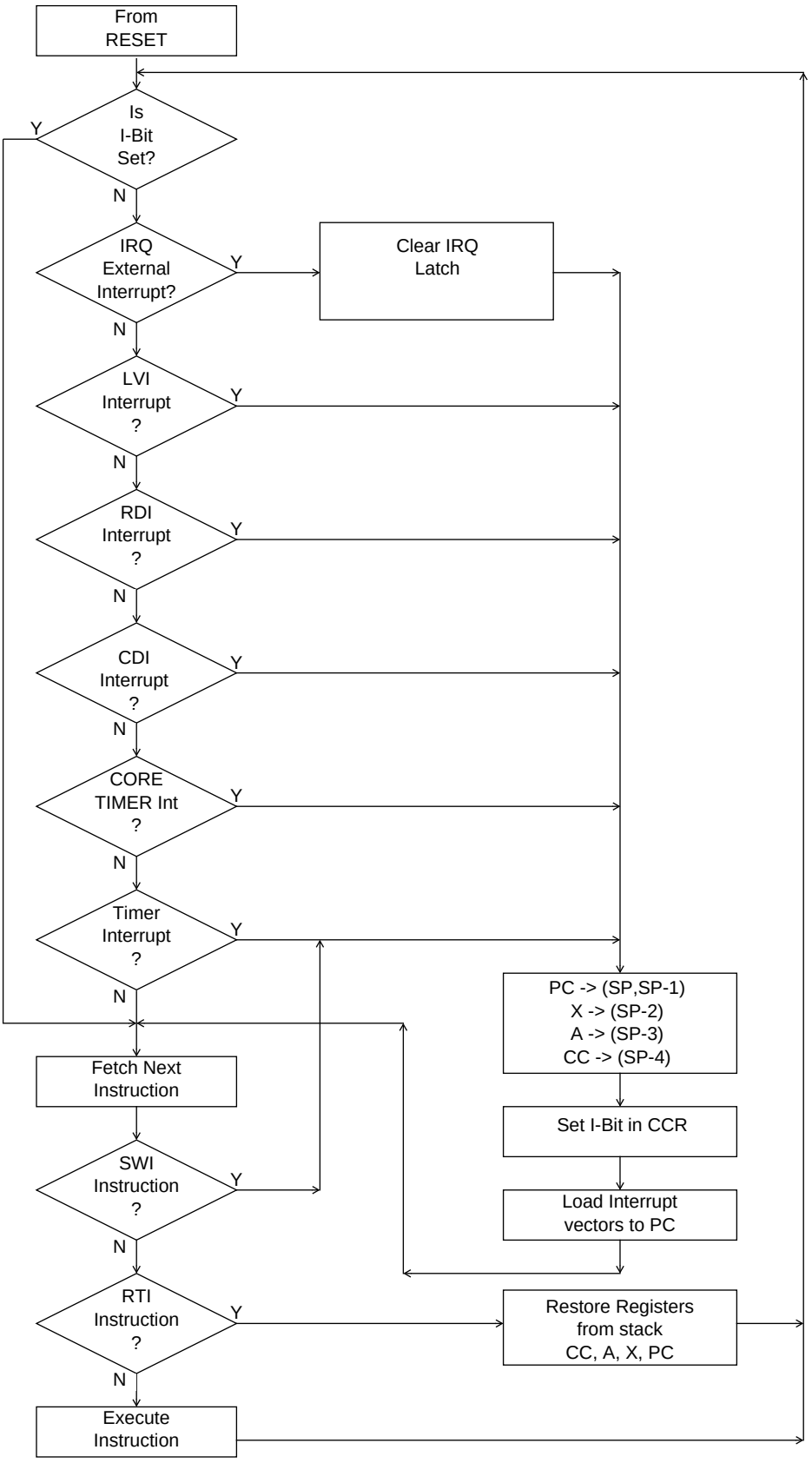


Figure 4-1. Interrupt Processing Flowchart

4.2 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 $1FFE and $1FFF. 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**.

4.3 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 generated after the SWI was fetched. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD

4.4 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. The hardware interrupts are explained in the following sections.

4.5 EXTERNAL INTERRUPT (IRQ)

If the interrupt mask bit (I bit) of the CCR is set, all maskable interrupts (internal and external) are disabled. Clearing the I bit enables interrupts (subject to their individual interrupt enable control flag status). IRQ now has an independent interrupt mask bit in the Interrupt Status and Control Register (ISCR) which must also be cleared to enable its corresponding interrupt.

The interrupt mask bit operates by inhibiting the interrupt signal ***after*** the appropriate interrupt request latch. This feature allows the interrupt to be recognized and latched even if the mask is set.

When the IRQ input goes to the active level for at least one t_{ILIH}, a logic one is latched internally to signify an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch contains a logic one, and the interrupt mask bit (I bit) in the condition code register and the IRQ mask bit (IRQM) in the ISCR are both clear, then the MCU can begin the interrupt sequence. The state of the interrupt latch is reflected in the interrupt request bit (REQ) in the ISCR, and is automatically cleared during interrupt processing. see **Figure 4-2** .

IRQ interrupt requests are automatically acknowledged and cleared during interrupt processing. It may also be cleared through software by setting the acknowledge bit in the ISCR. Setting this bit is a “one-shot” operation and will not effect subsequent interrupt operation. The action of clearing the acknowledge bit will

clear the request bit. This allows the programmer the option to cancel spurious interrupts which occur while the interrupt mask bits are set. This may be necessary in systems where it is desirable to prevent redundant (ghost) entries to the interrupt service routine (where the interrupt mask is eventually cleared).

Note that the IRQM is cleared (enabled) during reset, though no interrupts can occur until the interrupt mask bit (I-bit) of the CCR is cleared (it is set during reset). The interrupt request latches are also cleared during reset.

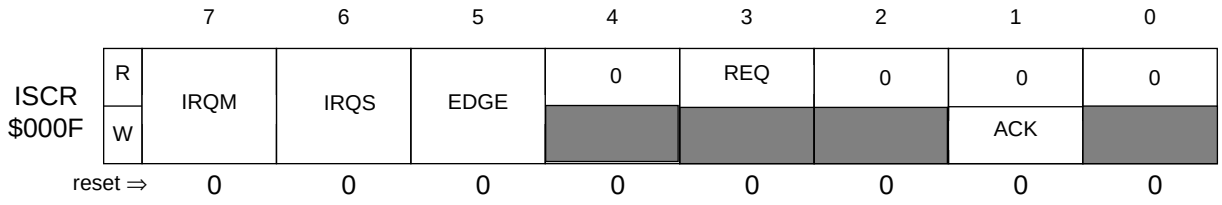


Figure 4-2. Interrupt Status and Control Register

IRQM - IRQ Enable Mask

The IRQM bit is a read/write bit which will disable the IRQ interrupt when set. IRQM is cleared by reset.

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

IRQS - IRQ Sensitivity

The IRQS bit is a read/write bit which will select whether the IRQ interrupt is edge-sensitive only or both edge-sensitive and level-sensitive. IRQS is cleared by reset.

1 = both edge-sensitive and level-sensitive
0 = edge-sensitive only

EDGE - IRQ Active Edge Select

The EDGE bit is a read/write bit which allows the user to select which edge, rising or falling, of the signal at the IRQ pin will generate an interrupt. Both rising and falling edge sensitivity may be achieved in software by toggling the EDGE bit from within the IRQ service routine. EDGE is cleared by reset.

1 = Rising edge IRQ interrupt
0 = Falling edge IRQ interrupt

REQ - IRQ Interrupt Request

The REQ bit is a read-only bit. The IRQ interrupt request bit and latch are cleared during IRQ exception processing. Therefore, one external IRQ interrupt pulse can be latched and subsequently serviced as soon as the I bit is cleared. REQ will be cleared by reset.

1 = IRQ interrupt request pending
0 = No IRQ interrupt request pending

ACK - IRQ Interrupt Request Acknowledge

This bit is write only - it will always read as a zero. Writing a one to this bit will acknowledge the interrupt by clearing the corresponding interrupt request bit.

NOTE

The use of separate request and acknowledge bits allows the safe use of read-modify-write instructions (e.g. BSET, BCLR) on the ISCR register.

NOTE

When the Edge and Level-Sensitive Mask Option is selected, the voltage applied to the $\overline{IRQ}$ pin must return to the inactive state before the RTI instruction in the interrupt service routine is executed. If the $\overline{IRQ}$ pin remains in at the active level, the interrupt service routine will be re-entered after the RTI is executed. Setting the ACK bit will have no effect under these circumstances.

4.5.1 Keyboard Interrupt (KBI)

The KBI interrupt is generated by the Keyboard Interrupt circuit in PORTA as described in **Section 7**. The interrupt enable bit for the KBI interrupt is located at bit 4-7 of Keyboard Interrupt Register (KBIR) at $0011. The I-bit in the CCR must be clear in order for the KBI interrupt to be enabled. This address for the interrupt vector is shared with the IRQ located at $1FFA and $1FFB.

		7	6	5	4	3	2	1	0
KBIR $0011	R	KBE7	KBE6	KBE5	KBE4	0	KEDGE	KBIF	0
	W								KBIC
reset		0	0	0	0	0	0	0	0

4.5.2 Low Voltage Interrupt

The LVI is generated by the low voltage detect circuit when the supply voltage falls below the LVI threshold (LVI_{th}). Enable and flag bits for this interrupt are located in the LVSCR register located at address $0010. The I-bit in the CCR must be clear in order for the LVI interrupt to be enabled. This interrupt will vector to the interrupt service routine located at the address specified by the contents of memory locations $1FF8 and $1FF9. **The user is required to set the LVRE bit in the Option register before enabling the LVIE bit in the LVSCR register**.

4.5.3 LVI Control Register

		7	6	5	4	3	2	1	0
LVSCR $0010	READ						LVIS	LVIF	LVIE
	WRITE							0	
	RESET	U	U	U	U	U	U	0	0

LVIE — Low Voltage Reset Enable

Low Voltage Interrupt enable bit, when set will enable an interrupt to be generated when LVIF, the low voltage interrupt flag bit is set. **This bit should only be enabled after LVRE bit in the Option Register has been set.**

LVIF — Low Voltage Reset Flag

This bit is set when VDD drops below the threshold voltage of LVI_{th}. When set this bit will generate an interrupt provided the LVI enable bit (LVIE) is set. It can be reset by writing a “0” to LVIF, provided VDD is above LVI_{th}.

LVIS — Low Voltage Interrupt Status

This bit is set when VDD drops below the threshold voltage of LVI_{th}.. The user can read this bit to check if the LV interrupt is true or if it is just a glitch. The difference between LVIS and LVIF is that LVIF is latched while LVIS is not.

4.5.4 Ring/carrier Detect Interrupts (RDI/CDI)

This interrupt is caused by the Caller ID module when a valid Ring signal or a Carrier is detected. The enable and flag bits for the Ring Detect and the Carrier detect are located in CLCSR1 register. These two interrupts will vector to separate interrupt service routines located at the addresses specified by the contents of memory locations $1FF6-$1FF7 and $1FF4-$1FF5 respectively. These interrupts will wake the MCU from the Wait mode. See **Section 10**.

4.5.5 Ctimer Interrupt (CORE TIMER)

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. These two interrupts will vector to the same interrupt service routine located at the address specified by the contents of memory locations $1FF2 and $1FF3.

4.5.6 Timer Interrupt (TIMER)

The TIMER interrupt is generated by the multi-function Timer when a timer overflow, output compare or input capture has occurred as described in **Section 8**. The interrupt enable bits and flags for the Timer interrupts are located in the Timer Control Register (TCR) and Timer Status Register (TSR) located at $0012 & $0013. The I-bit in the CCR must be clear in order for the TIMER interrupt to be

enabled. These three interrupts will vector to the same interrupt service routine located at the address specified by the contents of memory locations $1FF0 and $1FF1.

SECTION 5
RESETS

The MCU can be reset from four sources:

- Power-On Reset (POR)
- COP Watchdog Reset (COPR)
- Low-Voltage Reset (LVR)
- External $\overline{\text{RESET}}$ pin

All the internal peripheral modules which drive external pins will be reset by the synchronous reset signal (RST) coming from a latch, which is synchronized to the internal bus clock and set by the any of the four reset sources.

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

The $\overline{\text{RESET}}$ pin has an Schmitt trigger stage and 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 $\overline{\text{RESET}}$ input can alter the operating mode of the MCU.

NOTE

Activation of the RST signal is generally referred to as reset of the device, unless otherwise specified.

The $\overline{\text{RESET}}$ pin can also be pulled to a low state by an internal pull-down that is activated by the internal COP Watchdog, Power-on resets and Low Voltage Reset. This $\overline{\text{RESET}}$ pin pull-down device will only be activated for four cycles of the internal clock, t_{cyc}, 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 INTERNAL RESETS

The three internally generated resets are the Power-On reset , the COP Watchdog Timer reset, and the Low-Voltage reset.

5.2.1 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 4064 t_{cyc} 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 4064 cycle delay, the RST signal will remain in the reset condition until the other reset condition(s) end.

5.2.2 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 reset will activate the internal pull-down device connected to the $\overline{\text{RESET}}$ pin for four cycle of the internal clock.

The COP Watchdog Timer is enabled or disabled by a mask option. Refer to **Section 8** for more information on COP watchdog.

