HC05CT4GRS/D
REV. 2.0

68HC05CT4

General Release Specification

September 15, 1997

Motorola, Inc.
Austin, Texas

MOTOROLA

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.

List of Sections

List of Sections

Table of Contents

Section 1. General Description

Section 2. Memory

Section 3. Central Processing Unit

Section 4. Interrupts

Section 5. Resets

Section 6. Operating Modes

Section 7. Parallel Input/Output (I/O)

Section 8. 16-Bit Timer

Section 9. Synchronous Serial Interface (SSI)

Section 10. Core Timer

Section 11. Dual Phase-Locked Loop (PLL)

Section 12. Pulse Width Modulator (PWM)

Section 13. Comparators

Section 14. Miscellaneous Register

Section 15. Instruction Set

Section 16. Electrical Specifications

Section 17. Mechanical Specifications

Section 18. Ordering Information

List of Figures

Figure	Title	Page

List of Tables

List of Tables

Section 1. General Description

1.1 Contents

1.2 Introduction

The MC68HC05CT4 is a 44-pin member of the MC68HC05 Family of microcontrollers (MCUs), which supports cordless telephone applications. The memory map includes 5376 bytes of on-chip ROM and 256 bytes of RAM. The microcontroller unit has three 8-bit input/output (I/O) ports: A, B, and C; and one 7-bit I/O port: D. Port C has pullup options and keyscan capability. The MC68HC05CT4 includes a bird core, a bird timer, a synchronous serial interface (SSI), 16-bit timer, a

dual 60-MHz phase-locked loop (PLL), pulse width modulator (PWM), and an on-chip computer operating properly (COP) watchdog circuit.

Features of the MC68HC05CT4 include:

- Low Cost
- HC05 Core
- 44-Pin Plastic Leaded Chip Carrier (PLCC) Package
- 10.24-MHz On-Chip Crystal Oscillator
- 2.048-MHz Internal CPU Speed
- 5136 Bytes of User ROM
- 240 Bytes of Self-Check ROM
- 256 Bytes of On-Chip RAM
- 16-Bit Timer
- 31 Bidirectional I/O Lines
- Power-Saving Stop and Wait Modes
- Core Timer
- Dual 60-MHz PLL
- Simple Serial with Bidirectional Data (SSI)
- Keyscan Interrupt with Pullups on Port C
- Mask Selectable Options:
  - COP Watchdog Timer
  - Edge-Sensitive or Edge- and Level-Sensitive Interrupt Trigger
  - Port C Pullups for Keyscan
  - 1-Channel, 6-Bit Pulse Width Modulator (PWM)
- Three Comparators
- ROM Security Feature

1.3 Mask Options

The MC68HC05CT4 has eight mask options:

- Six port C pullups
- The IRQ sensitivity
- COP enable/disable

These are nonprogrammable options in that they are selected at the time of code submission when masks are made. These options are:

PC7PU — Port C7 Pullup

This bit enables or disables the pullup on port C bit 7.
1 = Enables the pullup
0 = Disables the pullup

NOTE: *Since port C bit 7 also is shared with the PWM output, PC7PU cannot be selected if PWM output is needed.*

PC6PU — Port C6 Pullup

This option enables or disables the pullup on port C bit 6.
1 = Enables pullup
0 = Disables pullup

PC5PU — Port C5 Pullup

This option enables or disables the pullup on port C bit 5.
1 = Enables pullup
0 = Disables pullup

PC4PU — Port C4 Pullup

This option enables or disables the pullup on port C bit 4.
1 = Enables pullup
0 = Disables pullup

PC23PU — Port C Bits 2 and 3 Pullups

When PC23PU = 1, the pullups on port C bits 2 and 3 are enabled simultaneously.

When PC23PU = 0, the pullups on port C bits 2 and 3 are disabled simultaneously.

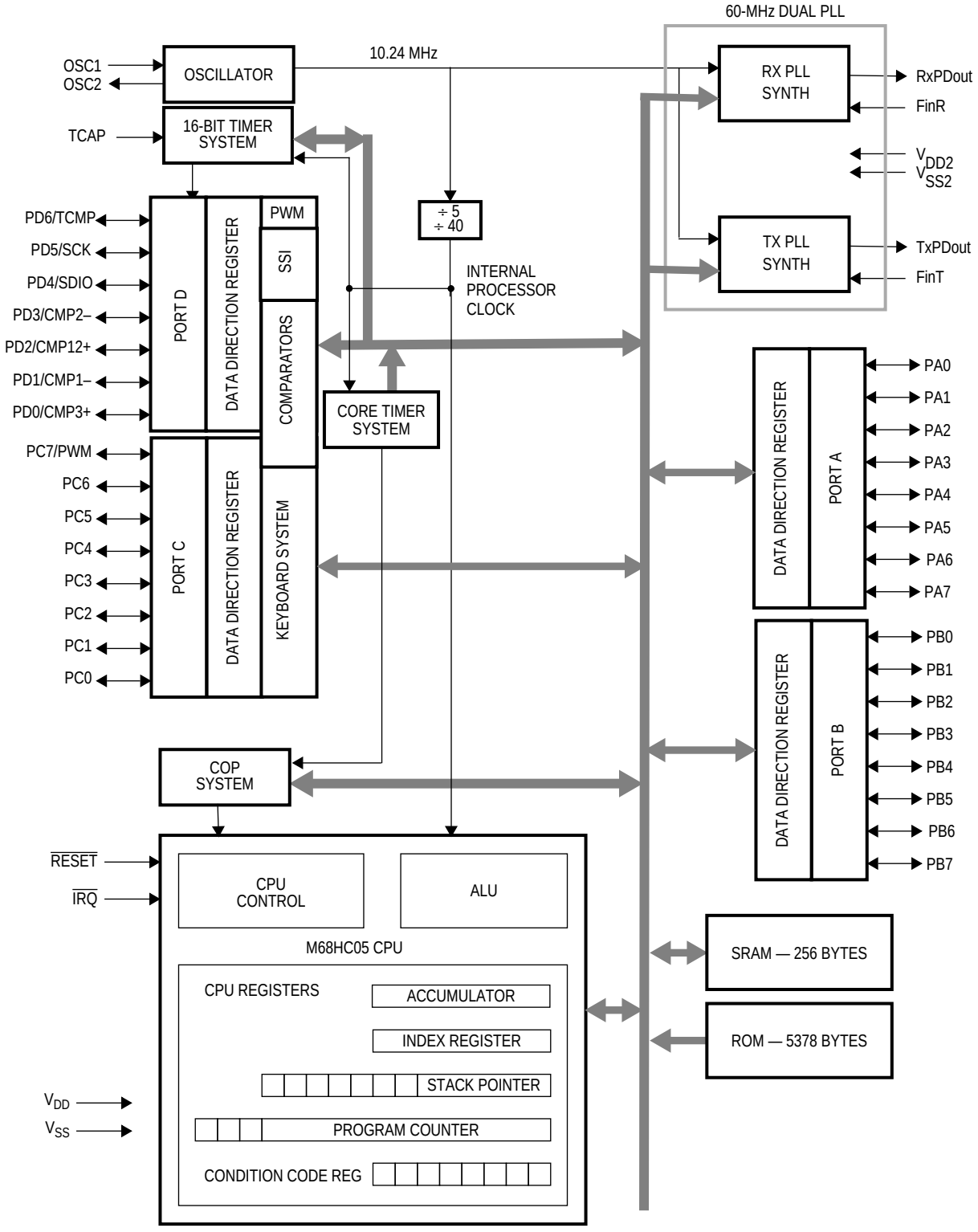


Figure 1-1. MC68HC05CT4 Block Diagram

PC01PU — Port C Bits 0 and 1 Pullups

When PC01PU = 1, the pullups on port C bits 0 and 1 are enabled simultaneously.

When PC01PU = 0, the pullups on port C bits 0 and 1 are disabled simultaneously.

COP — COP Enable

When the COP option is selected, the COP watchdog timer is enabled.

When the COP option is deselected, the COP watchdog timer is disabled.

IRQ — IRQ Sensitivity

When the IRQ option is selected (IRQEN = 1), edge- and level-sensitive IRQ is enabled.

When the IRQ option is deselected (IRQEN = 0), edge-only sensitive IRQ is enabled.

NOTE: *A line over a signal name indicates an active-low signal. For example, $\overline{RESET}$ is active low.*

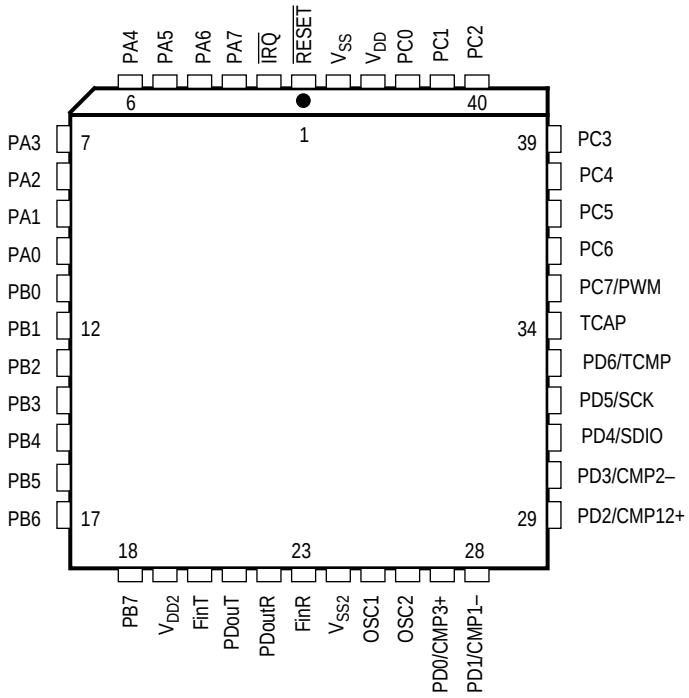


Figure 1-2. 44-Lead PLCC Pinout

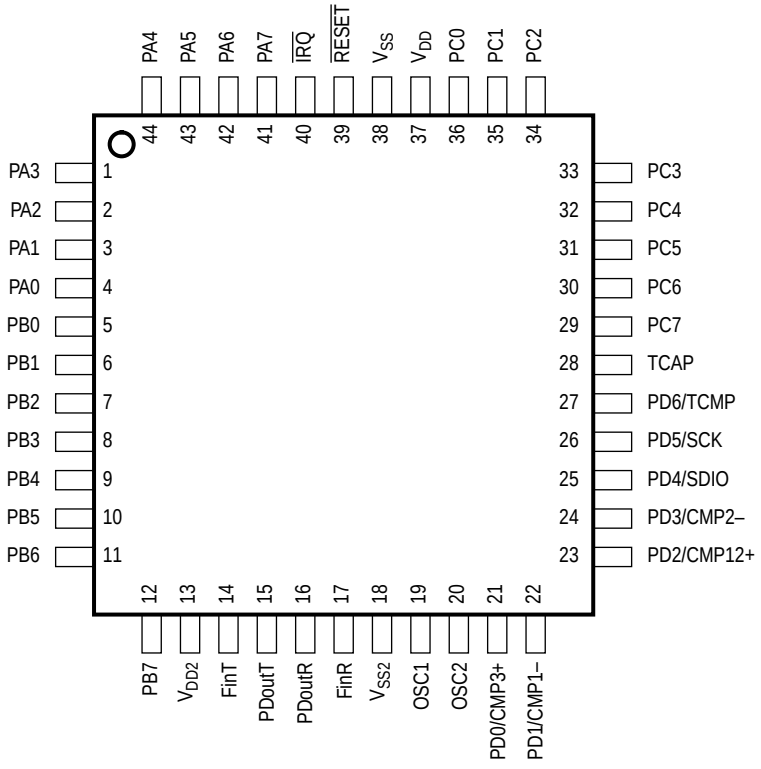


Figure 1-3. 44-Lead QFP Pinout

1.4 Signal Description

The following paragraphs describe the signals.

1.4.1 V_{DD} and V_{SS}

Power is supplied to the microcontroller's digital circuits using these two pins. V_{DD} is the positive supply and V_{SS} is ground.

1.4.2 V_{DD2} and V_{SS2}

Power is supplied to noise-susceptible circuitry such as the phase-locked loop (PLL), comparators, and oscillators that require "cleaner" supplies using these two pins. V_{DD2} is the positive supply and V_{SS2} is ground.