5.2.3 Low Voltage Reset (LVR)

The internal LVR is generated automatically when the VDD drops below a certain level LVR_{th}. The VDD level which will trigger the LVR is specified in **Section 12**.

The LVR will activate the internal pull-down device connected to the $\overline{\text{RESET}}$ pin as long as low voltage is detected.

The LVR function is enabled or disabled by a mask option.

SECTION 6
LOW POWER MODES

The MC68HC05CL4 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 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, so that any hardware interrupt (IRQ, RDI & KBI) can 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 hardware interrupt or an externally generated RESET. When exiting the STOP Mode the internal oscillator will resume after a 4064 internal processor clock cycle oscillator stabilization delay.

6.2 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 Core Timer or a reset to be generated from the COP Watchdog Timer. The Core Timer may be used to generate a periodic exit from the WAIT Mode.

In the Wait mode, the carrier detect and the ring detect circuit can be kept active by writing to the enable bit in the CLCSR2 register before the Wait instruction is executed. 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.

6.3 DATA-RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages 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.

6.4 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, insted 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.

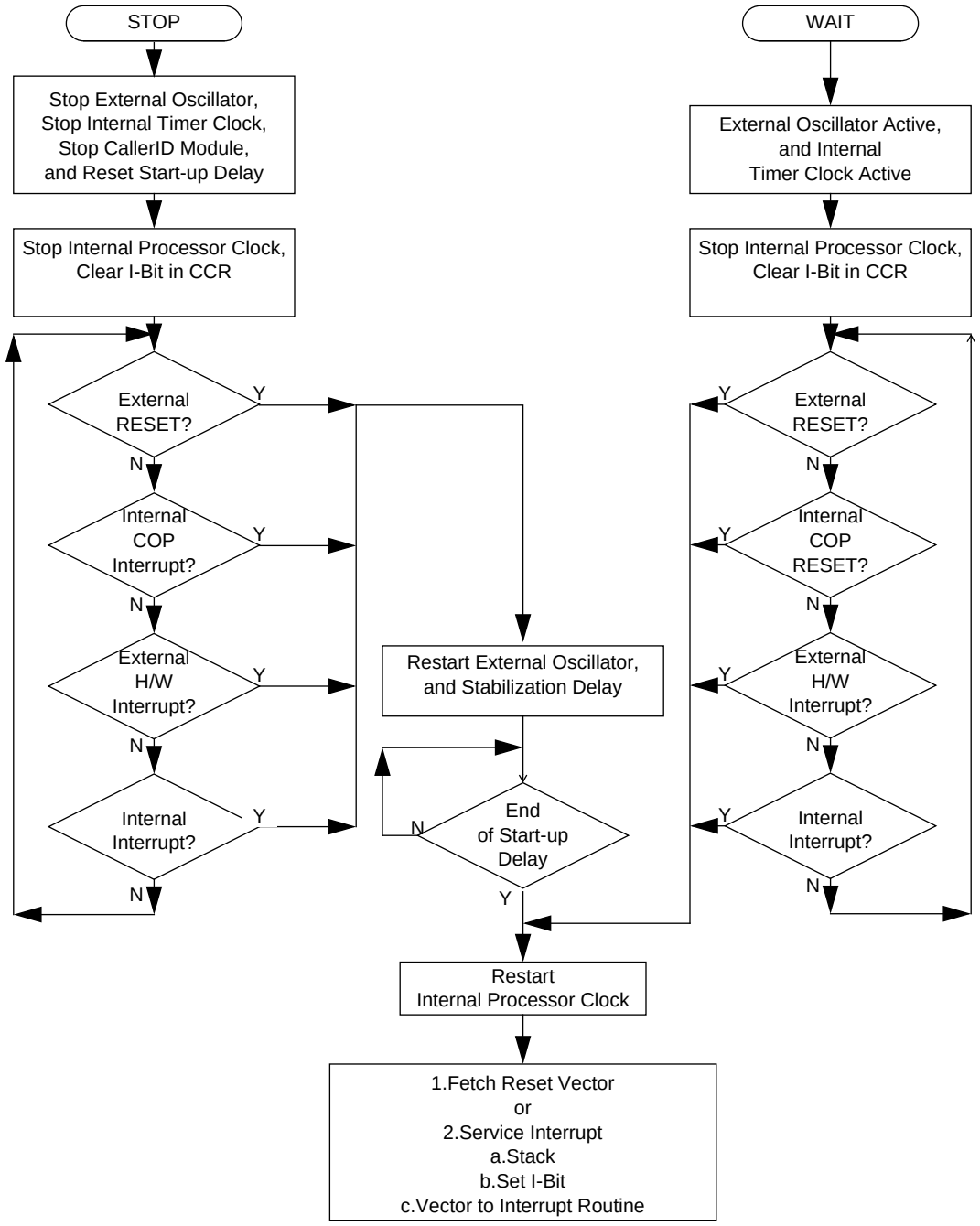


Figure 6-1. STOP/HALT/WAIT Flowcharts

SECTION 7
INPUT/OUTPUT PORTS

In the Single-Chip Mode there are 30 I/O lines fourteen of which are arranged as one 8-bit I/O port (Port A) and one 6-bit I/O port (Port B). The remaining sixteen lines arranged as two 8-bit ports, Port C and Port D. The individual bits in these ports are programmable as either inputs or outputs under software control by the data direction registers (DDRs). When Port C and D are configured as outputs, they are the LCD frontplanes. Also, if enabled by software, PA4 to PA7 will have additional functions for Keyboard Interrupt.

7.1 PARALLEL PORTS A AND B

Port A, is 8-bit bidirectional port and Port B is a 6-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**.

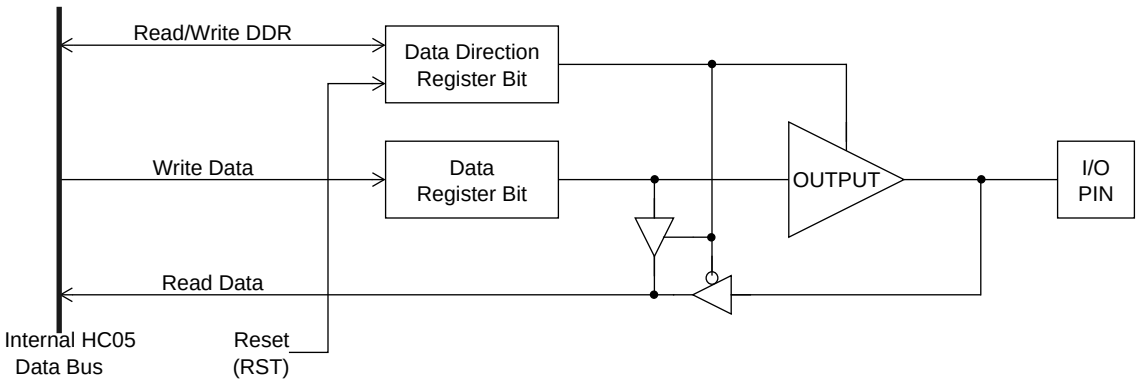


Figure 7-1. Port I/O Circuitry

Table 7-1. Port A & B I/O 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.

7.1.1 Port A and B Data Registers

Each Port I/O pin has a corresponding bit in the Port Data Register. When a Port pin is programmed as an output the state of the corresponding data register bit determines the state of the output pin. When a Port pin is programmed as an input, any read of the Port Data Register will return the logic state of the corresponding I/O pin. The locations of the Data Registers for Port A & B are at $0000, $0001. The Port data registers are unaffected by reset.

7.1.2 Port A and B Data Direction Registers

Each Port I/O pin may be programmed as an input by clearing the corresponding bit in the DDR, or programmed as an output by setting the corresponding bit in the DDR. If Keyboard interrupt enable bits are set for any of PA4 to PA7 (see **Section 7.2.3**), these bits are not affect by DDR. The DDRs for Port A and Port B are located at $0004 and $0005 respectively. The DDRs are cleared by reset.

NOTE

A “glitch” can be generated on an I/O pin when changing it from an input to an output unless the data register is first pre-conditioned to the desired state before changing the corresponding DDR bit from a zero to a one.

7.2 PORTS C AND D

Port C and D are 8-bit ports shared with LCD frontplanes FP29 - FP44. Each Port pin is controlled by the corresponding bits in a port configuration register as shown in **Figure 7-2**. The functions of these pins are summarized in **Table 7-2**.

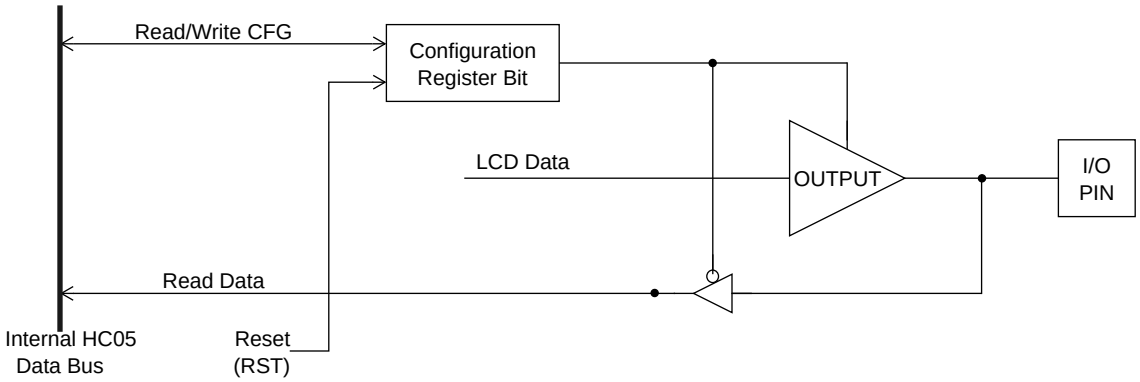


Figure 7-2. Port C/D Circuitry

Table 7-2. Port C and D Pin Functions

R/$\overline{W}$	CFG	I/O Pin Functions
0	0	The I/O pin is in input mode.
1	0	The state of the I/O pin is read.
0	1	Pin is an LCD Frontplane output controlled by the LCD module
1	1	Pin is an LCD Frontplane output controlled by the LCD module

7.2.1 Port C and D Data Registers

When a Port pin is configured as an input, any read of the Port Data Register will return the logic state of the corresponding I/O pin. The locations of the Data Registers for Port C & D are at $0002, $0003. The Port data registers are unaffected by reset. When the port is configured as LCD output any read of the port will return a zero value.

7.2.2 Port C and D Configuration Registers

Each Port C and D pin may be configured as an input by clearing the corresponding bit in the CFG, or programmed as an LCD output by setting the corresponding bit in the CFG. The CFG for Port C & D are located at $0006, $0007. The CFGs are cleared by reset.

7.2.3 Keyboard Interrupt

Keyboard Interrupt function is associated with bit 4 to bit 7 of Port A. The function is enabled by setting the keyboard interrupt enable bits KBIE4 - KBIE7 (Bits 4 - 7 in KBI Register at $0011).When a KBIE bit is set, **the corresponding Port A pin**

will be configured into an input pin, and a pull-up resistor is connected to the pin. 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'1' to the KBIC bit (Bit 0 of KBI Register) 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 can be configured to be either negative-edge sensitive or level sensitive. The interrupt vector address is shared with the IRQ and the interrupt service routine is specified by the contents of the memory locations $1FFA and $1FFB.

7.2.4 KBI Register

		7	6	5	4	3	2	1	0
KBIR $0011	R	KBE7	KBE6	KBE5	KBE4	0	KEDGE	KBIF	0
	W								KBIC
reset		0	0	0	0	0	0	0	0

KBIC—Keyboard Interrupt Clear

Writing a'1' to this bit will clear the keyboard interrupt flag latch. This bit should be cleared in the keyboard interrupt service routine, or the CPU will keep on serving this interrupt. This bit always reads'0'.

KBIF—Keyboard Interrupt Flag

Key board interrupt flag, this bit is set when a Port A pin PA4-PA7 is pulled low provided the corresponding KBIE bit is set. When this bit is set an interrupt is generated. This bit should be cleared in the keyboard interrupt service routine, or the CPU will keep on serving this interrupt.

KEDGE—Keyboard Interrupt Edge

When this bit is set the key board interrupt is negative edge sensitive and when this bit is clear the interrupt is level sensitive/edge sensitive.

KBE7-4—Keyboard Interrupt Enables 7-4

Port A I/O lines selected by KBE4-KBE7 will be configured as input lines with an internal pull-up resistor of **50 KΩ**. Once a High to Low transition is sensed on any of the input lines of PA4-PA7 selected by the KBI register, provided the interrupt mask bit of the condition code register is cleared, a keyboard interrupt is generated. Power-on or external reset will clear these bits.

SECTION 8
TIMERS

The MCU has two independent timers, the multi-functional timer and the core timer.

8.1 MULTI-FUNCTIONAL TIMER

The timer consists of a 16-bit free-running counter 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.

8.1.1 Counter

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 microseconds if the internal bus clock is 2.0 MHz. 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, $18-$19 (counter register) or $1A-$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter ($19, $1B) 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) ($18, $1A), the LSB ($19, $1B) 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 ($19 or $1B) 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.

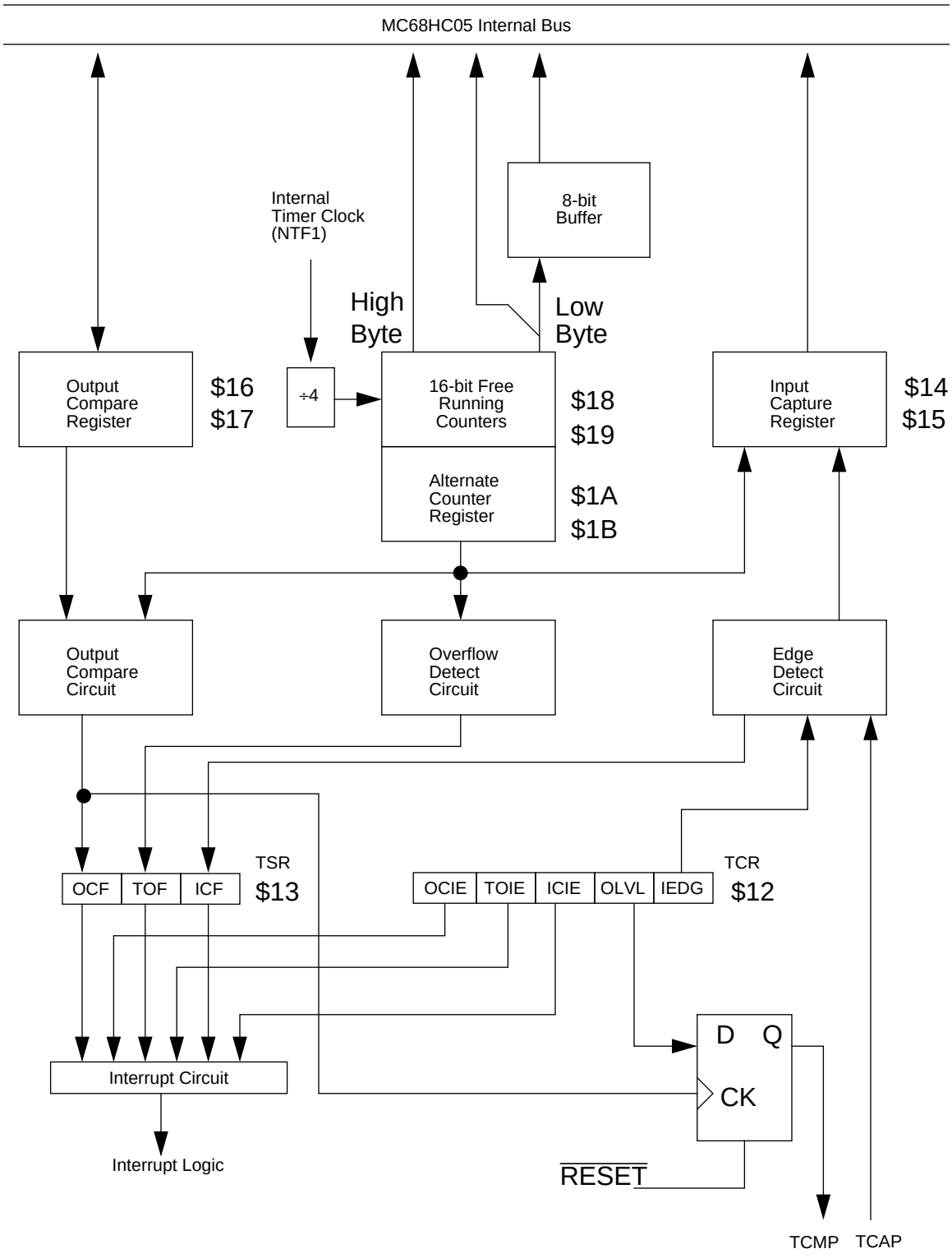
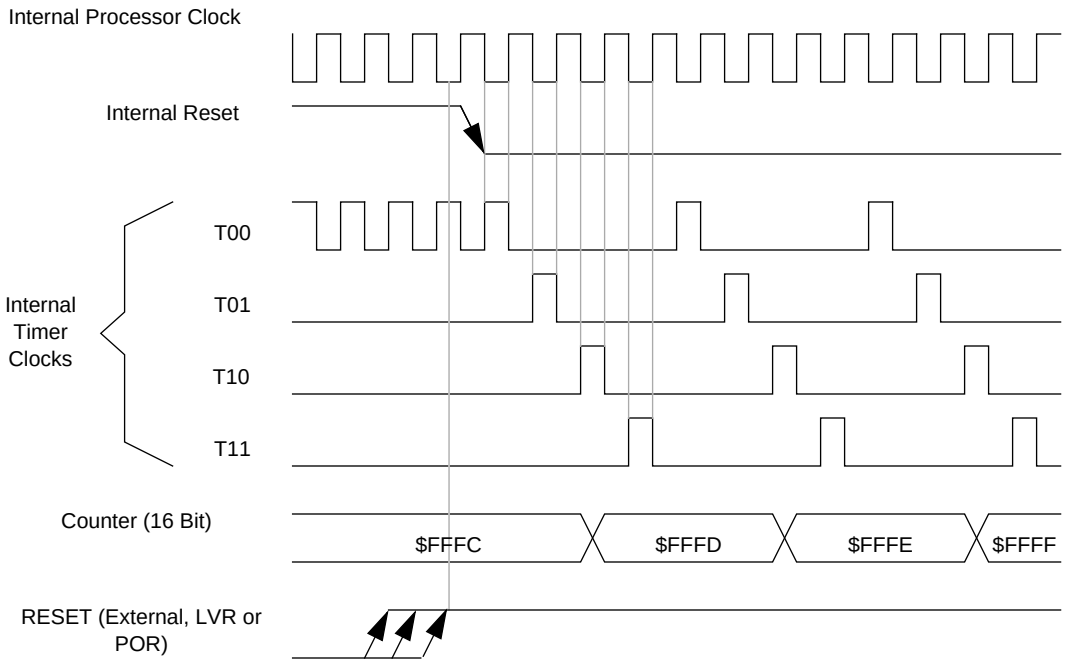
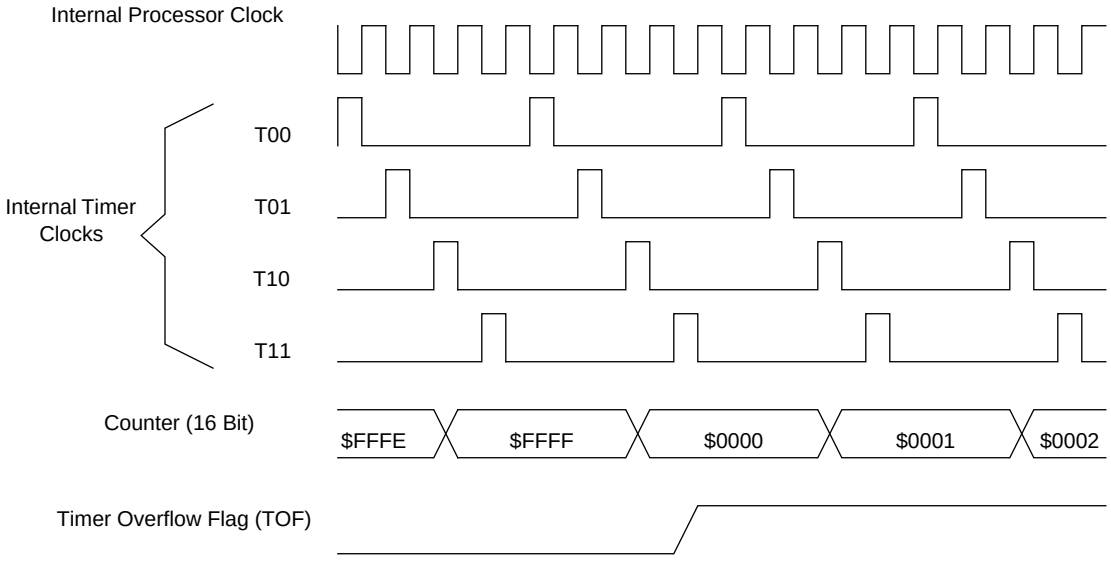


Figure 8-1. Timer Block Diagram



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

Figure 8-2. Timer State Timing Diagram for Reset



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 high time followed by a read of the counter low register.

Figure 8-3. Timer State Timing Diagram for Timer Overflow

		7	6	5	4	3	2	1	0
CNTH $0018	R	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8
	W								
	reset	1	1	1	1	1	1	1	1

		7	6	5	4	3	2	1	0
CNTL $0019	R	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
	W	Counter Reset							
	reset	1	1	1	1	1	1	0	0

		7	6	5	4	3	2	1	0
ACNTH $001A	R	ACNT15	ACNT14	ACNT13	ACNT12	ACNT11	ACNT10	ACNT9	ACNT8
	W								
	reset	1	1	1	1	1	1	1	1

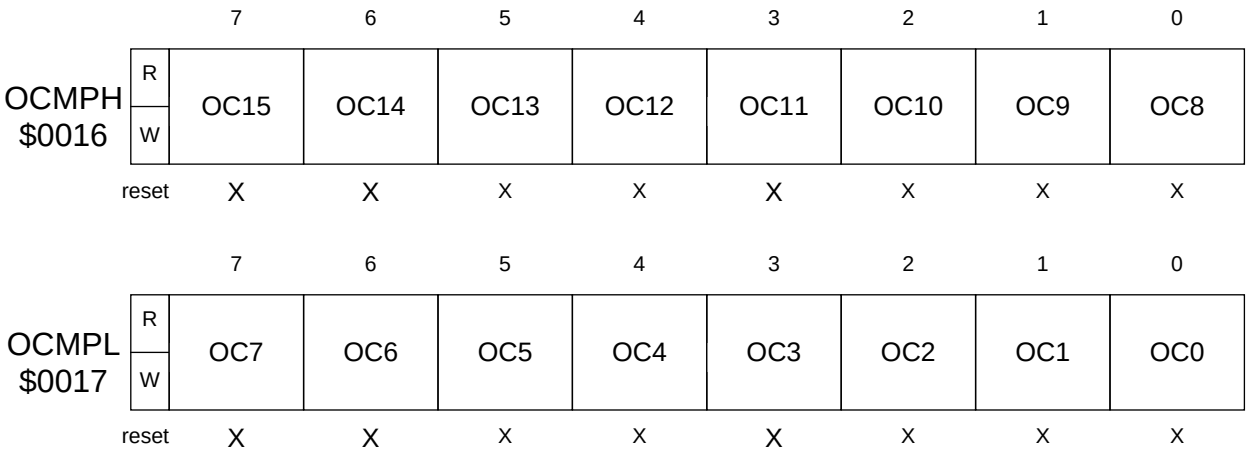
		7	6	5	4	3	2	1	0
ACNTL $001B	R	ACNT7	ACNT6	ACNT5	ACNT4	ACNT3	ACNT2	ACNT1	ACNT0
	W	Counter Reset							
	reset	1	1	1	1	1	1	0	0

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 ($19 or $1B) is written to, the counter is configured to its reset value ($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 ($18 or $1A), then the reset counter operation terminates the access sequence.

8.1.2 Output Compare Register

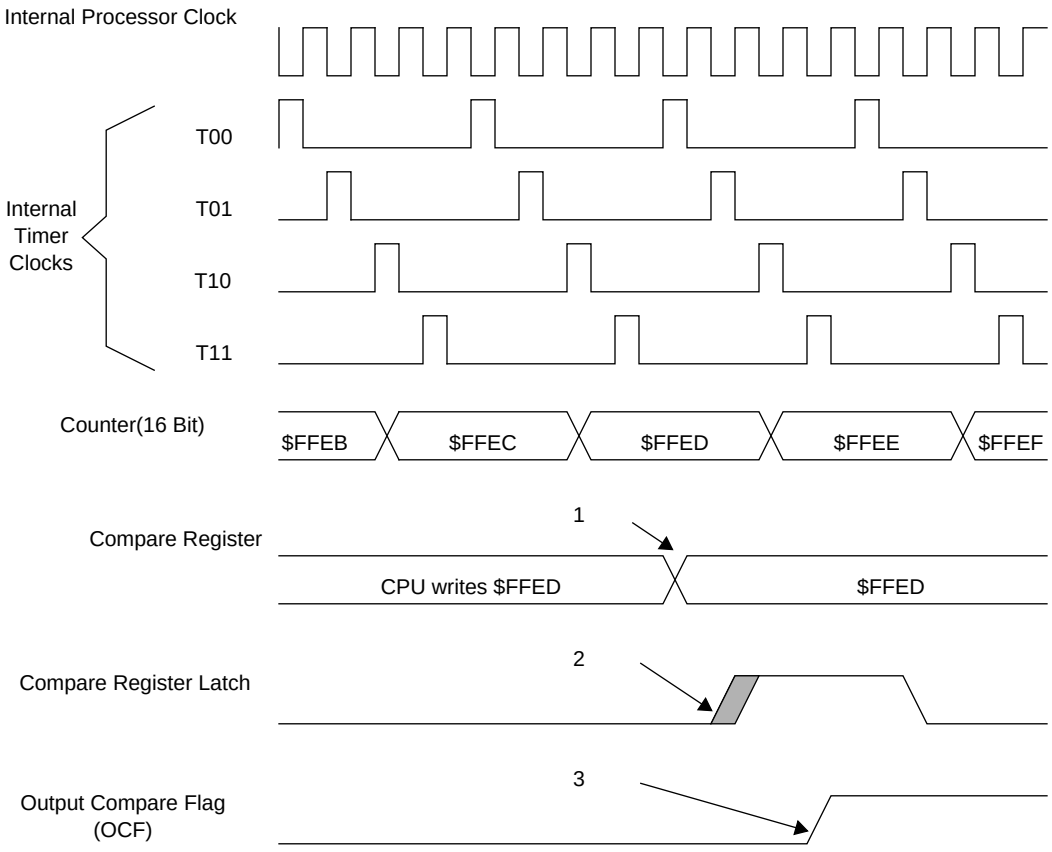


The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLVL) bit is clocked to an output level register. The output compare register values and the output level bit 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 (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) 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. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.