1.4.3 Maskable Interrupt Request ($\overline{IRQ}$)

This pin has a mask option as specified by the user that provides one of two different choices of interrupt triggering sensitivity. The options are:

1. Negative edge-sensitive triggering only
2. Both negative edge-sensitive and level-sensitive triggering

The microcontroller unit (MCU) completes the current instruction before it responds to the interrupt request. When $\overline{IRQ}$ goes low for at least one t_{ILIH}, a logic 1 is latched internally to signify that an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch contains a logic 1 and the interrupt mask bit (I bit) in the condition code register is clear, the MCU then begins the interrupt sequence.

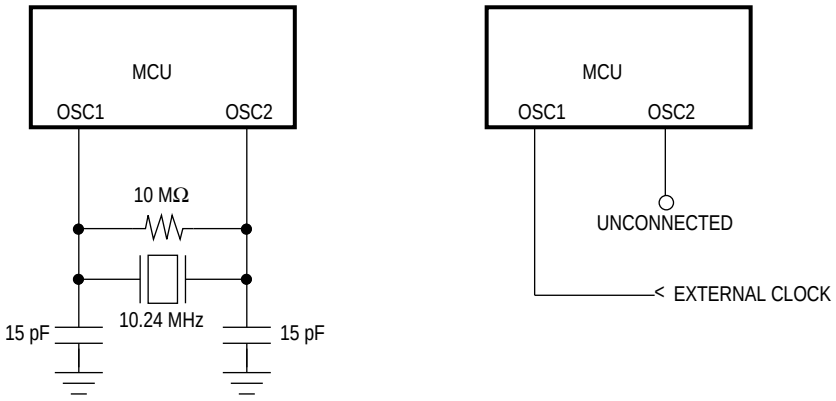
If the option is selected to include level-sensitive triggering, the $\overline{IRQ}$ input requires an external resistor to V_{DD} for wire-OR operation.

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

NOTE: *The voltage on the $\overline{IRQ}$ pin affects the mode of operation. See* ***Section 6. Operating Modes****.*

1.4.4 OSC1 and OSC2

These pins provide control input for an on-chip clock oscillator circuit that drives the PLL reference source. A crystal resonator, a ceramic resonator, or an external signal connects to these pins providing a system clock. The oscillator frequency is selectable between 5 or 40 times the internal bus rate. Typical oscillator configurations and component values are shown in **Figure 1-4**. The manufacturer of the crystal should be consulted, since actual component values are dependent on the type of crystal used.



(a) Crystal/Ceramic Resonator Oscillator Connections

(b) External Clock Source Connections

Figure 1-4. Oscillator Connections

1.4.5 Reset ($\overline{RESET}$)

This active-low pin is used to reset the MCU to a known startup state by pulling $\overline{RESET}$ low. The $\overline{RESET}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. See **Section 5. Resets**.

1.4.6 Port A (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 input during power-on or reset.

1.4.7 Port B (PB0–PB7)

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

1.4.8 Port C (PC0–PC7/PWM)

These eight I/O lines comprise port C. All port C lines are configured as input during power-on or reset. Port C has pullup devices and interrupt capability by pin; however, the state of any pin is determined by the user at the time of code submission. For a detailed description of I/O programming, refer to **7.7 Input/Output Port Pin Programming**. PC7 is shared with the output of the pulse-width modulation (PWM) function.

1.4.9 Port D (PD0/CMP3+, PD1/CMP1–, PD2/CMP12+, PD3/CMP2–, PD4/SDIO, PD5/SCK, and PD6/TCMP)

These seven port lines comprise port D. The state of any pin is software programmable and the lines are configured as input during power-on or reset. PD4 and PD5 are shared with the SSI subsystem, PD6 is shared with the 16-bit timer subsystem, and PD0–PD3 are shared with the comparators. For a detailed description on I/O programming, refer to **7.7 Input/Output Port Pin Programming**.

1.4.10 TCAP

This pin is used for the16-bit timer input capture operation. Depending on the value of the IEDG bit in the timer control register (TCR), the appropriate level of transition on TCAP will be monitored. When the

correct level of transition has occurred, the free-running counter will be transferred to the input capture register.

1.4.11 FinT and FinR

These pins are inputs to the PLL transmit and the receive counters, respectively. They typically are driven by the loop VCO and are also AC-coupled. The minimum input signal level is 200 mV peak to peak @ 60.0 MHz.

1.4.12 PDoutT and PDoutR

These PLL pins are 3-state outputs of the transmit and receive phase detectors, respectively, for use as either loop error signals or phase detector signals.

Section 2. Memory

2.1 Contents

2.2 Introduction

This section describes the organization of the on-chip memory. The MC68HC05CT4 8-Kbyte memory map is shown in **Figure 2-1** and the input/output (I/O) registers in **Figure 2-2**.

2.3 Memory Map

The MC68HC05CT4 has an 8-Kbyte memory map consisting of user read-only memory (ROM), random-access memory (RAM), self-check ROM, and I/O.

2.4 Read-Only Memory (ROM)

The user ROM consists of 5120 bytes of ROM located from $0B00 to $1EFF and 16 bytes of user vectors located from $1FF0 to $1FFF.

The self-check ROM is located from $1F00 to $1FEF.

Twelve of the user vectors, $1FF4–$1FFF, are dedicated to reset and interrupt vectors. The four remaining locations — $1FF0, $1FF1, $1FF2, and $1FF3 — are general-purpose user ROM locations.

2.5 ROM Security Feature

A security feature has been incorporated into the MC68HC05CT4 to help prevent external viewing of the ROM contents. This feature aids in maintaining the security of proprietary information in customer-developed software.[1]

2.6 Random-Access Memory (RAM)

The user RAM consists of 256 bytes of a shared stack area. The RAM starts at address $0030 and ends at address $012F. The stack begins at address $00FF. The stack pointer can access 64 bytes of RAM in the range $00FF to $00C0.

NOTE: *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.*

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

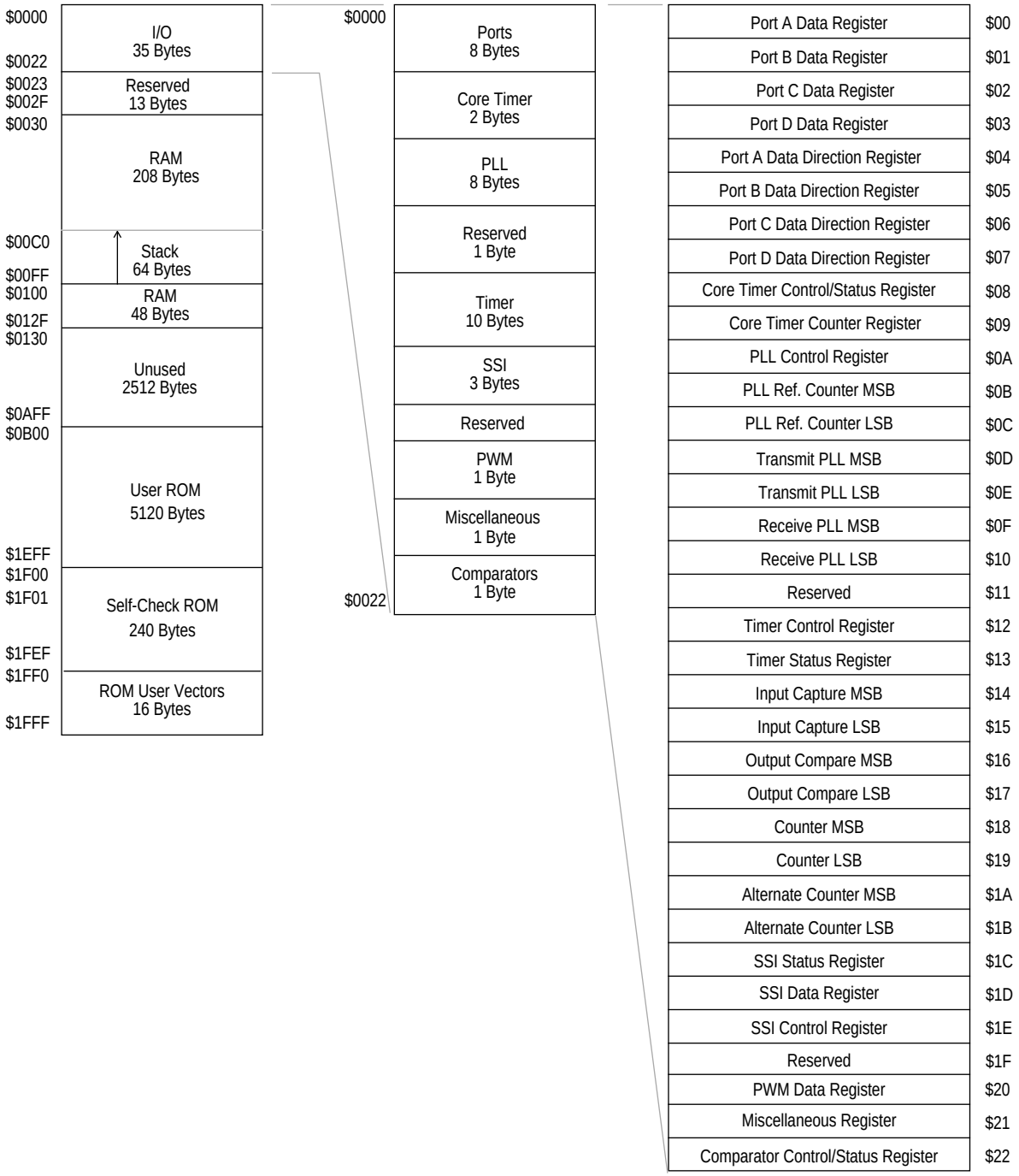


Figure 2-1. MC68HC05CT4 8-K Memory Map

Addr	Register Name	Bit 7	6	5	4	3	2	1	0
$00	Port A Data Register								
$01	Port B Data Register								
$02	Port C Data Register								
$03	Port D Data Register	0							
$04	Port A Data Direction Register								
$05	Port B Data Direction Register								
$06	Port C Data Direction Register								
$07	Port D Data Direction Register	0							
$08	Timer Control and Status Register	CTOF	RTIF	TOFE	RTIE	TOFC	RTFC	RT1	RT0
$09	Timer Counter Register								
$0A	PLL Control Register	0	TLOCK	RLOCK	REFON	TXON	RXON	PLS1	PLS0
$0B	PLL Reference Counter — MSB	0	0	0	0				
$0C	PLL Reference Counter — LSB								
$0D	PLL Transmit Counter — MSB								
$0E	PLL Transmit Counter — LSB								
$0F	PLL Receive Counter — MSB								
$10	PLL Receive Counter — LSB								
$11	Reserved	R	R	R	R	R	R	R	R
$12	Timer Control Register	ICIE	OCIE	TOIE	0	0	TON	IEDGE	OLVL
$13	Timer Status Register	ICF	OCF	0	0	0	0	0	0
$14	Timer Input Capture — MSB								
$15	Timer Input Capture — LSB								
$16	Timer Output Compare — MSB								
$17	TImer Output Compare — LSB								
$18	Timer Counter — MSB								
$19	Timer Counter — LSB								

R = Reserved

Figure 2-2. I/O Registers

Addr	Register Name	Bit 7	6	5	4	3	2	1	0
$1A	Timer Alternate Counter — MSB								
$1B	TImer Alternate Counter — LSB								
$1C	SSI Status Register	SF	DCOL	0	0	0	0	0	0
$1D	SSI Data Register								
$1E	SSI Control Register	SIE	SE	LSBF	MSTR	CPOL	T/R	SR1	SR0
$1F	Reserved	R	R	R	R	R	R	R	R
$20	PWM Data Register	0	0						
$21	Miscellaneous Register	0	0	0	0	SPEED	COE	PWME	0
$22	Comparator Control/Status Register	CMP3	CMP2	CMP1	CM3IE	CM3IC	CEN3	CEN2	CEN1

R = Reserved

Figure 2-2. I/O Registers (Continued)

Memory

Section 3. Central Processing Unit

3.1 Contents

3.2 Introduction

The MCU contains five registers as shown in **Figure 3-1**. The interrupt stacking order is shown in **Figure 3-2**.