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 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 Register

		7	6	5	4	3	2	1	0
ICAPH $0014	R	IC15	IC14	IC13	IC12	IC11	IC10	IC9	IC8
	W								
	reset	X	X	X	X	X	X	X	X

		7	6	5	4	3	2	1	0
ICAPL $0015	R	IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0
	W								
	reset	X	X	X	X	X	X	X	X

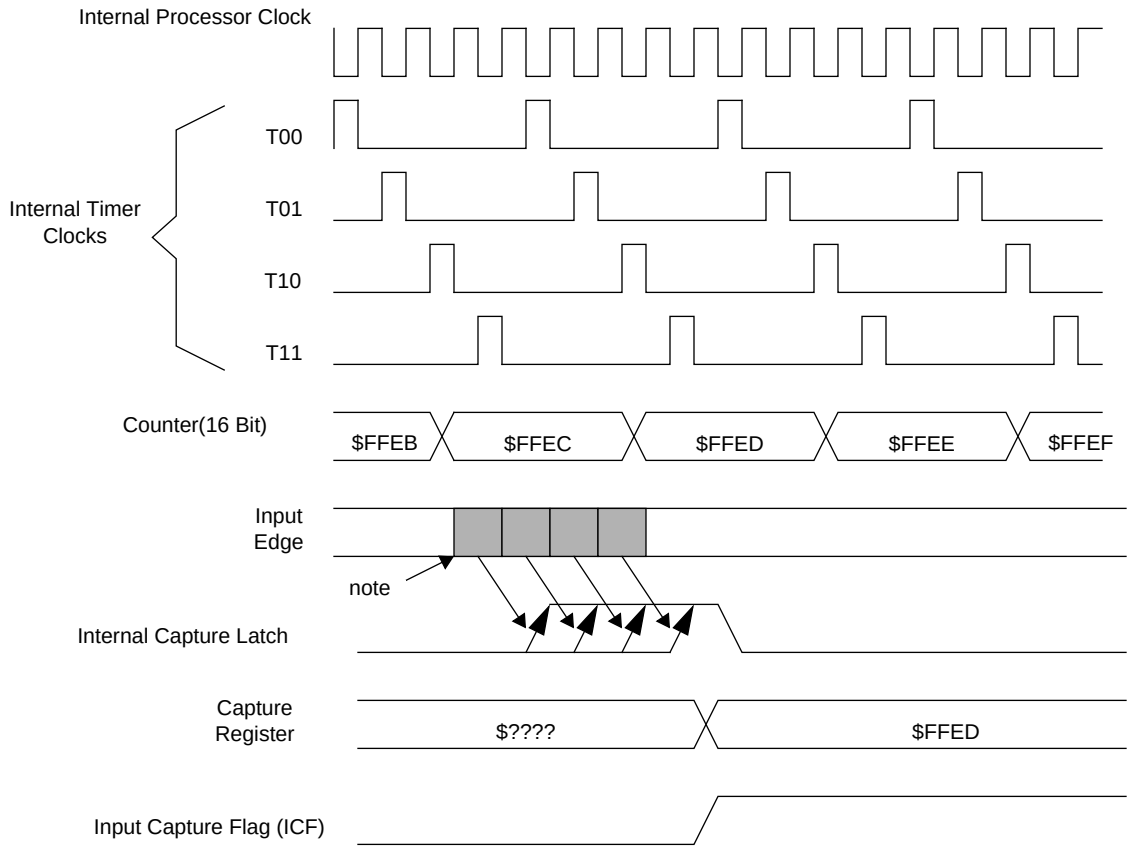
Two 8-bit registers, which make up the 16-bit input capture register, are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding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

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 (ICF) 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 ($14) MSB, the counter transfer is inhibited until the LSB ($15) 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 ($15) 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 TCR is a read/write register containing six control bits. Three bits control interrupts associated with each of the three flag bits found in the timer status register. The other two bits control: 1) which edge is significant to the input capture edge detector (i.e., negative or positive), and 2) the next value to be clocked to the output level register in response to a successful output compare. The timer control register and the free running counter are the only sections of the timer affected by reset. The TCMP pin is forced low during external reset and stays low until a valid compare changes it to high. The timer control register is illustrated below by a definition of each bit.

		7	6	5	4	3	2	1	0
TCR $0012	R	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
	W								
reset		0	0	0	0	0	0	X	1

ICIE

If the input capture interrupt enable (ICIE) bit is set, a timer interrupt is enable when the ICF status flag is set, provided the I bit in CCR is cleared. If the ICIE bit is cleared, the interrupt is inhibited. The ICIE bit is cleared by reset.

OCIE

If the output compare interrupt enable (OCIE) bit is set, a timer interrupt is enabled whenever the OCF status flag is set, provided the I bit in CCR is cleared. If the OCIE bit is cleared, the interrupt is inhibited. The OCIE bit is cleared by reset.

TOIE

If the timer overflow interrupt enable (TOIE) bit is set, a timer interrupt is enabled whenever the TOF status flag is set, provided the I bit in CCR is cleared. If the TOIE bit is cleared, the interrupt is inhibited. The TOIE bit is cleared by reset.

IEDG

The value of the input edge (IEDG) bit determines which level transition on TCAP pin will trigger a free running counter transfer to the input capture register. Reset does not affect the IEDG bit.

0 = negative edge
1 = positive edge

OLVL

The value of the output level (OLVL) bit is clocked into the output level register by the next successful output compare and will appear at TCMP pin. This bit and the output level register are cleared by reset.

0 = low output
1 = high output

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 8-2**, **Figure 8-3** and **Figure 8-4** for timing relationship to the timer status register bits.

TSR $0013

	7	6	5	4	3	2	1	0
R	ICF	OCF	TOF	0	0	0	0	0
W								
reset	X	X	X	0	0	0	0	0

ICF—Input Capture Flag

The input capture flag (ICF) is set when a proper edge has been sensed by the input capture edge detector. It is cleared by a processor access of the timer status register (with ICF set) followed by accessing the low byte ($15) of the input capture register. Reset does not affect the input compare flag.

OCF—Output Compare Flag

The output compare flag (OCF) is set when the output compare register contents matches the contents of the free running counter. The OCF is cleared by accessing the timer status register (with OCF set) and then accessing the low byte ($17) of the output compare register. Reset does not affect the output compare flag.

TOF—Timer Overflow Flag

The timer overflow flag (TOF) bit is set by transition of the free run from $FFFF to $0000. It is cleared by accessing the timer status register (with TOF set) followed by an access of the free running counter least significant byte ($19). Reset does not affect the TOF bit.

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 $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

8.1.6 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 Core Timer (or Ctimer) for this device 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 $08. A timer overflow function is implemented on the last stage of this counter, giving a possible interrupt at the rate of E/1024. Two additional stages produce the POR function at E/4064. 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 $1A.

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 $1FF0. 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.

NOTE

COP Watchdog Reset function is enabled or disabled by a mask option.

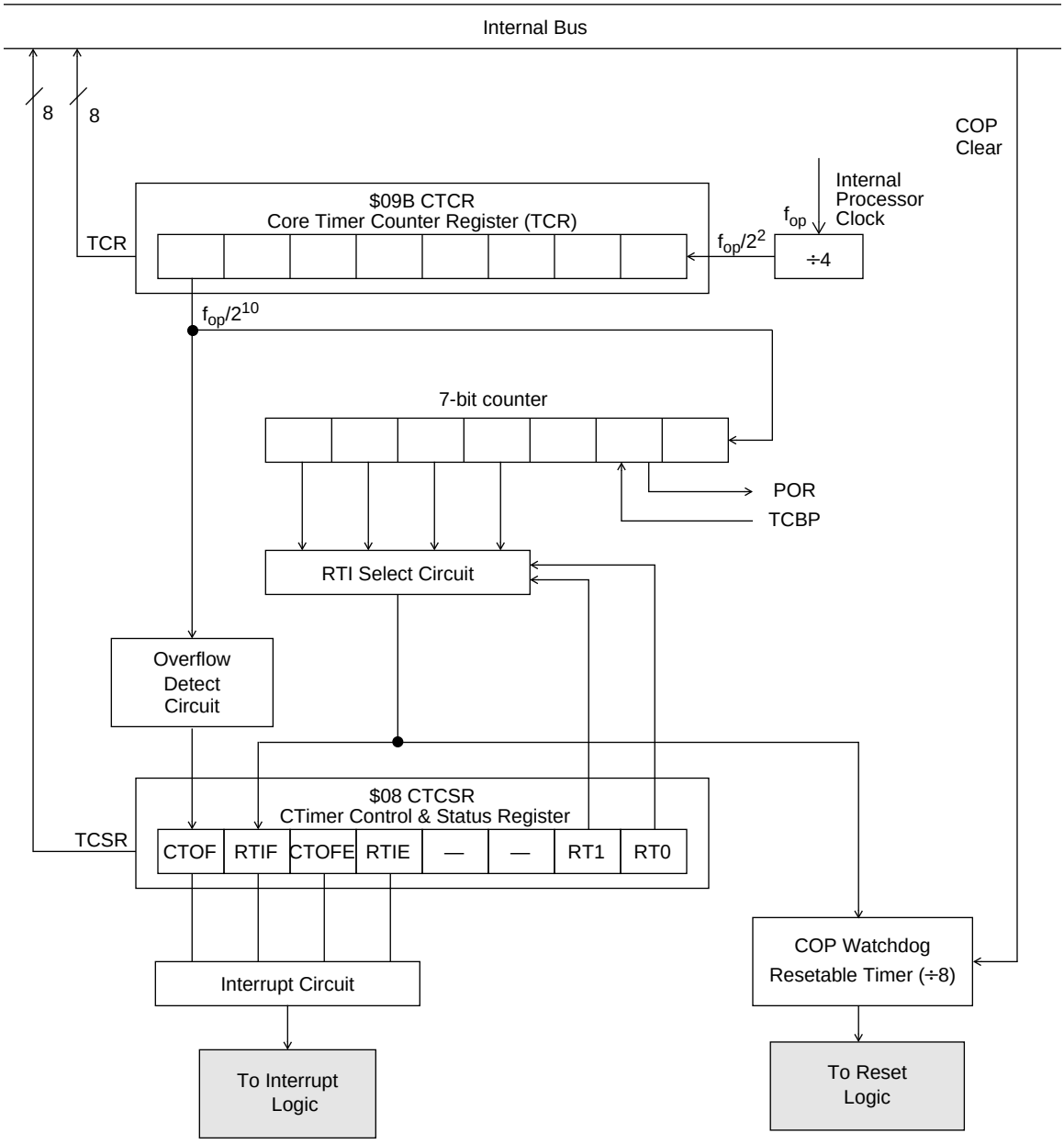


Figure 8-6. Core Timer Block Diagram

8.2.2 Ctimer Control And Status Register (CTCSR)

The TCSR 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	R	CTOF	RTIF	CTOFE	RTIE	0	0	RT1	RT0
	W								
reset		0	0	0	0	0	0	1	1

CTOF—Core Timer Overflow Flag

Core Timer Over Flow bit is a clearable, read-only status bit and is set when the 8-bit ripple counter rolls over from $FF to $00. Clearing the TOF is done by writing a “0” to it. Writing a “1” to CTOF has no effect on the bit’s value. Reset clears CTOF.

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. Real Time Interrupt Flag is a clearable, read-only status bit and is set when the output of the chosen (1 of 4 selection) stage goes active. Clearing the RTIF is done by writing a ’0’ to it. Writing a ’1’ to RTIF has no effect on this bit. Reset clears RTIF.

CTOFE—Core Timer Overflow Enable

When the Core Timer Over Flow Enable bit is set, a CPU interrupt request is generated when the TOF bit is set, provided the I bit in CCR is cleared. Reset clears this bit.

RTIE—Real Time Interrupt Enable

When the Real Time Interrupt Enable bit is set, a CPU interrupt request is generated when the RTIF bit is set, provided the I bit in the CCR is cleared. Reset clears this bit.

RT1:RT0—Real Time Interrupt Select Bits

These two Real Time Interrupt Rate Select 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.

CTCR $0009

	7	6	5	4	3	2	1	0
R	CT7	CT6	CT5	CT4	Ct3	Ct2	Ct1	CT0
W								
reset	0	0	0	0	0	0	0	0

The power-on cycle clears the entire counter chain and begins clocking the counter. After 4064 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 anytime 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 4064 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

	BUS FREQUENCY = 2.10 MHz		
RT1:RT0	**Div. Ratio**	**RTI Rate**	**COP Rate (RTI ×7)**
00	2^{14}	7.81 ms	54.7 ms
01	2^{15}	15.6 ms	109 ms
10	2^{16}	31.2 ms	219 ms
11	2^{17}	62.5 ms	438 ms

SECTION 9
LCD DRIVER

The LCD driver module supports a 45 frontplane by 8 backplane or 49 frontplane by 4 backplane display. This allows a maximum of 360 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, and all the port pins which are shared with this subsystem are configured as inputs. **Table 9-1** shows a block diagram of the LCD subsystem.

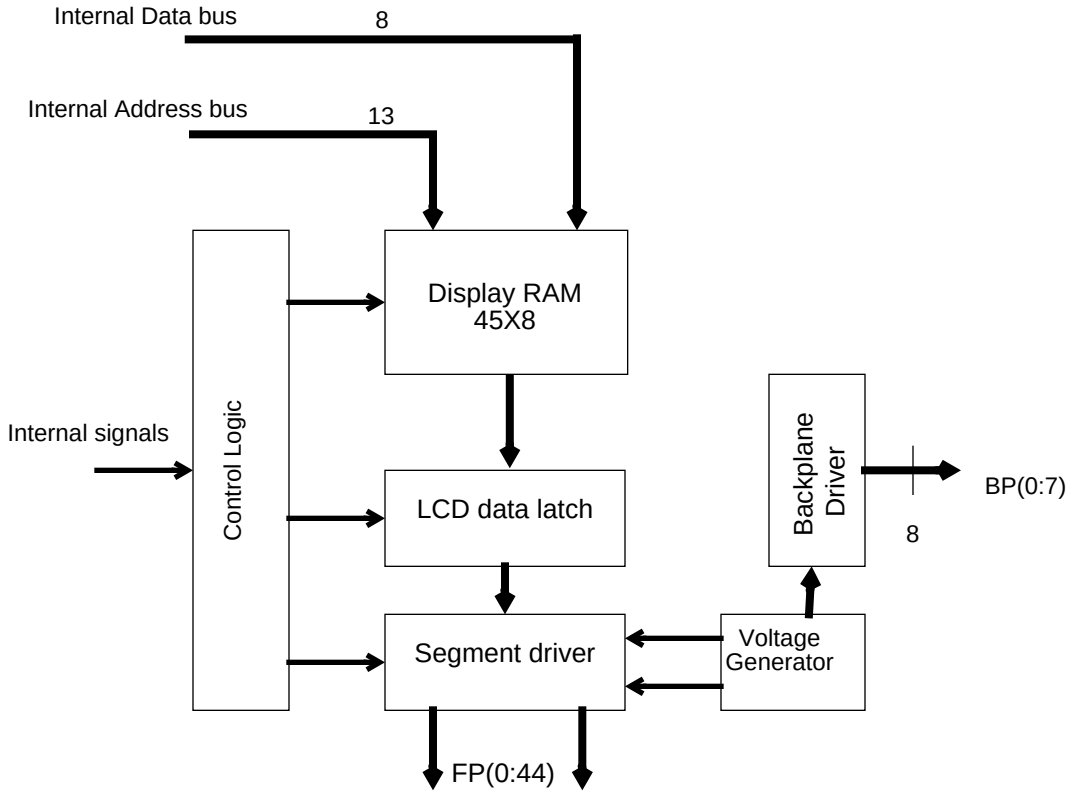


Figure 9-1. LCD Block diagram.

9.1 LCD RAM.

The data to be displayed by the LCD is written to a 45 byte display RAM located at $20 in the memory map. The bits are organized according to **Section 9-1**. 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 configured to operate with four backplanes only bit0-bit4 of each byte in the display RAM are used expect for the four bytes at $0020-$0023 where bit4-bit7 are used for the extra frontplanes which replace backplanes 4-7 on the pins. When the display is disabled, the LCD RAM can be used as on-chip RAM.

Table 9-1. LCD RAM Organization.

ADDR	DATA							
	0	1	2	3	4	5	6	7
$0020	FP0-BP0	FP0-BP1	FP0-BP2	FP0-BP3	FP0-BP4 FP45-BP0	FP0-BP5 FP45-BP1	FP0-BP6 FP45-BP2	FP0-BP7 FP45-BP3
$0021	FP1-BP0	FP1-BP1	FP1-BP2	FP1-BP3	FP1-BP4 FP46-BP0	FP1-BP5 FP46-BP1	FP1-BP6 FP46-BP2	FP1-BP7 FP46-BP3
$0022	FP2-BP0	FP2-BP1	FP2-BP2	FP2-BP3	FP2-BP4 FP47-BP0	FP2-BP5 FP47-BP1	FP2-BP6 FP47-BP2	FP2-BP7 FP47-BP3
$0023	FP3-BP0	FP3-BP1	FP3-BP2	FP3-BP3	FP3-BP4 FP48-BP0	FP3-BP5 FP48-BP1	FP4-BP6 FP48-BP2	FP4-BP7 FP48-BP3
	.							
$n	FPn-BP0	FPn-BP1	FPn-BP2	FPn-BP3	FPn-BP4	FPn-BP5	FPn-BP6	FPn-BP7
	.							
	.							
$004B	FP43-BP0	FP43-BP1	FP43-BP2	FP43-BP3	FP43-BP4	FP43-BP5	FP43-BP6	FP43-BP7
$004C	FP44-BP0	FP44-BP1	FP44-BP2	FP44-BP3	FP44-BP4	FP44-BP5	FP44-BP6	FP44-BP7

9.2 LCD OPERATION

Table 9-2 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. 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 $0A) is set.

The LCD drivers can be configured to operate with either 8 backplanes or 4 backplanes under software control.

The bias ratio is 1/4 for a 8 backplane LCD. The voltage levels required are generated internally by a resistive divider between Vdd and Vss.

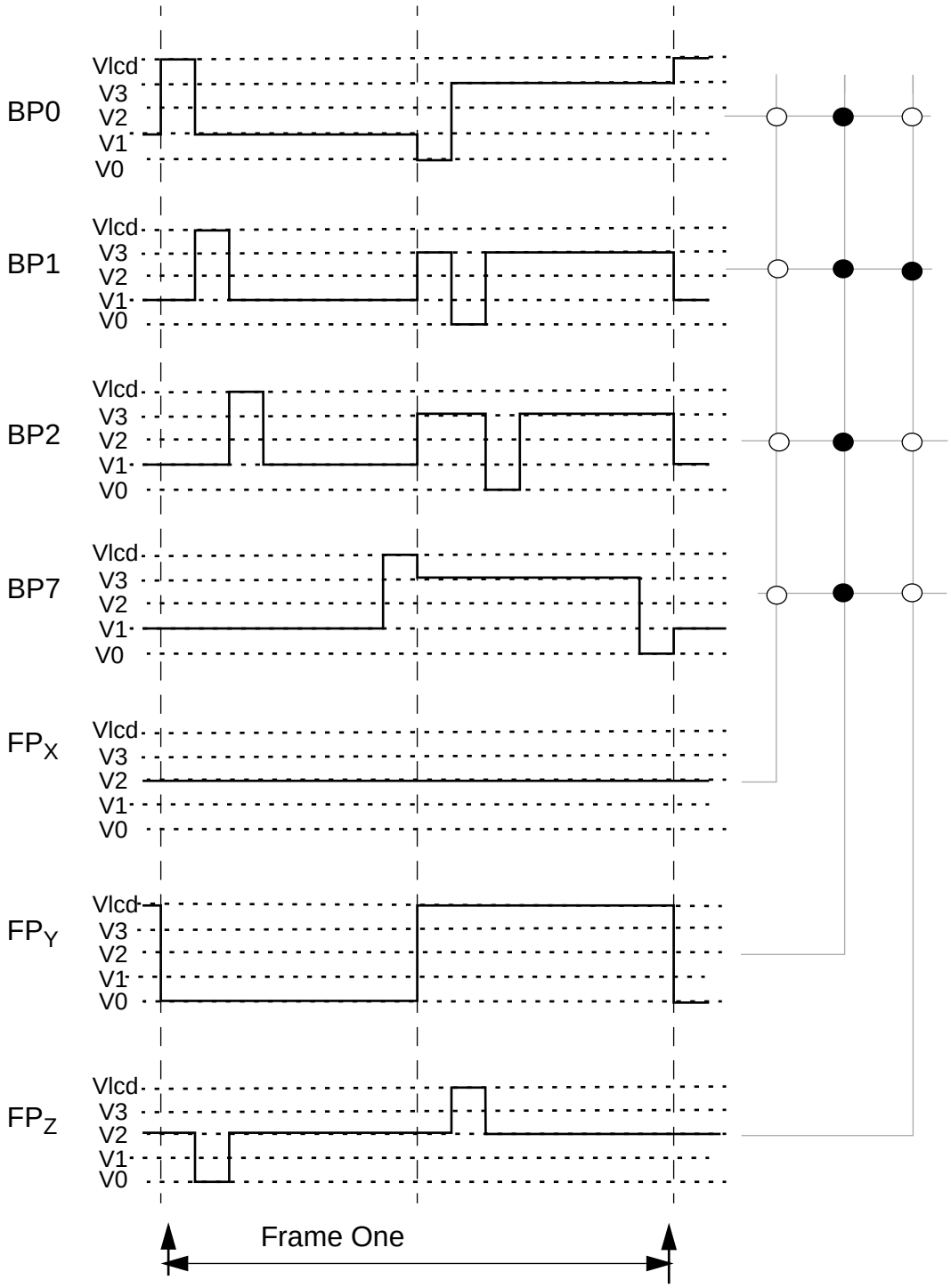


Figure 9-2. LCD Waveforms with 8 back planes.

9.3 LCD VOLTAGE GENERATION

Figure 9-3. Voltage Generation 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 drive waveforms and hence the contrast of the LCD can be altered by selecting appropriate resistors by setting the corresponding values to the CC1 to CC3 bits in the LCDCTR register.

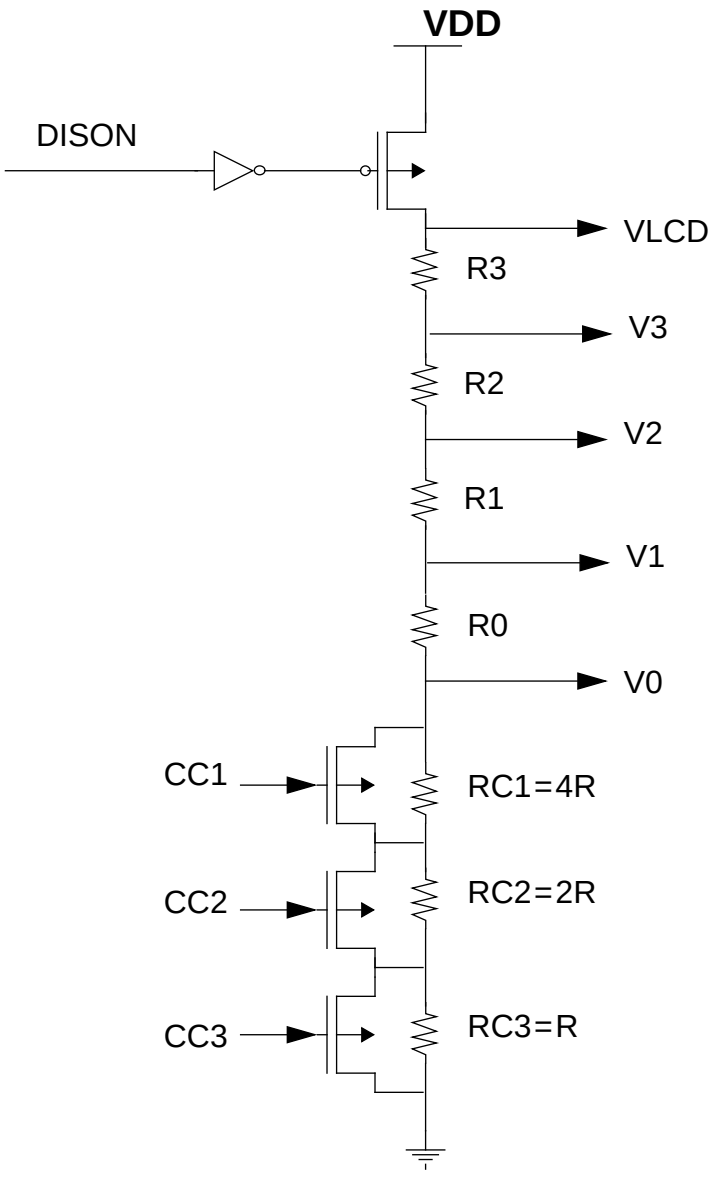


Figure 9-3. Voltage Generation

9.4 LCD CONTROL REGISTER (LCDCTR)

		7	6	5	4	3	2	1	0
LCDCTR1 $000A	READ	CC3	CC2	CC1		MX4	FC	LC	DISON
	WRITE								
	RESET	0	0	0	U	0	0	0	0

U = UNAFFECTED

DISON—Display On

The Display is on when this bit is ’1’ and off when this bit is ’0’. Setting this bit to ’0’ also disconnects the voltage generator resistor chain from VDD, thus reducing power.

LC, FC

If the quality of the display is not critical, the LCD block on the device can be put into a low current mode using the FC and LC bits in the LCD register. By selecting the appropriate values for each bit, as shown in **Table 9-2**, an extra resistor can be added to the divider chain, hence reducing the current consumption. When normal quality is required for the display, the Fast Charge option should be used which resumes default resister values for a pre-determined period in each frame. The default value of these resistors is approximately 30 KΩ.