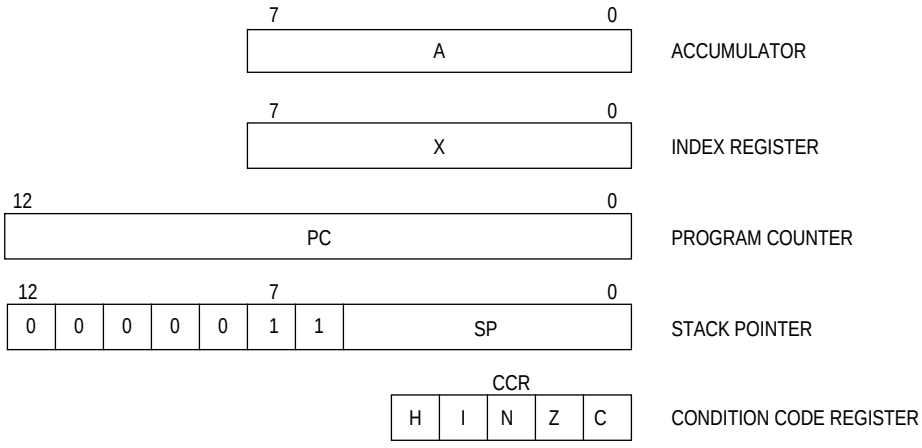
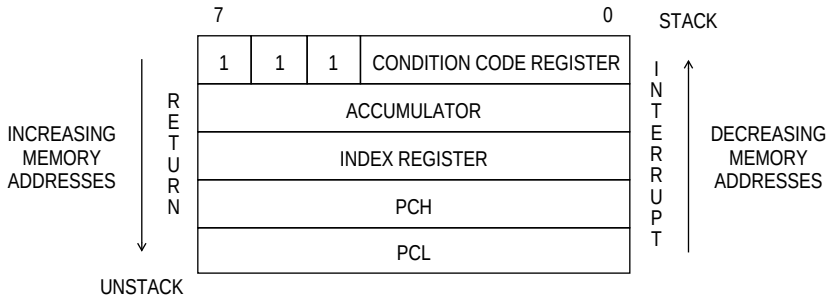


Figure 3-1. Programming Model

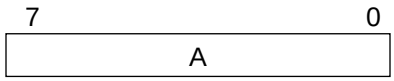


NOTE: Since the stack pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 3-2. Stacking Order

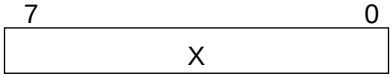
3.3 Accumulator

The accumulator (A) is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



3.4 Index Register

The index register (X) is an 8-bit register used for the indexed addressing value to create an effective address. The index register may also be used as a temporary storage area.



3.5 Condition Code Register

The condition code register (CCR) is a 5-bit register in which the H, N, Z, and C bits are used to indicate the results of the instruction just executed, and the I bit is used to enable interrupts. These bits can be tested individually by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

CCR

H	I	N	Z	C

Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer and external interrupt are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the I bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative.

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

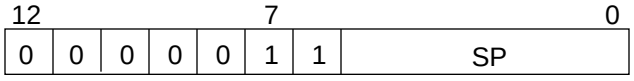
Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit also is affected during bit test and branch instructions and during shifts and rotates.

3.6 Stack Pointer

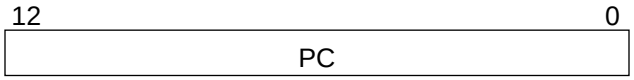
The stack pointer contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer then is decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits (MSB) are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.



3.7 Program Counter

The program counter (PC) is a 13-bit register that contains the address of the next byte to be fetched. .



NOTE: *The HC05 CPU core is capable of addressing 16-bit locations. For this implementation, however, the addressing registers are limited to an 8-Kbyte memory map*

Section 4. Interrupts

4.1 Contents

4.2 Introduction

The MCU can be interrupted seven different ways:

1. Nonmaskable software interrupt instruction (SWI)
2. External asynchronous interrupt ($\overline{\text{IRQ}}$)
3. External interrupt via IRQ on PC0–PC7
4. Internal 16-bit timer interrupt (TIMER)
5. Internal synchronous serial interface (SSI) interrupt
6. Internal core timer interrupt
7. Comparator 3

4.3 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, 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 after 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 $1FF4–$1FFF as defined in **Table 4-1**.

Table 4-1. Vector Address for Interrupts and Reset

Register	Flag Name	Interrupt	CPU Interrupt	Vector Address
N/A	N/A	Reset	RESET	$1FFE–$1FFF
N/A	N/A	Software	SWI	$1FFC–$1FFD
CMCSR	CMP3	CMP3, External Interrupts*	CMP3/IRQ	$1FFA–$1FFB
TSR	OCF, ICF, TOF	16-bit Timer Interrupts	TIMER	$1FF8–$1FF9
SSSR	SSIF	SSI Interrupt	SSI	$1FF6–$1FF7
CTCSR	TOFE, RTIE	Core Timer Interrupts	TIMER, RTI	$1FF4–$1FF5

* External interrupts include IRQ and PORTC sources.

The M68HC05 CPU does not support interruptible instructions. The maximum latency to the first instruction of the interrupt service routine must include the longest instruction execution time plus stacking overhead.

Latency = (Longest instruction execution time + 10) x t_{cyc} seconds

A return-to-interrupt (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 occurs during interrupt processing.

4.4 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 an internally generated RST 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 **4.3 CPU Interrupt Processing**.

4.5 Software Interrupt (SWI)

The SWI is an executable instruction and a nonmaskable 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 that 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.

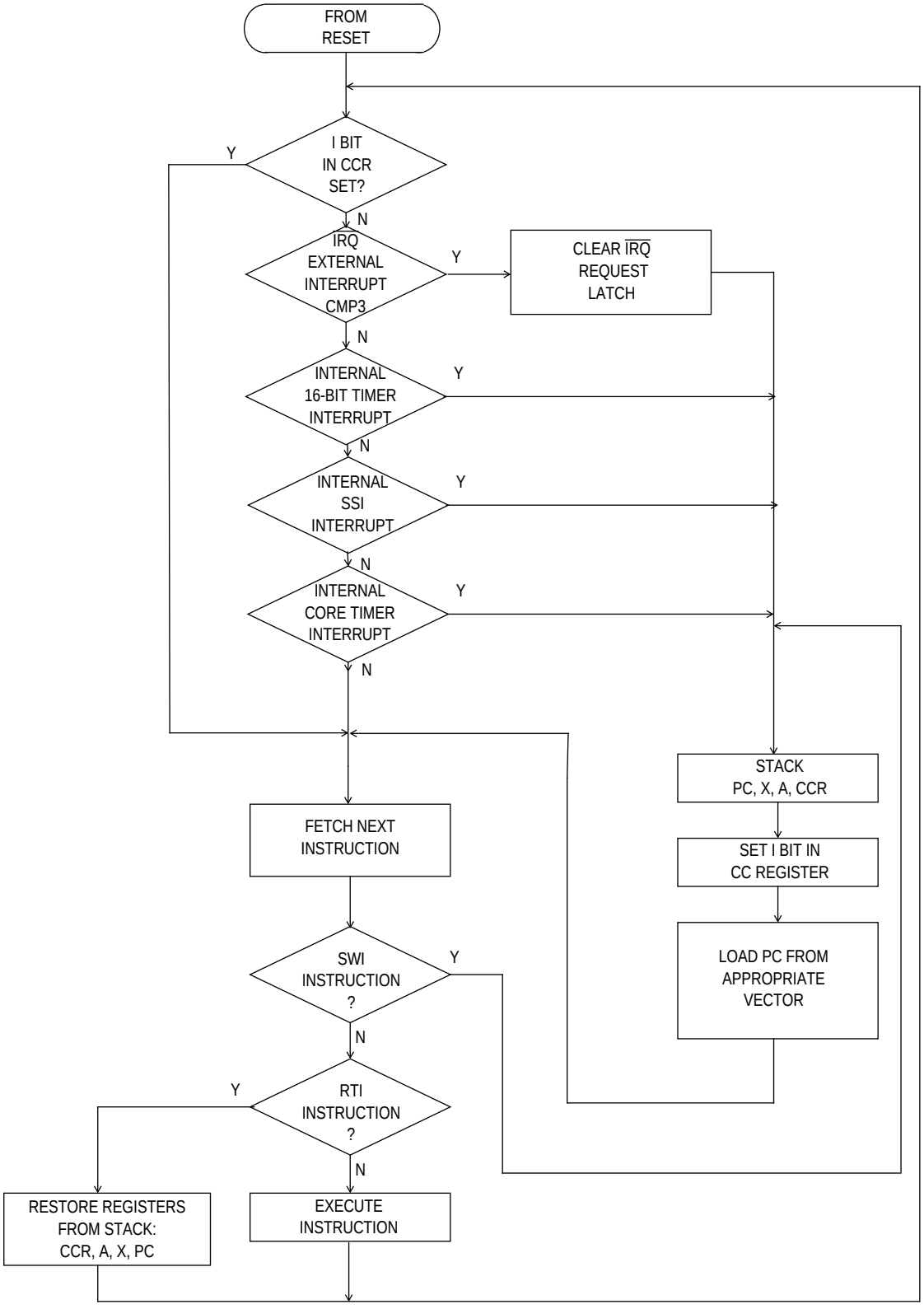


Figure 4-1. Interrupt Processing Flowchart

4.6 Hardware Interrupts

All hardware interrupts except $\overline{\text{RESET}}$ 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 two types of hardware interrupts are explained in the following sections.

4.7 External Interrupt (IRQ)

The $\overline{\text{IRQ}}$ pin provides an asynchronous interrupt to the CPU. A block diagram of the IRQ function is shown in **Figure 4-2**.

NOTE: *The BIH and BIL instructions will apply only to the level on the $\overline{\text{IRQ}}$ pin itself, and not to the output of the logic OR function with the port C IRQ interrupts. The state of the individual port C pins can be checked by reading the appropriate port C pins as inputs.*

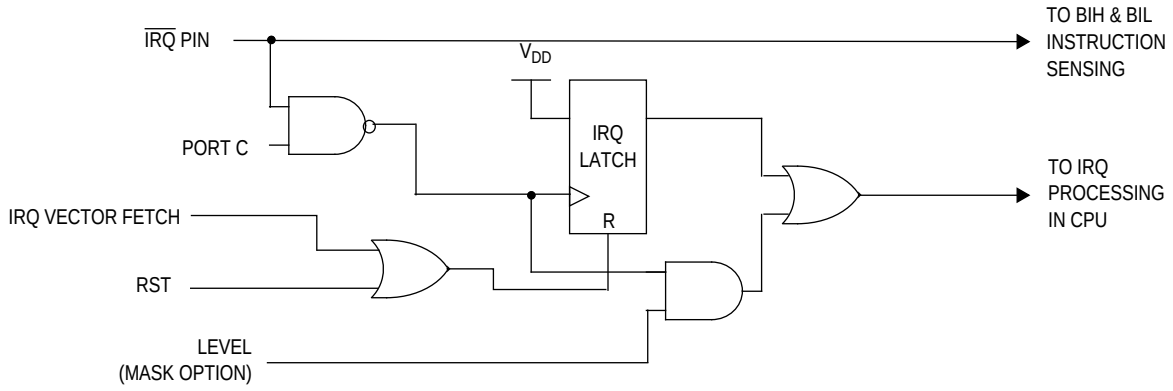


Figure 4-2. IRQ Function Block Diagram

The $\overline{\text{IRQ}}$ pin is one source of an external interrupt. All port C pins (PC0–PC7) act as other external interrupt sources if the keyscan feature is enabled as specified by the user.

When edge sensitivity is selected for the IRQ interrupt, it is sensitive to:

- Falling edge on the $\overline{\text{IRQ}}$ pin
- Falling edge on any port C pin with keyscan enabled

When edge and level sensitivity is selected for the IRQ interrupt, it is sensitive to the following cases:

- Low level on the $\overline{\text{IRQ}}$ pin
- Falling edge on the $\overline{\text{IRQ}}$ pin
- Falling edge or low level on any port C pin with keyscan enabled

4.8 External Interrupt Timing

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. The interrupt request is latched immediately following the falling edge of the $\overline{\text{IRQ}}$ source. The interrupt request is then synchronized internally and serviced as specified by the contents of $1FFA and $1FFB.