Table 9-2. Voltage Divider Resistor Options

LC	FC	ACTION
0	0	Default value of ≈30KΩ
0	1	No Action
1	0	Resistor value ≈30KΩ per resistor selected
1	1	Fast-Charge, for a period lcdclk/128 in each time slot the resistor values are reduced to default

MX4

When this bit is set the system operates with 49 frontplanes and 4 backplanes, when cleared the system operates with 45 frontplanes and 8 backplanes.

CC1 - CC3—Contrast Control

These bits can be used to select the values of the contrast control resistors. When set to 1 the corresponding resistor will be shorted. On reset these bits are set to 0.

SECTION 10
CALLER ID

This section describes the Caller ID module of the MC68HC05CL4.

10.1 INTRODUCTION.

The Caller ID module demodulates the Bell 202 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 10-1**.

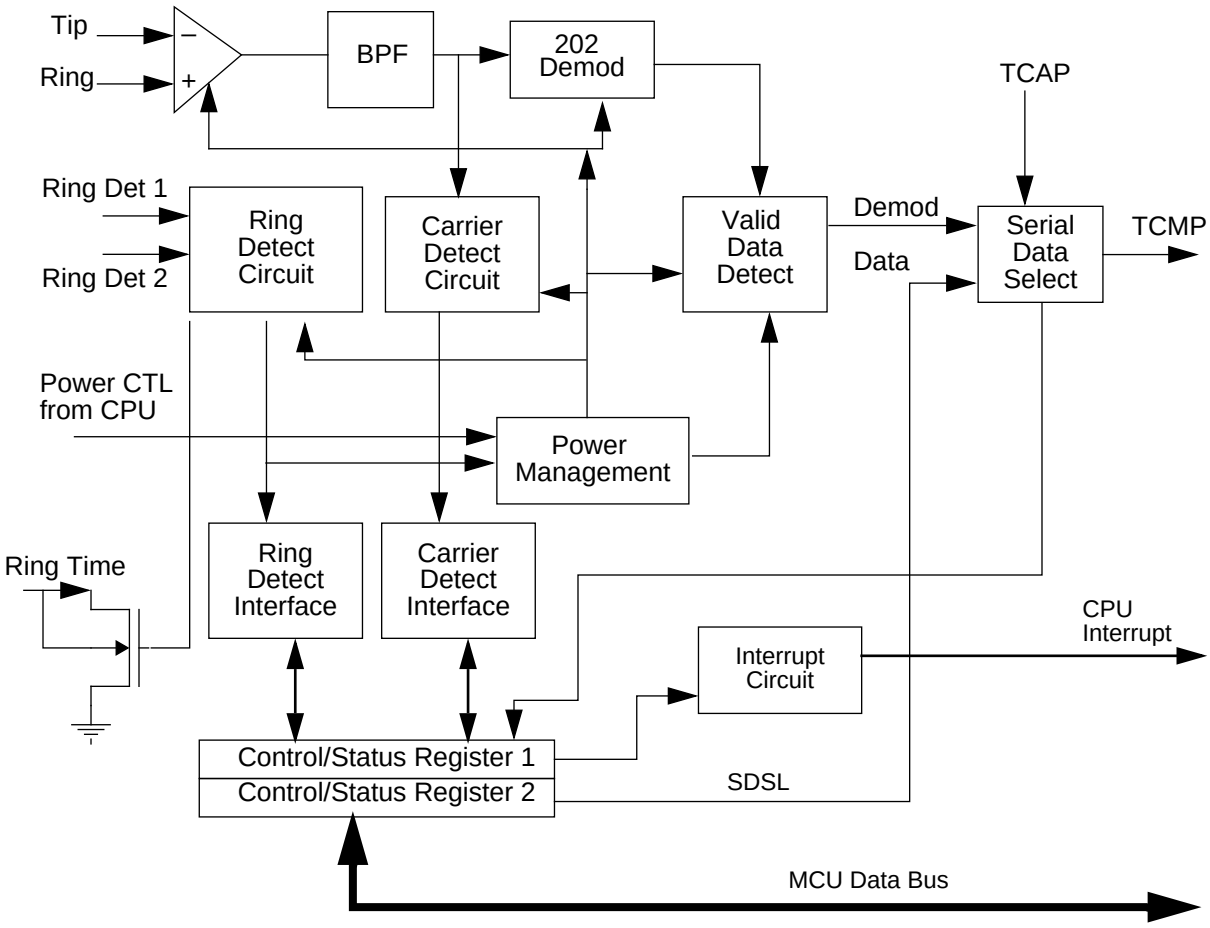


Figure 10-1. CLID Block Diagram

10.1.1 FSK Demodulator

The demodulator recovers the FSK data transmitted over the telephone line. It first band limits the incoming signal with a bandpass filter whose output is fed to the carrier detect threshold comparator as well as to the differential detector for demodulation. 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 10.2.2**.

10.1.2 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(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 10.2.1** and **Section 10.2.2**.

10.1.3 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 (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 10.2.1** and **Section 10.2.2**.

If the ring detect feature is not required it can be disabled by writing to RDPW bit in the CLCSR2 register. See **Section 10.2.1** and **Section 10.2.2**

10.1.4 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 $\overline{RT}$ signal is below the threshold, $R_{D2}V_T$ (see **Figure 10-5**) the oscillator circuit is forced on and the Ring Signal is validated. If the $\overline{RT}$ 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.

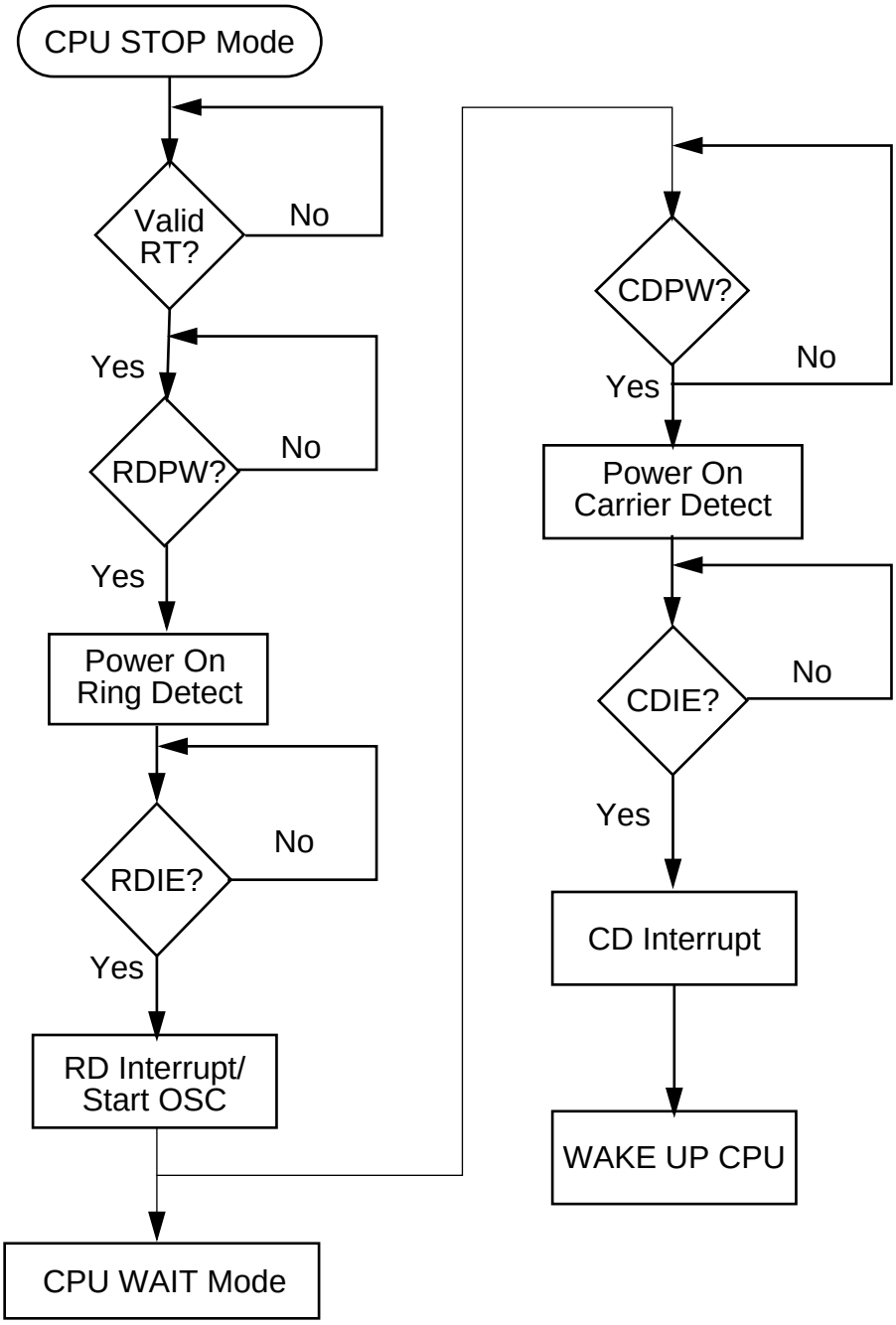


Figure 10-2. CPU-CLID Power-Up Sequence from STOP Mode

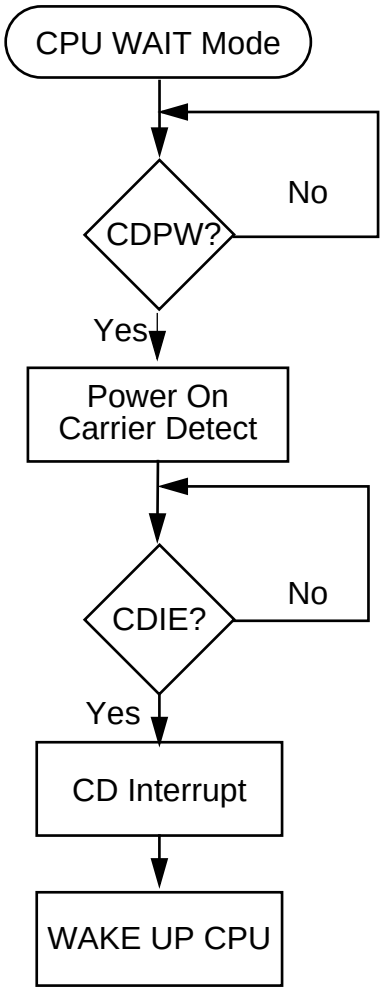


Figure 10-3. CPU-CLID Power-Up Sequence from WAIT Mode

10.1.5 Data Interface

The demodulated data from this module is available in serial. The serial data can be read from CIDSD (bit 2) of the CLCSR2 register. 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.

10.2 CALLER ID REGISTER

10.2.1 Control/Status Register1 (CLCSR1)

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

U = UNAFFECTED

CDO — 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 — 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—Carrier Detect Interrupt Enable

When enabled (set) an interrupt will be generated when a carrier is detected or forced by writing to CDO.

CDIF—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. The CDIF bit must be cleared by writing a zero.

RDIE—Ring Detect Interrupt Enable

When enabled (set) an interrupt will be generated when a carrier is detected or forced by writing to RDO.

RDIF—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. The RDIF bit must be cleared by writing a zero.

10.2.2 Control/status register 2 (CLCSR2)

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

U = UNAFFECTED

$\overline{CD}$—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.

$\overline{RD}$—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—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. The source of this data bit is selected by bit 7 in this register.

CDPW—Carrier Detect Power Up

Carrier detection is enabled when this bit is set. An interrupt will be generated when a valid carrier is detected provided the CDIE bit in the CLCSR1 register is set.

RDPW—Ring Detect Power Up

Ring detection is enabled when this bit is set. An interrupt will be generated when a valid ring is detected provided the RDIE bit in the CLCSR1 register is set.

10.2.3 Control/status register 3 (CLCSR3)

		7	6	5	4	3	2	1	0
REG $000D	READ	SDSL				**RDEDG**	**CDEDG**	RDOE	CDOE
	WRITE								
	RESET	0	U	U	U	0	0	0	0

U = UNAFFECTED

CDOE - Carrier Detect Override Enable

Setting this bit allows the user to force the carrier detect signal by writing a zero to CDO, bit 0 in the CLCSR1 register.

RDOE — Ring Detect Override Enable

Setting this bit allows the user to force the ring detect signal by writing a zero to RDO, bit 1 in the CLCSR1 register.

CDEDG

When this bit is set, the Carrier Detect Interrupt Flag (Bit 5 of CLCSR1) will be triggered at the falling edge of the carrier detect signal and will be cleared by pulling reset low. If CDEDG is kept low, the Interrupt Flag is active for as long as the carrier detect signal is valid.

RDEDG

When this bit is set, the Ring Detect Interrupt Flag (Bit 7 of CLCSR1) will be triggered at the falling edge of the ring detect signal and will be cleared by pulling reset low. If RDEDG is kept low, the Interrupt Flag is active for as long as the ring detect signal is valid.

SDSL—Serial Data Select

When this bit is zero, the CIDSD bit is the output of the on-chip demodulator. When this bit is set it disconnects the demodulator output from the CIDSD which appears at the TCMP pin and the TCAP pin now becomes the input for the CIDSD bit. This allows the external Caller ID device to interface directly with the MCU. On reset this bit is cleared.

10.3 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 10-1. Typical Input parameters

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

Table 10-2. Critical Design Characteristics

Characteristics	Typical	Unit
Input Tip/Ring Sensitivity	–40	dBm
Bandpass Filter (BPF) Frequency Response (relative to 1700 Hz @ 0 dB)		
60 Hz	–58	dB
1000 Hz	–1	dB
2400 Hz	–1	dB
≥3300 Hz	–34	dB
Carrier Detect Sensitivity	–40	dBm

Table 10-3. Switching Characteristics (V_{DD}= 5V; T_A=25 C)

Description	Symbol	Minimum	Typical	Maximum	Unit
OSC Start-up	t_{DOSC}	—	2	—	ms
Carrier Detect Acquisition	t_{DAQ}	—	14	25	ms
End of Carrier Detect	t_{DCH}	8	—	-	ms

The transmission level from the terminating “Central Office” 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.

10.4 MESSAGE FORMAT

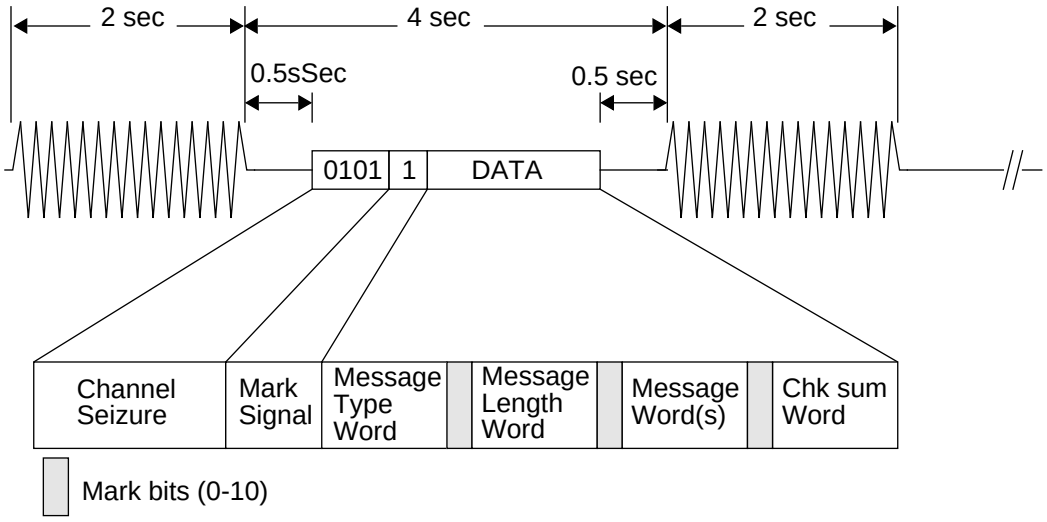


Figure 10-4. Single Message Format

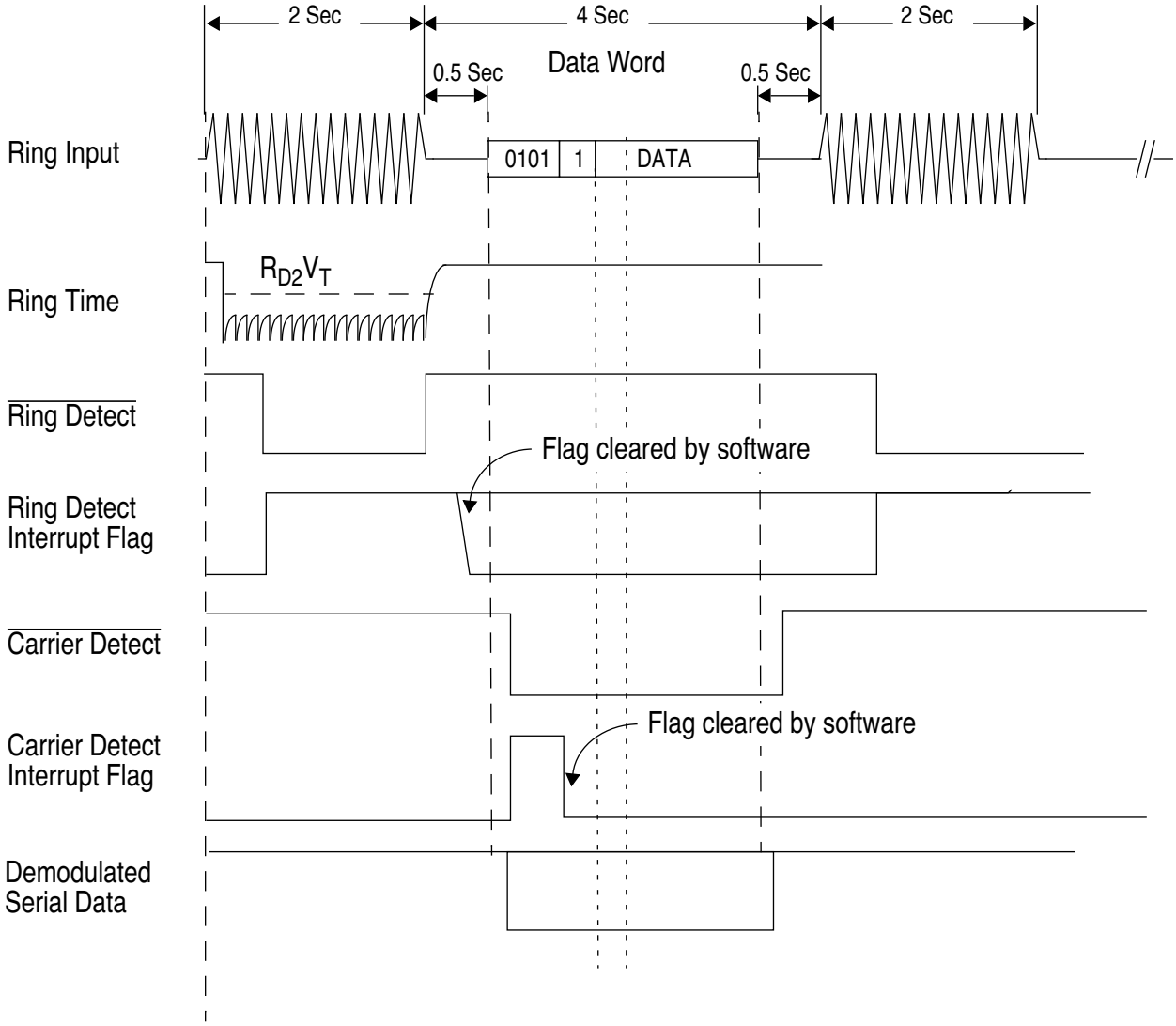


Figure 10-5. CLID Timing Diagram

SECTION 11
INSTRUCTION SET

This section describes the addressing modes and instruction types.

11.1 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

11.1.1 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.

11.1.2 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.

11.1.3 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.

11.1.4 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.

11.1.5 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.

11.1.6 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.

11.1.7 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.

11.1.8 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.

11.1.9 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

11.1.10 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 11-1** lists the register/memory instructions.

Table 11-1. 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

11.1.11 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 11-2** lists the read-modify-write instructions.

Table 11-2. 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

11.1.12 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 11-3** lists the jump and branch instructions.

Table 11-3. 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

11.1.13Bit 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 11-4** lists these instructions.

Table 11-4. Bit Manipulation Instructions

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

11.1.14Control Instructions

These register reference instructions control CPU operation during program execution. Control instructions, listed in **Table 11-5**, use inherent addressing.

Table 11-5. 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

11.1.15Instruction Set Summary

Table 11-6 is an alphabetical list of all M68HC05 instructions and shows the effect of each instruction on the condition code register.

Table 11-6. Instruction Set Summary

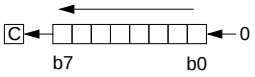
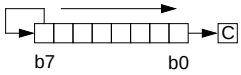
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	◊	—	◊	◊	◊	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* 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 11-6. 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 11-6. 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 11-6. Instruction Set Summary (Continued)

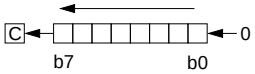
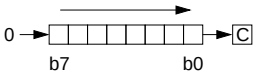
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
INC *opr* INCA INCX INC *opr*,X INC ,X	Increment Byte	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1	—	—	◊	◊	—	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff 	5 3 3 6 5
JMP *opr* JMP *opr* JMP *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 11-6. Instruction Set Summary (Continued)

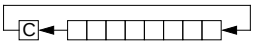
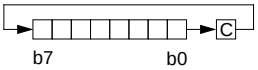
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *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 11-6. 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 11-7. 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 12
ELECTRICAL SPECIFICATION

12.1 MAXIMUM RATINGS

(Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	–0.3 to +7.0	V
Input Voltage Normal Operation Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{IN}	 V_{SS}–0.3 to V_{DD}+0.3 V_{SS}–0.3 to $2V_{DD}$+0.3	 V V
Current Drain per pin excluding VDD and VSS	I	25	mA
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{STG}	T_L to T_H –65 to +150	°C

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}).

12.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance 80-Pin PQFP	θ_{JA}	60	°C/W

12.3 DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ±10%, VSS = 0 Vdc, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output High Voltage (I_{LOAD} = –0.8 mA) PA0-PA7, PB0-PB5, TCMP	V_{OH}	V_{DD}–0.8	—	—	V
Output Low Voltage (I_{LOAD} = +1.6 mA) PA0-PA7, TCMP,	V_{OL}	—	—	0.4	V
Input High Voltage PA0-PA7, PB0-PB5, TCAP, $\overline{RESET}$, $\overline{IRQ}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB5, PC0-PC7, PD0-PD7, TCAP, $\overline{RESET}$, $\overline{IRQ}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Supply Current (see Notes) Run Wait Stop 25˚C 0˚C to +70˚C (Standard)	I_{DD}	 — — — —	 7 4.5 36 40	 8 5 50 50	 mA mA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB5	I_{IL}	—	—	10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1, TCAP	I_{IN}	—	—	1	µA
Capacitance Ports (as Input or Output), $\overline{RESET}$, $\overline{IRQ}$, OSC1, OSC2, TCAP,	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF
LCD Voltage Input	V_{LCD}	V_{ss}	—	V_{DD}	V
Low Voltage Interrupt Threshold Voltage	V_{LVI}		4.5		V
Low Voltage Reset Threshold Voltage	V_{LVR}		4.0 (100mV Hysteresis)		V
Input Threshold Voltage (CALLER ID) Positive Going: VDD = 5V (RDI1, $\overline{RT}$)	V_{T+}	2.5	2.75	3.0	V
Input Threshold Voltage (CALLER ID) Negative Going: VDD = 5V (RDI1, $\overline{RT}$)	$V_{T–}$	2.0	2.3	2.6	V
RDI2 Threshold (CALLER ID)	$R_{D2}V_T$	0.9	1.0	1.1	V
TIP/RING Input DC Resistance (CALLER ID)	R_{IN}	—	500	—	kΩ

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Timer system active, carrier detect and ring detect circuit active.
4. Run (Operating) I_{DD}, Wait $_{IDD}$: Measured using external square wave clock source to OSC1 (f_{OSC}=4.0 MHz), all inputs 0.2 VDC from rail; no DC 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 VDC, V_{IH} = V_{DD}–0.2 VDC.
6. Stop I_{DD} measured with OSC1=V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

12.4 CONTROL TIMING

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Units
Frequency of Operation Crystal Oscillator Option External Clock Source	 f_{OSC} f_{OSC}	 3.58 —	 3.68 —	 MHz MHz
Internal Operating Frequency Crystal Oscillator (f_{OSC}/2) External Clock (f_{OSC}/2)	 f_{OP} f_{OP}	 1.79 —	 1.84 —	 MHz MHz
Cycle Time (1/f_{OP})	t_{CYC}	540	560	ns
Crystal Oscillator Start-up Time (Crystal Oscillator option)	t_{OXON}	—	100	ms
Stop Recovery Start-up Time (Crystal Oscillator option)	t_{ILCH}	—	100	ms
RESET Pulse Width Low	t_{RL}	1.5	—	t_{CYC}
Power-on RESET Output Width 4064 cycle	 t_{porl}	 4064	 —	 t_{CYC}
Watchdog RESET Output Pulse Width	t_{dogl}	—	4	t_{CYC}
Watchdog time-out	t_{dog}	6144	—	t_{CYC}
Timer Resolution (note 2) Input Capture Pulse Width Input Capture Pulse Period	t_{RESL} t_{TH} t_{TL} t_{TLTL}	4 540 note 3	4 — —	t_{CYC} ns t_{CYC}
IRQ Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	125	—	ns
IRQ Interrupt Pulse Period	t_{ILIL}	note 3	—	t_{CYC}
PA4 to PA7 interrupt pulse period	t_{IHIH}	note 3	—	t_{CYC}
OSC1 Pulse Width	t	270	280	ns

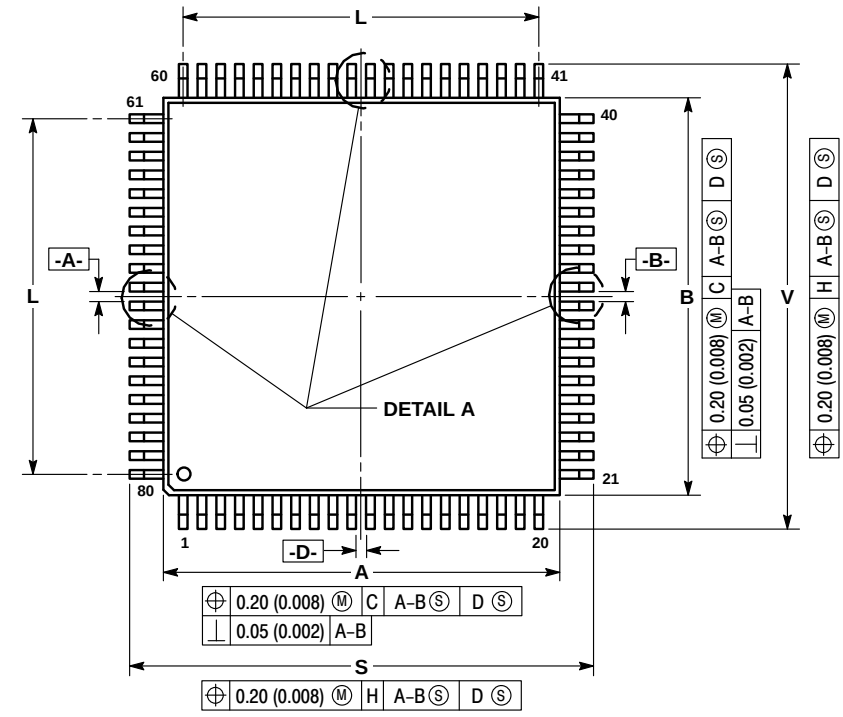
NOTES:

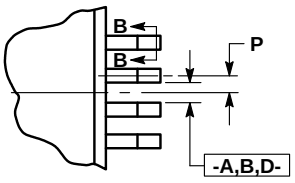
1. V_{DD} = 5.0 V_{DC} ± 10%, V_{SS} = 0 VDC, T_A = T_L to T_H
2. The 2-bit timer prescaler is the limiting factor in determining timer resolution.
3. The minimum period t_{TLTL}, t_{ILIL} or t_{IHIH} should not be less than the number of cycles it takes to execute the interrupt service routine plus 19 t_{CYC}.