Either a level-sensitive and edge-sensitive trigger or an edge-sensitive-only trigger is available via the mask programmable option for the $\overline{\text{IRQ}}$ pin.

4.9 16-Bit Timer Interrupt

Three different timer interrupt flags cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the timer status register (TSR), and the enable bits are in the timer control register (TCR). Any of these interrupts vector to the same interrupt service routine, located at the address specified by the contents of memory locations $1FF8 and $1FF9.

4.10 SSI Interrupt

Two different synchronous serial interrupt (SSI) flags cause an SSI interrupt whenever they are set and enabled. The interrupt flags are in the SSI status register (SSSR), and the enable bits are in the SSI control register (SSCR). Either of these interrupts vector to the same interrupt service routine, located at the address specified by the contents of memory locations $1FF6 and $1FF7.

4.11 Core Timer Interrupt

This timer can create two types of interrupts. A timer overflow interrupt occurs whenever the 8-bit timer rolls over from $FF to $00 and the enable bit TOFE is set. A real-time interrupt occurs whenever the programmed time elapses and the enable bit RTIE is set. Either of these interrupts vector to the same interrupt service routine, located at the address specified by the contents of memory locations $1FF4 and $1FF5.

4.12 Comparator 3 Interrupt

Comparator 3 can create an interrupt when its output (CMP3) gets set and the enable bit CM3IE is set. The interrupt service routine is located at the address specified by the contents of memory locations $1FFA and $1FFB.

Interrupts

Section 5. Resets

5.1 Contents

5.2 Introduction

The MCU can be reset from four sources: one external input and three internal restart conditions. The $\overline{\text{RESET}}$ pin is an input with a Schmitt trigger as shown in **Figure 5-1**. All the internal peripheral modules will be reset by the internal reset signal (RST). Refer to **Figure 5-2** for reset timing detail.

5.3 External Reset ($\overline{\text{RESET}}$)

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

reset the CPU and peripherals. Termination of the external RESET input or the internal COP watchdog reset are the only reset sources that 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.*

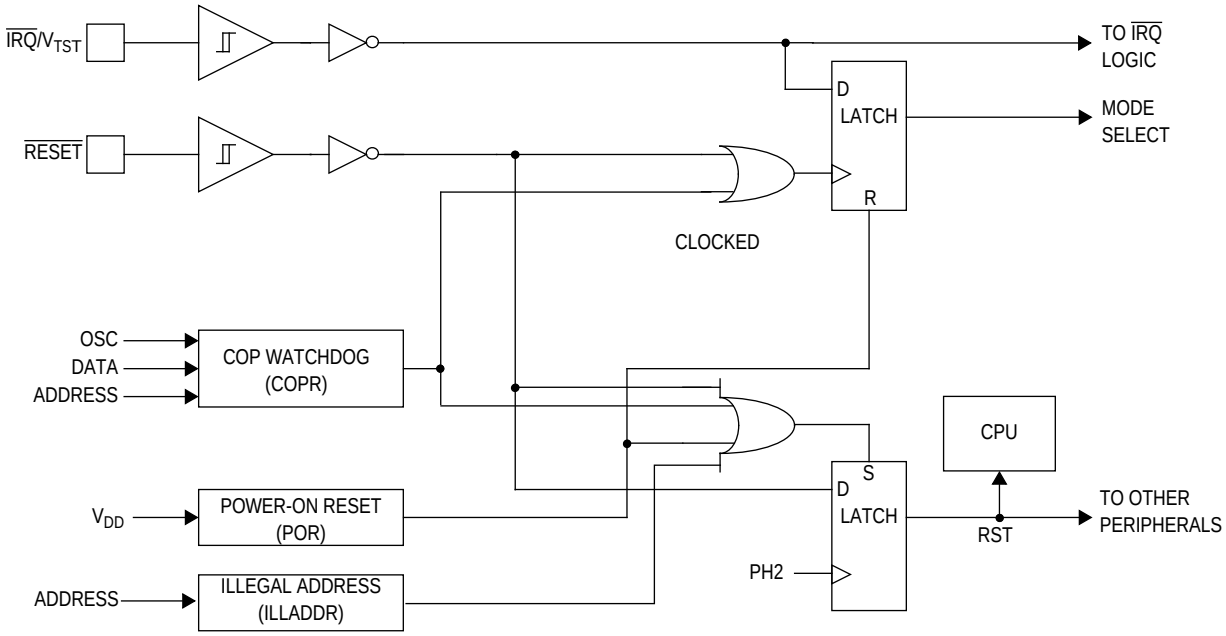


Figure 5-1. Reset Block Diagram

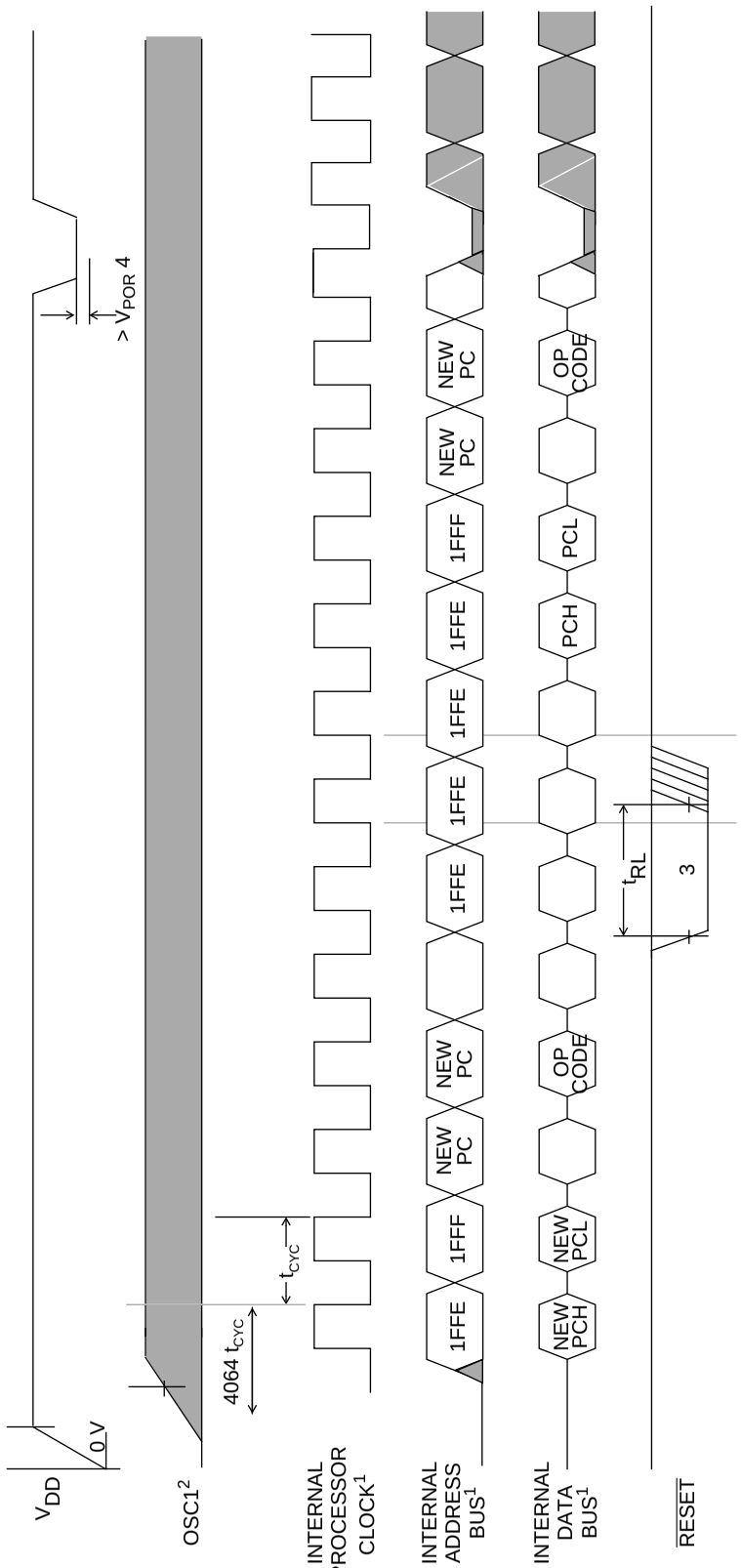


Figure 5-2. RESET and POR Timing Diagram

NOTES:
1. Internal timing signal and bus information not available externally
2. OSC1 line is not meant to represent frequency. It is only used to represent time.
3. The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.
4. V_{DD} must fall to a level lower than V_{POR} to be recognized as a power-on reset.

5.4 Internal Resets

The three internally generated resets are the initial power-on reset function, the COP watchdog timer reset, and the illegal address detector. Termination of the external $\overline{\text{RESET}}$ input or the internal COP watchdog timer are the only reset sources that can alter the operating mode of the MCU. The other internal resets do not have any effect on the mode of operation when their reset state ends.

5.4.1 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 internal processor bus clock cycles (PH2) after the oscillator becomes active.

The POR generates the RST signal that resets the CPU. If any other reset function is active at the end of this 4064-cycle delay, the RST signal remains in the reset condition until the other reset condition(s) end. During the POR, the $\overline{\text{RESET}}$ pin is forced low.

5.4.2 Computer Operating Properly Reset (COPR)

The MCU contains a watchdog timer that automatically times out if not reset (cleared) within a specific time by a program reset sequence. If the COP watchdog timer is allowed to time out, an internal reset is generated to reset the MCU. Regardless of an internal or external RESET, the MCU comes out of a COP reset according to the standard rules of mode selection.

The COP reset function is enabled or disabled by a mask option and is verified during production testing.

5.4.2.1 Resetting the COP

Writing a zero to the COPF bit prevents a COP reset. This action resets the counter and begins the timeout period again. The COPF bit is bit 0

of address $1FF0. A read of address $1FF0 returns user data programmed at that location.

5.4.2.2 COP During Wait Mode

The COP continues to operate normally during wait mode. The software should pull the device out of wait mode periodically and reset the COP by writing to the COPF bit to prevent a COP reset.

5.4.2.3 COP During Stop Mode

When the stop enable mask option is selected, stop mode disables the oscillator circuit and thereby turns the clock off for the entire device. When STOP is executed, the COP counter will hold its current state. If a reset is used to exit stop mode, the COP counter is reset and held until 4064 POR cycles are completed, at that time, counting will begin. If an external IRQ is used to exit stop mode, the COP counter does not wait for the completion of the 4064 POR cycles but it does count these cycles. It is recommended, therefore, that the COP is fed before executing the STOP instruction.

5.4.2.4 COP Watchdog Timer Considerations

The COP watchdog timer is active in all modes of operation if enabled by a mask option. If the COP watchdog timer is selected by a mask option, any execution of the STOP instruction (either intentional or inadvertent due to the CPU being disturbed) causes the oscillator to halt and prevent the COP watchdog timer from timing out. If the COP watchdog timer is selected by a mask option, the COP resets the MCU when it times out. Therefore, it is recommended that the COP watchdog is **disabled** for a system that must have intentional uses of the wait mode for periods longer than the COP timeout period.

The recommended interactions and considerations for the COP watchdog timer, STOP instruction, and WAIT instruction are summarized in **Table 5-1**.

Table 5-1. COP Watchdog Timer Recommendations

IF the following conditions exist:	THEN the COP Watchdog Timer should be:
WAIT Time	
WAIT Time less than COP Timeout	Enable or Disable COP by Mask Option
WAIT Time **More than** COP Timeout	Disable COP by Mask Option
Any length WAIT Time	Disable COP by Mask Option

5.4.2.5 COP Register

The COP register is shared with the MSB of an unimplemented user interrupt vector as shown in **Figure 5-3**. Reading this location returns whatever user data has been programmed at this location. Writing a zero to the COPR bit in this location clears the COP watchdog timer.

Addr	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$1FF0	Unimplemented Vector and COP Watchdog Timer	Read:	X	X	X	X	X	X	X	X
		Write:								COPR

= Unimplemented

Figure 5-3. COP Watchdog Timer Location

5.4.3 Illegal Address

An illegal address reset is generated when the CPU attempts to fetch an instruction from either unimplemented address space ($0130 to $0AFF) or I/O address space ($0000 to $002F).