SECTION 13
MECHANICAL SPECIFICATION

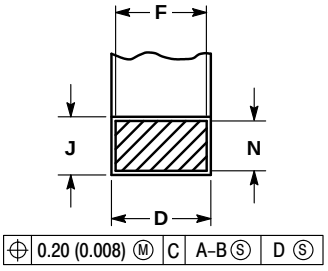
This section describes the dimensions of the 80-pin QFP package.

13.1 80-PIN QUAD-FLAT-PACKAGE (CASE 841B-01)

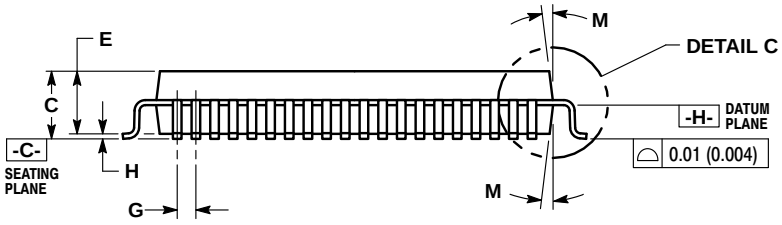


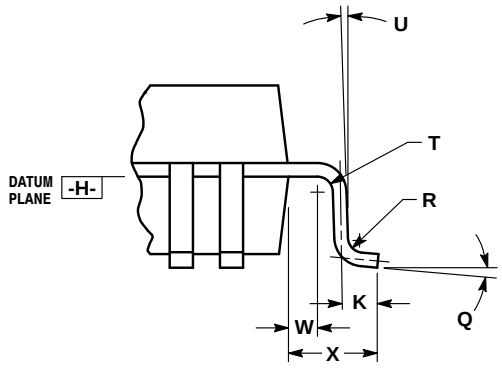


DETAIL A



SECTION B-B





DETAIL C

NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE -H- IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS -A-, -B- AND -D- TO BE DETERMINED AT DATUM PLANE -H-.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -C-.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	13.90	14.10	0.547	0.555
B	13.90	14.10	0.547	0.555
C	2.15	2.45	0.084	0.096
D	0.22	0.38	0.009	0.015
E	2.00	2.40	0.079	0.094
F	0.22	0.33	0.009	0.013
G	0.65 BSC		0.026 BSC	
H	—	0.25	—	0.010
J	0.13	0.23	0.005	0.009
K	0.65	0.95	0.026	0.037
L	12.35 BSC		0.486 BSC	
M	5°	10°	5°	10°
N	0.13	0.17	0.005	0.007
P	0.325 BSC		0.013 BSC	
Q	0°	7°	0°	7°
R	0.13	0.30	0.005	0.012
S	16.95	17.45	0.667	0.687
T	0.13	—	0.005	—
U	0°	—	0°	—
V	16.95	17.45	0.667	0.687
W	0.35	0.45	0.014	0.018
X	1.6 REF		0.06 REF	

APPENDIX A
MC68HC705CL4

This section describes the differences between the MC68HC705CL4 and the MC68HC05CL4.

A.1 INTRODUCTION

The MC68HC705CL4 is an EPROM version of the MC68HC05CL4, and is available for user system evaluation and debugging. The MC68HC705CL4 is functionally identical to the MC68HC05CL4 with the exception of the EPROM feature and the self-check routine is replaced by a bootstrap routine. Also, the mask option for COP and LVR in the MC68HC05CL4 is replaced by an Option Register in the MC68HC705CL4.

A.2 MEMORY

The MC68HC705CL4 has an 8K-byte memory map (see **Figure A-1**) consisting of user EPROM, RAM, bootstrap ROM, and I/O.

A.2.1 Option Register ($1D)

This register can only be written once following a Power-On or External Reset, but can be read at any time.

		7	6	5	4	3	2	1	0
OPT $001D	READ		COP				LVRE		
	WRITE								
POR		U	0	U	U	U	0	U	U

U = UNAFFECTED

LVRE—Low Voltage Reset Enable

1 = Low-Voltage-Reset enabled.
0 = Low-Voltage-Reset disabled.

COP—Computer Operating Properly

1 = COP watchdog reset enabled.
0 = COP watchdog reset disabled.

Refer to **Section 8** for detailed information on COP watchdog.

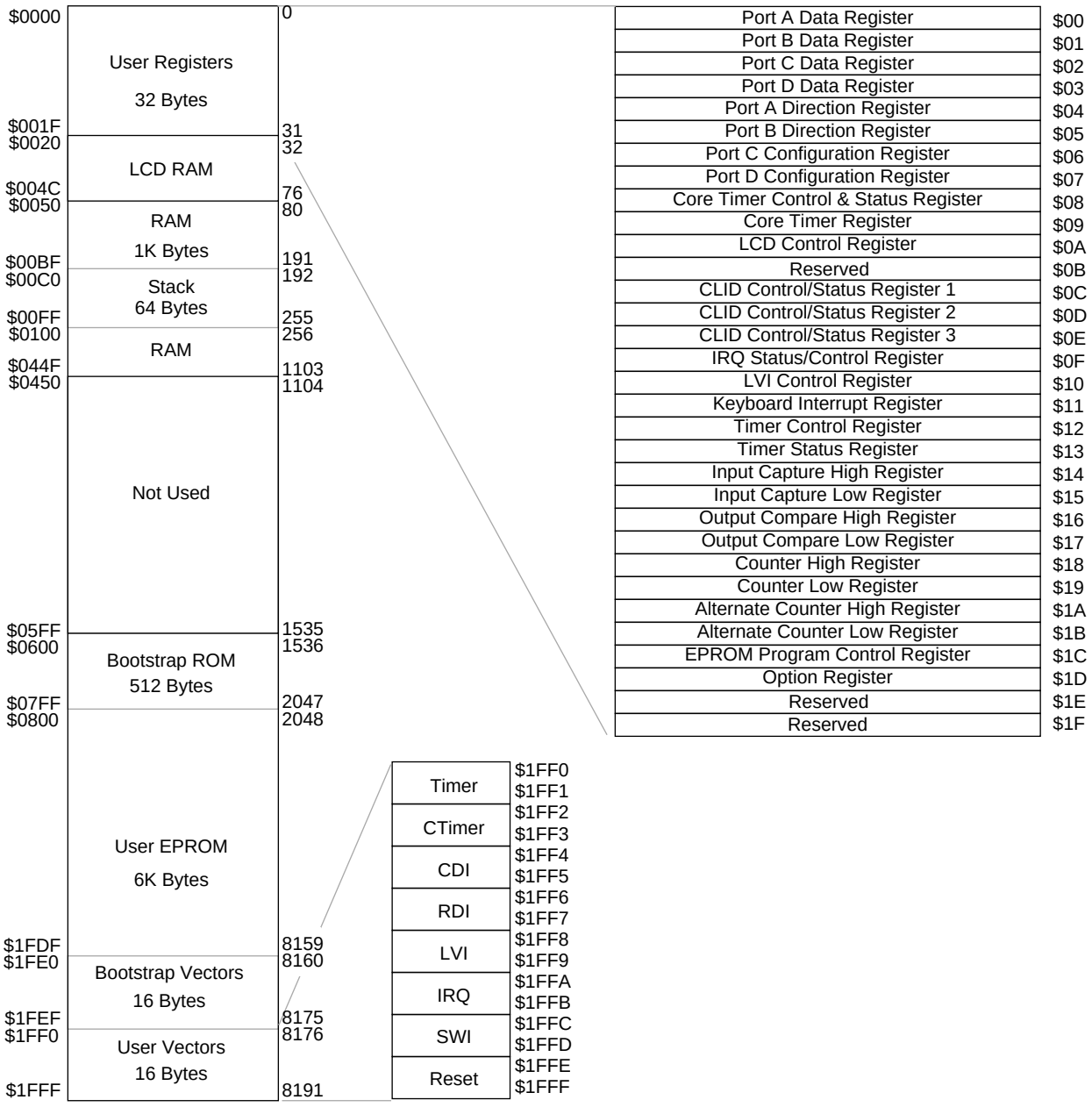


Figure A-1. MC68HC705CL4 Memory Map

A.3 EPROM

The user EPROM consists of 6112 bytes from $0800 to $1FDF and 16 bytes of user vectors from $1FF0 to $1FFF.

The bootstrap ROM is located from $0600 to $07FF.

The bootstrap vectors are located from $1FE0 to $1FEF.

A.4 BOOTSTRAP MODE

Bootstrap 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 PB5 pin is at logic one. The Bootstrap program is mask in the ROM area from $0600 to $07FF. This program handles copying of user code from an external EPROM into the on-chip EPROM.

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

A.5 EPROM PROGRAMMING

Programming the on-chip EPROM is achieved by using the Program Control Register located at address $1C.

Please contact Motorola for programming board availability.

A.5.1 Program Control Register (PCR)

This register is provided for programming the on-chip EPROM in the MC68HC705CL4

.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PCR	R							ELAT	PGM
$001C	W								
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.5.2 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.

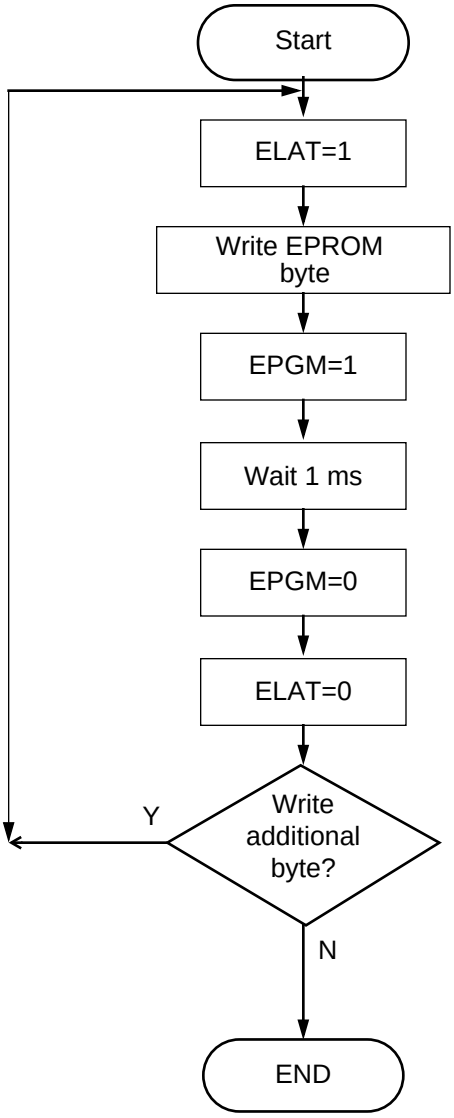


Figure A-2. EPROM Programming Sequence

A.6 MAXIMUM RATINGS

(Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Bootstrap Mode ($\overline{IRQ}$ /V_{PP} Pin Only)	V_{IN}	V_{SS}–0.3 to $2V_{DD}$+0.3	V

A.7 DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ±10%, VSS = 0 Vdc, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
EPROM Programming Voltage	V_{PP}		14.0		V
Supply Current (see Notes)					
Run	I_{DD}	—	7.5	9	mA
Wait		—	5	6	mA
Stop					
25˚C		—	22	50	µA
0˚C to +70˚C (Standard)		—	23	50	µA

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 $_{IDD}$: Measured using external square wave clock source to OSC1 (f_{OSC}=4.0 MHz), all inputs 0.2 VDC from rail; no DC 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 VDC, V_{IH} = V_{DD}–0.2 VDC.
6. Stop I_{DD} measured with OSC1=V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

A.8 CONTROL TIMING

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Units
EPROM Byte Programming Time	t_{EPROM}	—	10.0	ms

How to reach us:

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

INTERNET: http://Design-NET.com

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