Section 6. Operating Modes

6.1 Contents

6.2 Introduction

The MCU has two modes of operation: single-chip mode and self-check mode with two oscillator options. **Table 6-1** shows the conditions required to go into each mode.

Table 6-1. Operating Mode Conditions

RESET	IRQ	PB1	PD5	MODE
$V_{TST} = 2 \times V_{DD}$	V_{SS}–V_{DD}	V_{SS}–V_{DD}	V_{SS}–V_{DD}	Single-Chip
	V_{TST}	V_{DD}	V_{SS}	Self-Check

6.3 Single-Chip Mode

In single-chip mode, the address and data buses are not available externally, but there are three 8-bit input/output (I/O) ports and one 7-bit I/O port. This mode allows the MCU to function as a self-contained microcontroller, with maximum use of the pins for on-chip peripheral functions. All address and data activity occurs within the MCU. Single-chip mode is entered on the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}$ pin is within normal operating range.

6.4 Low-Power Modes

The following paragraphs describe the low-power modes.

6.4.1 Stop Mode

The STOP instruction places the MCU in its lowest power-consumption mode. In stop mode, the internal oscillator is turned off, halting all internal processing, including timer operation.

During stop mode, the CTCSR ($08) bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of stop mode only by an external interrupt or reset.

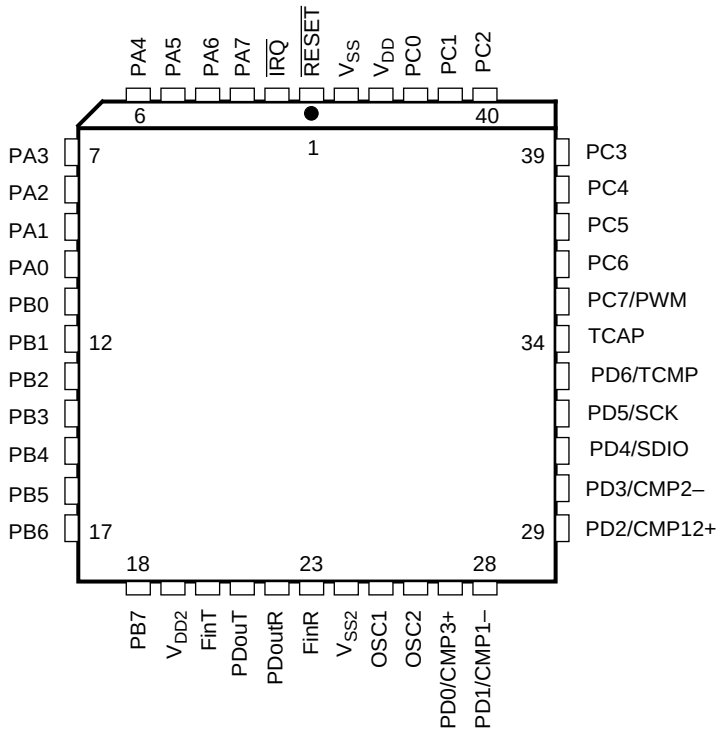


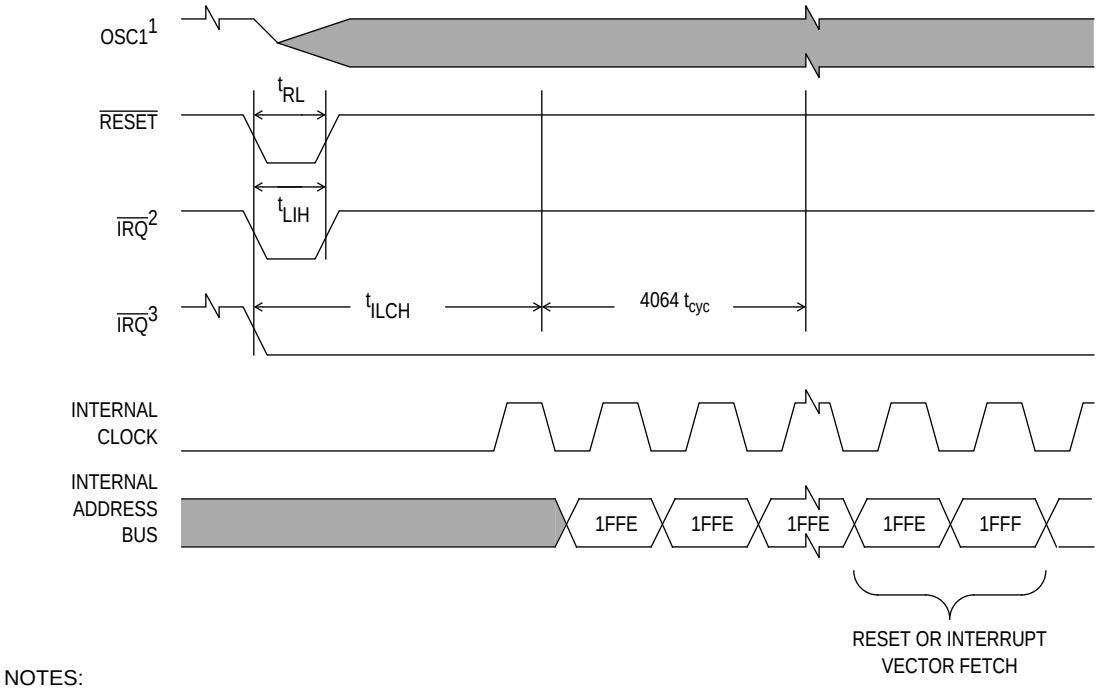
Figure 6-1. Single-Chip Mode Pinout of the MC68HC05CT4

6.4.2 Stop Recovery

The processor can be brought out of the stop mode only by an external interrupt or $\overline{\text{RESET}}$. See **Figure 6-2**.

6.4.3 Wait Mode

The WAIT instruction places the MCU in a low power-consumption mode, but the wait mode consumes more power than the stop mode. All CPU action is suspended, but the core timer, the oscillator, and any enabled module remain active. Any interrupt or reset will cause the MCU to exit the wait mode. The user must shut off subsystems to reduce power consumption. WAIT current specifications assume CPU operation only and do not include current consumption by any other subsystems.



NOTES:
1. Represents the internal gating of the OSC1 pin
2. $\overline{IRQ}$ pin edge-sensitive mask option
3. $\overline{IRQ}$ pin level- and edge-sensitive mask option

Figure 6-2. Stop Recovery Timing Diagram

During wait mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from wait mode.

6.4.4 Low-Power Wait

When the wait mode is entered by executing the WAIT instruction, the oscillator divider changes from a divide-by-5 to a divide-by-40 (additional divide-by-8) to lower the wait current. As a result, this gives a CPU clock rate of 256 kHz if the oscillator is running with a 10.24-MHz crystal. The oscillator divide-by-5 or divide-by-40 option is also controlled by the speed bit located in the miscellaneous control register ($21). **Section 14. Miscellaneous Register**. When returning from wait mode via an interrupt, the OSC rate prior to entering wait mode is restored.

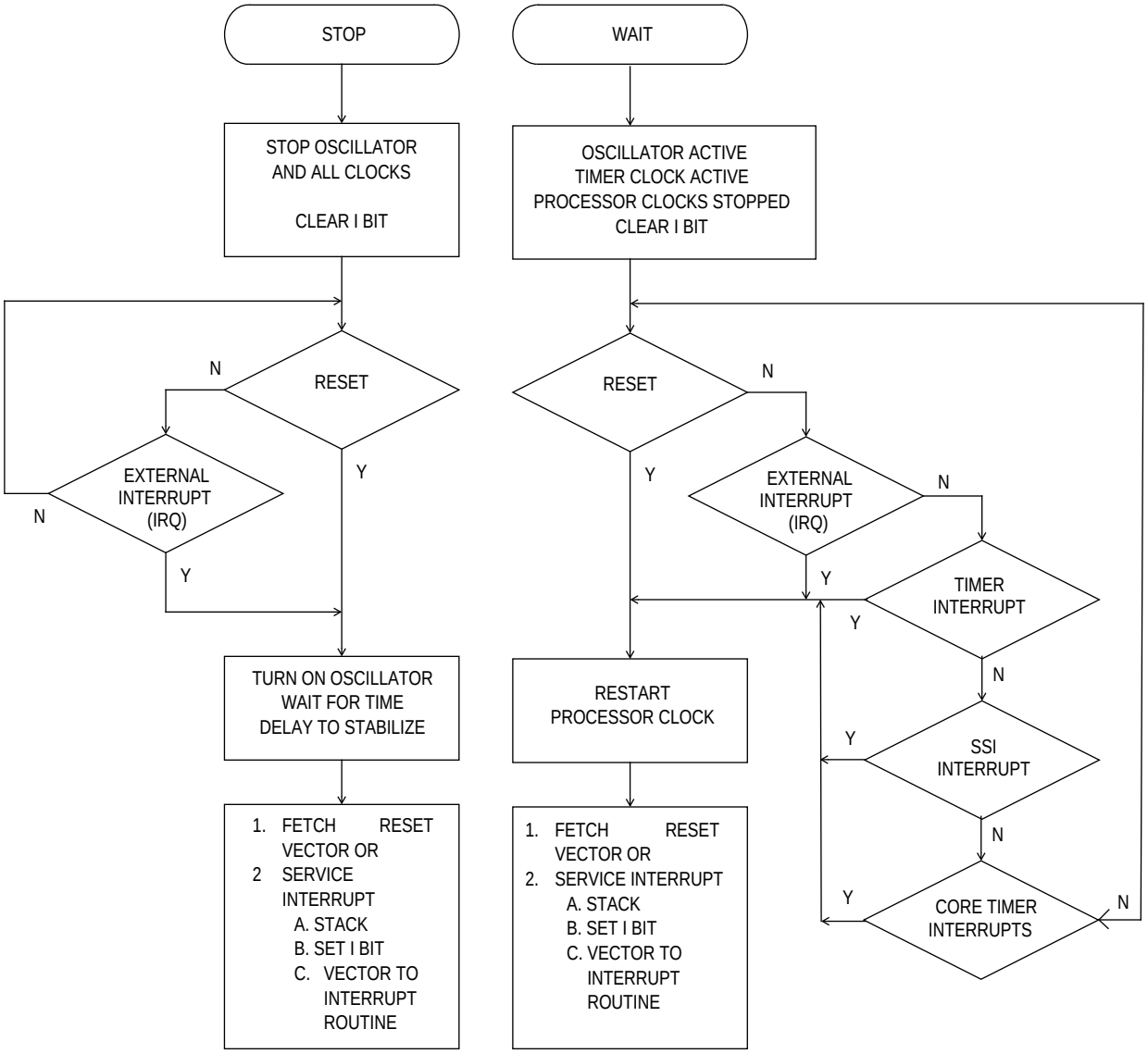


Figure 6-3. STOP/WAIT Flowchart

Operating Modes

Section 7. Parallel Input/Output (I/O)

7.1 Contents

7.2 Introduction

In user mode 31 lines are arranged as one 7-bit and three 8-bit ports. Most of these port pins are programmable as either inputs or outputs under software control of the data direction registers, though some are input only.

NOTE: *To avoid a glitch on the output pins, write data to the I/O port data register before writing a logic 1 to the corresponding data direction register.*

7.3 Port A

Port A is an 8-bit bidirectional port that does not share any of its pins with other subsystems. The port A data register is at $0000 and the data direction register (DDR) is at $0004. Reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a logic 1 to a DDR bit sets the corresponding port bit to output mode.

7.4 Port B

Port B is an 8-bit bidirectional port. The port B data register is at $0001 and the data direction register (DDR) is at $0005. Reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a logic 1 to a DDR bit sets the corresponding port bit to output mode.

7.5 Port C

Port C is an 8-bit bidirectional port. The port C data register is at $0002 and the data direction register (DDR) is at $0006. Reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a logic 1 to a DDR bit sets the corresponding port pin to output mode. Each of the port C pins has an optional pullup device. When the DDR bit is cleared and the pullup device is enabled, the pin will be a pullup *and* an interrupt pin. The edge- or edge- and level-sensitivity of the $\overline{IRQ}$ pin also pertains to the enabled port C pins. Care needs to be taken when using port C pins that have the pullup enabled. Before switching from an output to an input, the data should be preconditioned to a logic 1 to prevent an interrupt from occurring. Port C bit 7 is also shared with the PWM output. When PC7 is used as the PWM output, its pullup option should not be selected (see **Figure 7-1**).

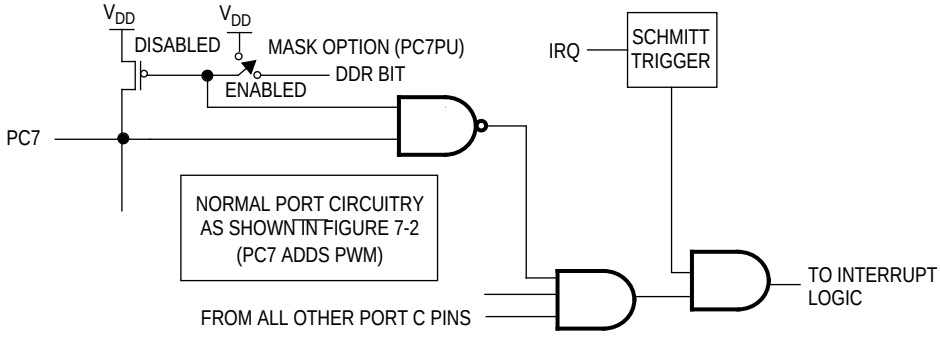


Figure 7-1. Port C Pullup Option

7.6 Port D

Port D is a 7-bit bidirectional port. Two of its pins are shared with the SSI subsystem, four are shared with the comparators, and one is shared with the timer. During reset, all seven bits become valid input ports because all special function output drivers associated with the timer and SSI subsystems are disabled.

7.7 Input/Output Port Pin Programming

Port pins may be programmed as inputs or outputs under software control. The direction of the pins is determined by the state of the corresponding bit in the port data direction register (DDR). Each I/O port has an associated DDR. Any I/O port pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic 0.

At power-on or reset, all DDRs are cleared, which configures all pins as inputs. The data direction registers are capable of being written to or read by the processor. During the programmed output state, a read of the data register actually reads the value of the output data latch and not the I/O pin.

Refer to **Table 7-1** and to **Figure 7-2** for additional information.

Table 7-1. I/O PIn Functions

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

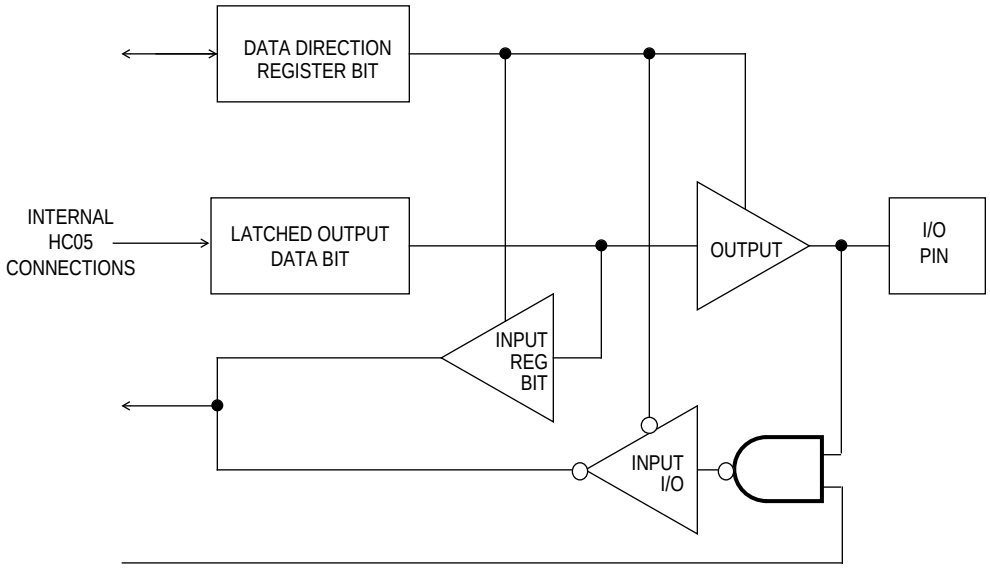


Figure 7-2. I/O Circuitry

Section 8. 16-Bit Timer

8.1 Contents

8.2 Introduction

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.

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. Access to 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 registers of a specific timer function to ensure that an interrupt does not occur.*

8.3 Counter Register

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 2.0 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). If a read of the free-running counter or counter alternate register is from the least significant byte (LSB) ($19, $1B), the LSB 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 also must 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). The counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

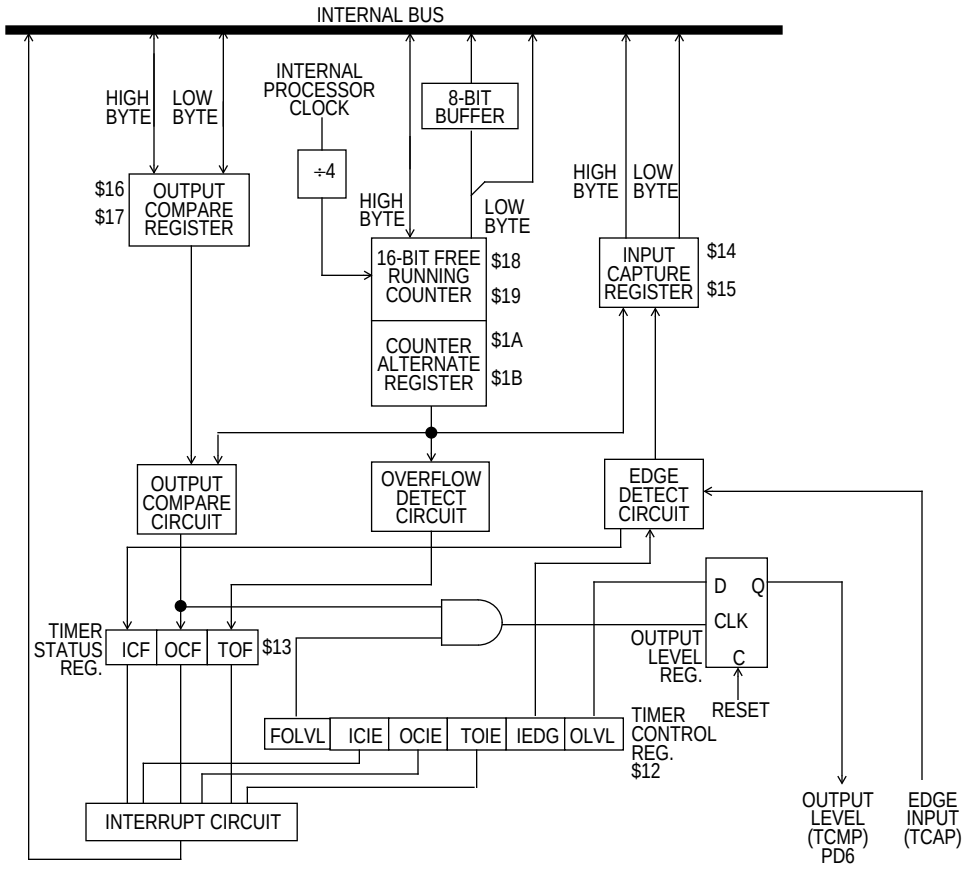


Figure 8-1. 16-Bit Timer Block Diagram

The free-running counter is configured to $FFFC during reset and is a read-only register but only when the timer is enabled. During a power-on reset, the counter is also preset to $FFFC and begins running only after the TON bit in the TIMER control register is set. Because the free-running counter is 16 bits preceded by a fixed divided-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).

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

8.4 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 continually compared with the contents of the free-running counter. 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.

8.5 Input Capture Register

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 that 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 is 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 MSB ($14), 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.

8.6 Timer Control Register

The TCR is a read/write register containing six control bits. Three bits control interrupts associated with the timer status register flags ICF, OCF, and TOF.

Address: $0012

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	ICIE	OCIE	TOIE	0	0	TON	IEDG	OLVL
Reset:	0	0	0	0	0	0	0	0

Figure 8-2. Timer Control Register (TCR)

ICIE — Input Capture Interrupt Enable

1 = Interrupt enabled
0 = Interrupt disabled

OCIE — Output Compare Interrupt Enable

1 = Interrupt enabled
0 = Interrupt disabled

TOIE — Timer Overflow Interrupt Enable

1 = Interrupt enabled
0 = Interrupt disabled

TON — Timer On

When disabled, the timer is initialized to the reset condition.

1 = Timer enabled
0 = Timer disabled

IEDG — Input Edge

Value of input edge determines which level transition on the TCAP pin will trigger a free-running counter transfer to the input capture register. Reset clears this bit.

1 = Positive edge
0 = Negative edge

OLVL — Output Level

Value of output level is clocked into the output level register by the next successful output compare and will appear on the TCMP pin.

1 = High output
0 = Low output

8.7 Timer Status Register

The TSR is a read-only register containing three status flag bits.

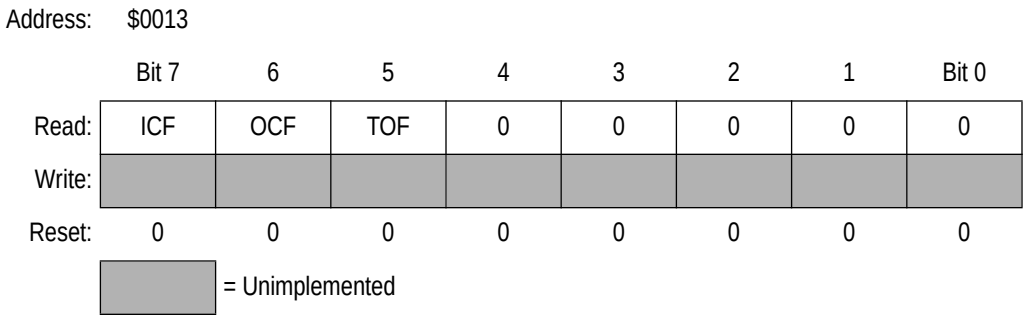


Figure 8-3. Timer Status Register (TSR)

ICF — Input Capture Flag

1 = Flag set when selected polarity edge is sensed by input capture edge detector
0 = Flag cleared when TSR and input capture low register ($15) are accessed

Reset clears this bit.

OCF — Output Compare Flag

1 = Flag set when output compare register contents match the free-running counter contents
0 = Flag cleared when TSR and output compare low register ($17) are accessed

Reset clears this bit.

TOF — Timer Overflow Flag

1 = Flag set when free-running counter transitions from $FFFF to $0000 occurs

0 = Flag cleared when TSR and counter low register ($19) are accessed

Reset clears this bit.

Bits 0–4 — Not Used

Always read zero.

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:

- The timer status register is read or written when TOF is set.
- The MSB 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; this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

8.8 Timer During Wait Mode

The CPU clock halts during wait mode, but the timer remains active if turned on prior to entering wait mode. If interrupts are enabled, a timer interrupt will cause the processor to exit wait mode.

8.9 Timer During Stop Mode

In stop mode, the timer stops counting and holds the last count value if stop is exited by an interrupt. If reset is used, the counter is forced to $FFFC. During stop, if the timer is on and at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags or wake up the MCU. When the MCU does wake up, however, there is an active input capture flag and data from the first valid edge that occurred during the stop mode. If reset is used to exit stop mode, no input capture flag or data remains, even if a valid input capture edge occurred.

8.10 Timer Power Supply Source

The timer's power is supplied by V_{DD} and V_{SS}. V_{DD2} and V_{SS2} will not be needed since this module is not susceptible to supply noise.

16-Bit Timer

Section 9. Synchronous Serial Interface (SSI)

9.1 Contents

9.2 Introduction

The synchronous serial interface (SSI) is a 2-wire master/slave system including serial clock (SCK) and serial data input/output (SDIO). Data is transferred eight bits at a time. A software programmable option determines whether the SSI transfers data most significant bit (MSB) or least significant bit (LSB) first and an interrupt may be generated at the completion of each transfer. When operating as a master device, the serial clock speed is selectable from a choice of four rates.

Transmission in master mode is initiated by a write to the SSI data register (SDR). A transfer cannot be initiated in slave mode; the external master initiates the transfer. The programmer must choose between master or slave mode before the SSI is enabled. The programmer must ensure that only one master exists in the system at any one time. All devices in the system must operate with the same clock polarity and data rates. Slaves should always be disabled before the master is disabled.

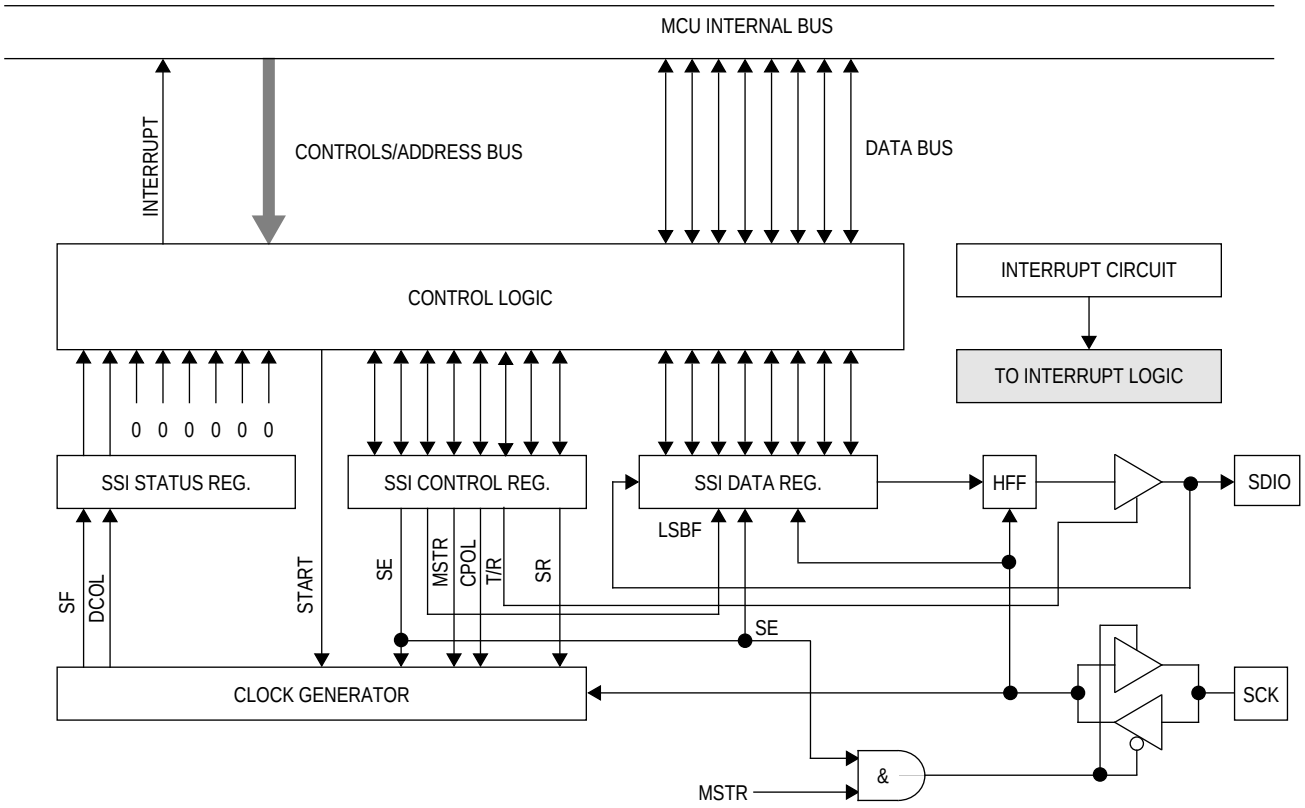


Figure 9-1. SSI Block Diagram

9.3 Signal Format

The SSI is comprised of two main input/output (I/O) signals that interface with port D serial clock and serial data.

9.3.1 Serial Clock (SCK)

When SE = 0, this pin is a port D bit 5 pin, which follows the port D DDR assignment.

In master mode (MSTR = 1), the serial clock (SCK) pin is an output with four selectable frequencies. This pin will be high (CPOL = 1) or low (CPOL = 0) between transmissions.

In slave mode (MSTR = 0), the SCK pin is an input and the clock must be supplied by an external master with a maximum frequency of $f_{OP}/2$. There is no minimum SCK frequency. This pin should be driven high (CPOL = 1) or low (CPOL = 0) between transmissions by the external master and must be stable before the SSI is first enabled (SE = 1).

Data is always captured at the serial data in/out (SDIO) pin on the rising edge of SCK.

Data is always shifted out and presented at the serial data in/out (SDIO) pin on the falling edge of SCK.

9.3.2 Serial Data In/Out (SDIO)

Prior to enabling the SSI (SE = 0), the serial data in/out (SDIO) pin is a port D bit 4 pin, which follows the port D DDR assignment. When the SSI is enabled (SE = 1) the SDIO pin becomes a high-impedance input pin if the T/$\overline{R}$ bit is low or it idles high if the T/$\overline{R}$ bit is high.

The data can be sent or received in either MSB first format (LSBF = 0) or LSB first format (LSBF = 1).

If (CPOL = 1), the first falling edge of SCK will shift the first data bit out to the SDIO pin. Subsequent falling edges of SCK will shift the remaining data bits out.

If (CPOL = 0), the first data bit will be driven out to the SDIO pin before the first rising edge of SCK. Subsequent falling edges of SCK will shift the remaining data bits out.

When receiving data in master mode, the T/$\overline{R}$ bit must be low and data must be written to the data register to initiate clock generation.

When transmitting data in master mode, the T/$\overline{R}$ bit must be high.

When receiving data in slave mode, T/$\overline{R}$ bit must be low and the clock and data must be supplied by external device.

When transmitting data in slave mode, T/$\overline{R}$ bit must be high, and data must be written to the data register before the SSI is enabled to ensure that proper data is transferred.

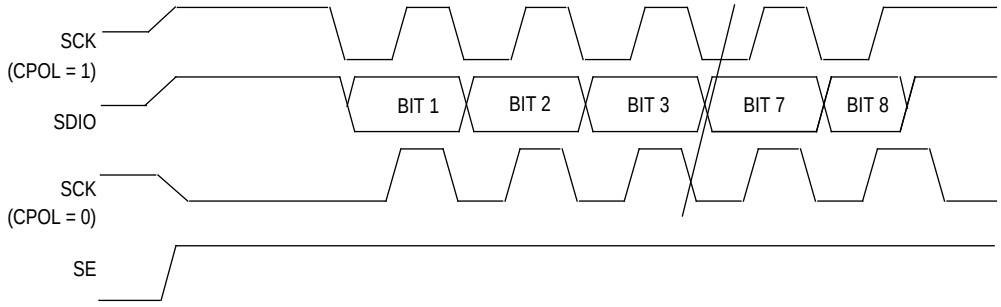


Figure 9-2. Serial I/O Port Timing

9.4 SSI Registers

The SSI has three registers: control, status, and data.

9.4.1 SSI Control Register

This register is located at address $001E and contains seven bits. A reset clears all of these bits, except bit 3 which is set. Writes to this register during a transfer should be avoided, with the exception of clearing the SE bit to disable the SSI.

In addition, the clock polarity, rate, data format and master/slave selection should not be changed while the SSI is enabled (SE = 1) or

being enabled. Always disable the SSI first, by clearing the SE bit, before altering these control bits within the SSI control register (SCR).

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SIE	SE	LSBF	MSTR	CPOL	T/$\overline{R}$	SR1	SR0
Reset:	0	0	0	0	1	0	0	0

Figure 9-3. SSI Control Register (SCR)

SIE — SSI Interrupt Enable

This bit determines whether an interrupt request should be generated when a transfer is complete.

When set, an interrupt request is made if the CPU is in the run or wait mode of operation and status flag bit SF is set.

When cleared, no interrupt requests are made by the SSI.

SE — SSI Enable

When set, this bit enables the SSI, makes PD5 the SCK pin, and makes PD4 the SDIO pin.

When SE is cleared, any transmission in progress is aborted, the bit counter is reset, and pins SCK and SDIO revert to being PD5 and PD4.

LSBF — Least Significant Bit (LSB)First

When set, data is sent and received in a least significant bit (LSB) first format.

When cleared, data is sent and received in a most significant bit (MSB) first format.

MSTR — Master Mode

When set, this bit configures the SSI to the master mode. This means that the transmission is initiated by a write to the data register and the SCK pin becomes an output providing a synchronous data clock at a rate determined by the SR bits.

When cleared, this bit configures the SSI to the slave mode and aborts any transmission in progress. Transfers are initiated by an external master, which should supply the clock to the SCK pin.

CPOL — Clock Polarity

The clock polarity bit controls the state of the SCK pin between transmissions.

When this bit is set, pin SCK is high between transmissions.

When this bit is cleared, pin SCK is low between transmissions.

In both cases the data is latched on the rising edge of SCK for serial input and is valid on the rising edge of SCK for serial output. A reset sets this bit.

NOTE: *If the SSI is used as a slave, the SCK input pin must be active before enabling the SSI. For example, if CPOL = 0, SCK must be low; if CPOL = 1, SCK must be high.*

T/$\overline{R}$ — Transmit/Receive

This bit must be set to allow data to be driven on the SDIO pin (transmitting). It must be cleared to disable the SDIO drivers when receiving data. It is cleared by a reset.

SR1 and SR0 — SSI Rate

These bits determine the frequency of SCK when in master mode (MSTR = 1). They have no effect in slave mode (MSTR = 0).

Table 9-1. SSI Rates

SR1 and SR0	SCK Rates (Hz) at f_{osc} Frequency
00	32 kHz
01	64 kHz
10	128 kHz
11	256 kHz

9.4.2 SSI Status Register

This register is located at address $001C and contains two bits. Reset clears both of these bits.

Address: $001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SF	DCOL	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 9-4. SSI Status Register (SSR)

SF — SSI Flag

This bit is set upon occurrence of the last rising clock edge and indicates that a data transfer has taken place. If MSTR = 0 and SIE = 0, this bit has no effect on any further transmissions and can be ignored without problem. However, the SF flag must be clear to write the data register, or if SIE = 1 to clear the interrupt. If MSTR = 1, the SF flag must be cleared between transfers. The SF flag can be cleared three different ways: (1) by reading the SSR with SF set, followed by a read or write of the serial data register, (2) by a system reset, or, (3) by disabling the SSI. If the SF flag is cleared before the last edge of the next byte, it will be set again.

DCOL — Data Collision

This is a read-only status bit, which indicates that an invalid access to the data register was made. An invalid access can be one of the following conditions:

- An access of the SDR register in the middle of a transfer (after the first falling edge of SCK and before SF is set)
- An access of the SDR register made before an access of the SSR register (after SF is set)

DCOL is cleared by reading the status register with SF set followed by a read or write of the data register. A reset also clears this bit.

9.4.3 SSI Data Register

This register is located at address $001D and is both the transmit and receive data register. This system is not double buffered, but any writes to this register during transfers are masked and will not destroy the previous contents. The SDR can be read at any time, but, if a transfer is in progress the results may be ambiguous. The contents of this register could be altered whenever the CPOL bit is altered. This register should be written to only upon completion of a transfer, after the SF flag has been cleared. Otherwise, the new data will not be stored.

For an SSI configured as a master, to initiate a transfer, the data register write must occur after the SSI is enabled.

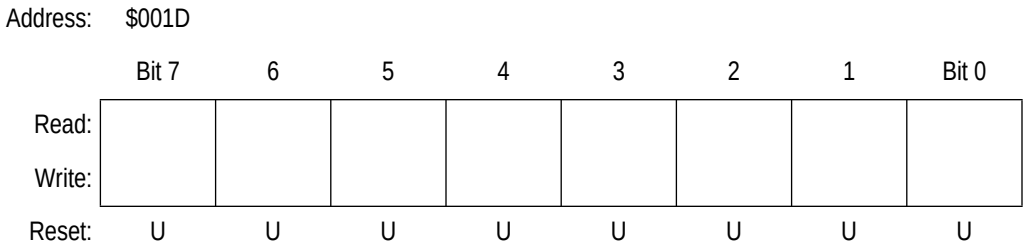


Figure 9-5. SSI Data Register (SDR)

9.5 Operation During Wait Mode

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

9.6 Operation During Stop Mode

In stop mode, the SSI halts operation. The SDIO and SCK pins will maintain their states.

If the SSI is nearing completion of a transfer when the stop mode is entered, it might be possible for the SSI to generate an interrupt request and thus cause the processor immediately to exit stop mode. To prevent

this occurrence, the programmer should ensure that all transfers are complete before entering the stop mode.

If the SSI is configured to slave mode, further care should be taken in entering stop mode. The SCK pin will still accept a clock from an external master, allowing potentially unwanted transfers to take place and power consumption to be increased. The SSI will not generate interrupt requests in this situation but, on exiting stop mode through some other means, the SF flag may be found to be set and an interrupt request will be generated if SIE is also set at this point.

To avoid these potential problems, it is safer to disable the SSI completely (SE = 0) before entering stop mode.

The synchronous serial interface (SSI) is a 2-wire master/slave system including serial clock (SCK) and serial data input/output (SDIO). When operating as a master device, the serial clock speed is selectable between four rates.

9.7 SSI Power Supply Source

The power supplied to the SSI is V_{DD} and V_{SS}, thus keeping the V_{DD2} and V_{SS2} free for the noise susceptible modules.

Section 10. Core Timer

10.1 Contents

10.2 Introduction

The core timer for this device is a 12-stage multifunctional ripple counter. Features include timer overflow, power-on reset (POR), real-time interrupt (RTI), and COP watchdog timer.

As seen in **Figure 10-1**, the internal peripheral clock is divided by four, and then 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 core timer counter register (CTCR) at address $09. A timer overflow function is implemented on the last stage of this counter, giving a possible interrupt rate of the internal peripheral clock(E)/1024. This point is then followed by three more stages, with the resulting clock (E/16384) driving the real-time interrupt circuit (RTI). The RTI circuit consists of three divider stages with a one-of-four selector. The output of the RTI circuit is further divided by eight to drive the mask optional COP watchdog timer circuit. The RTI rate selector bits and the RTI and CTOF enable bits and flags are located in the timer control and status register at location $08.

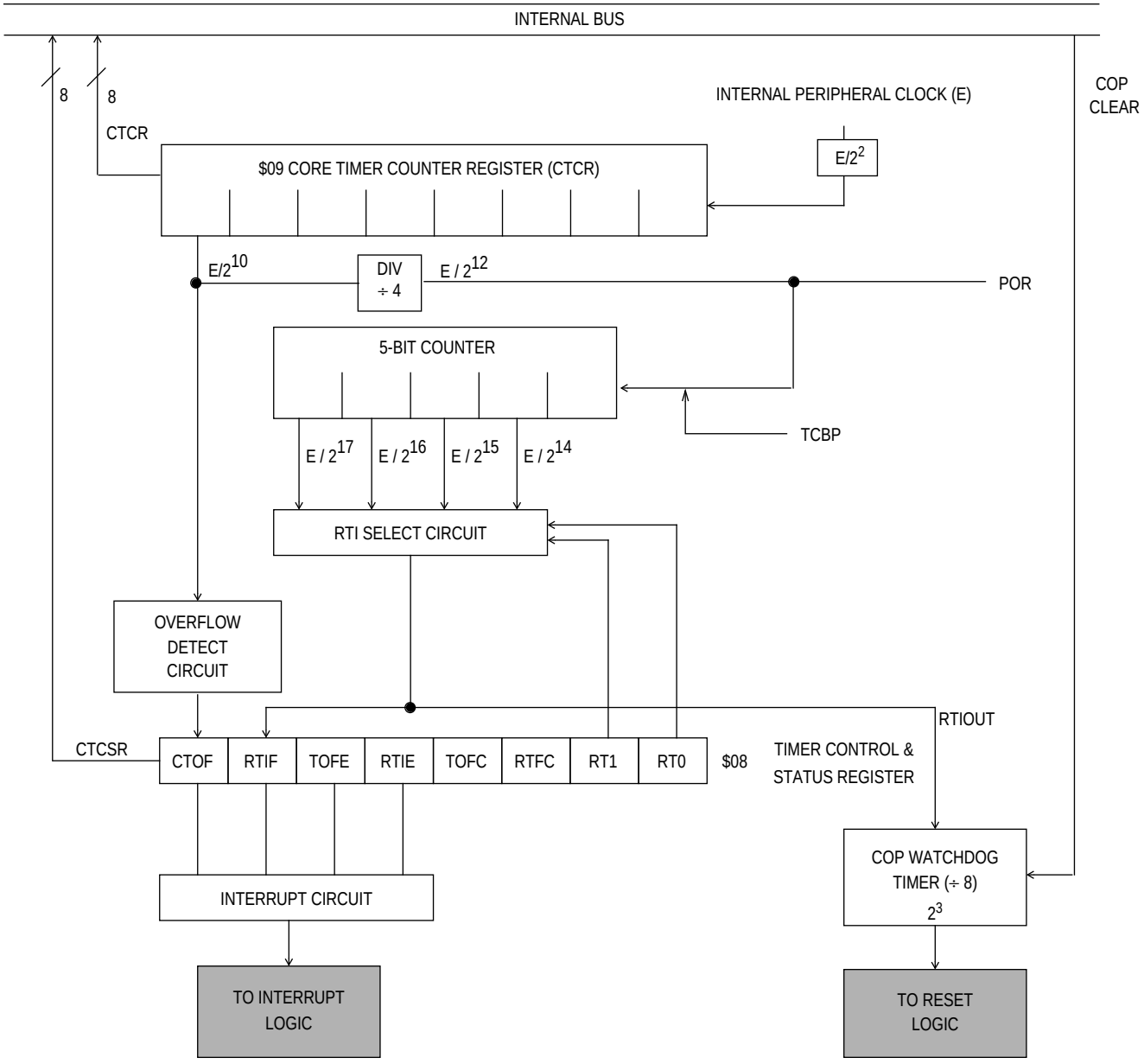


Figure 10-1. Core Timer Block Diagram

10.3 Core Timer Control and Status Register

The CTCSR contains the timer interrupt flag, the timer interrupt enable bits, and the real time interrupt rate select bits. **Figure 10-2** shows the value of each bit in the CTCSR when coming out of reset.

Address $0008

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CTOF	RTIF	TOFE	RTIE	0	0	RT1	RT0
Write:					TOFC	RTRC		
Reset:	0	0	0	0	0	0	1	1

= Unimplemented

Figure 10-2. Core Timer Control and Status Register (CTCSR)

CTOF — Core Timer Overflow

CTOF is a read-only status bit set when the 8-bit ripple counter rolls over from $FF to $00. Clearing CTOF is done by writing a logic 1 to TOFC. Writing to CTOF has no effect. Reset clears CTOF.

RTIF — Real Time Interrupt Flag

The real-time interrupt circuit consists of a 3-stage divider and a one-of-four selector. The clock frequency that drives the RTI circuit is E/2**14 (or E/16,384) with three additional divider stages, giving a maximum interrupt period of 64 milliseconds at a bus rate of 2.048 MHz. RTIF is a clearable, read-only status bit and is set when the output of the chosen (one-of-four selection) stage goes active. Clearing RTIF is done by writing a logic 1 to RTFC. Writing to RTIF has no effect. Reset clears RTIF.

TOFE — Timer Overflow Enable

When this bit is set, a CPU interrupt request is generated when the CTOF bit is set. Reset clears this bit.

RTIE — Real-Time Interrupt Enable

When this bit is set, a CPU interrupt request is generated when the RTIF bit is set. Reset clears this bit.

TOFC — Timer Overflow Flag Clear

When a logic 1 is written to this bit, CTOF is cleared. Writing a logic 0 has no effect on the CTOF bit. This bit always reads as zero.

RTFC — Real-Time Interrupt Flag Clear

When a logic 1 is written to this bit, RTIF is cleared. Writing a logic 0 has no effect on the RTIF bit. This bit always reads as zero.

RT1 and RT0 — Real-Time Interrupt Rate Select

These two bits select one of four taps from the real-time interrupt circuit. See **Table 10-1**. Reset sets these two bits, which selects the slowest periodic rate and gives the maximum time in which to alter these bits if necessary. Care should be taken when altering RT0 and RT1 if the timeout 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.

Table 10-1. RTI and COP Rates at 2.048 MHz

RTI Rate 2.048 MHz		RT1 and RT0	Minimum COP Rates 2.048 MHz	
8 ms	$2^{14}/E$	00	$(2^{17}–2^{14})/E$	56 ms
16 ms	$2^{15}/E$	01	$(2^{18}–2^{15})/E$	112 ms
32 ms	$2^{16}/E$	10	$(2^{19}–2^{16})/E$	224 ms
64 ms	$2^{17}/E$	11	$(2^{20}–2^{17})/E$	448 ms

10.4 Core Timer Counter Register

The timer counter register is a read-only register that contains the current value of the 8-bit ripple counter at the beginning of the timer chain. This counter is clocked by the CPU clock (E/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.

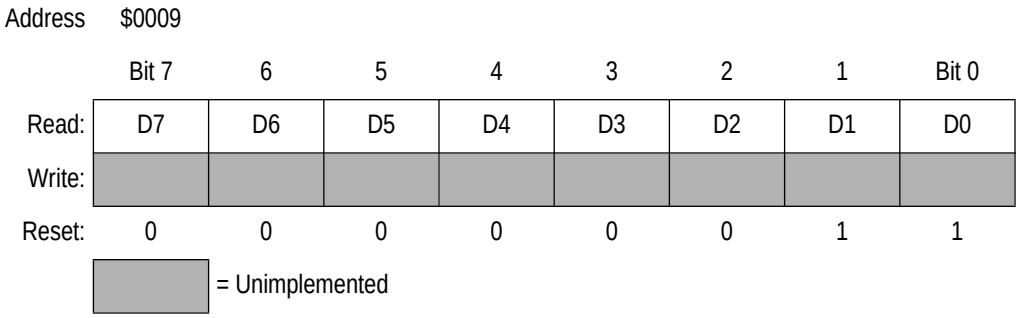


Figure 10-3. Core Counter Register (CTCR)

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 starts counting up from zero and normal device operation begins. When $\overline{\text{RESET}}$ is asserted any time during operation (other than POR), the counter chain is cleared.

10.5 Computer Operating Properly (COP) 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 10-1**. If the COP circuit times out, an internal reset is generated and the normal reset vector is fetched. Preventing a COP timeout or clearing the COP is accomplished by writing a logic 0 to bit 0 of address $1FF0. When the COP is cleared, only the final divide-by-eight stage (output of the RTI) is cleared.

If the COP watchdog timer is allowed to time out, an internal reset is generated to reset the MCU. In addition, the $\overline{\text{RESET}}$ pin is pulled low for a minimum of one E-clock cycle for emulation purposes.

The COP remains enabled after execution of the WAIT instruction and all associated operations apply. If the STOP instruction is disabled, execution of STOP instruction causes the CPU to execute a NOP instruction. In addition, the COP is prohibited from being held in RESET. This prevents a device lock-up condition.

This COP's objective is to make it impossible for this device to become stuck or locked-up and to be sure the COP is able to rescue the part from any situation where it might entrap itself in abnormal or unintended behavior. This function is a mask option.

10.6 Timer During Wait Mode

The CPU clock halts during wait mode, but the timer remains active. If interrupts are enabled, a timer interrupt will cause the processor to exit wait mode. The COP is always enabled while in user mode.

10.7 Core Timer Power Supply Source

The core timer is supplied by V_{DD} and V_{SS}. V_{DD2} and V_{SS2} are not needed here because this module is not susceptible to supply noise.

Section 11. Dual Phase-Locked Loop (PLL)

11.1 Contents

11.2 Introduction

This dual PLL is similar to that of the MC145162 60-MHz universal programmable dual PLL frequency synthesizer. It is especially designed for CT-1 cordless phone applications.

The PLL features fully programmable 16-bit receive, 16-bit transmit, and 12-bit reference ripple down counters. It also has two independent phase detectors for transmit and receive loops.

The FinTx and FinRx signals are input to the PLL transmit and receive counters, respectively. They are typically driven by the loop VCO and AC-coupled. The minimum input signal level is 200 mV_{p-p} @ 60.0 MHz.

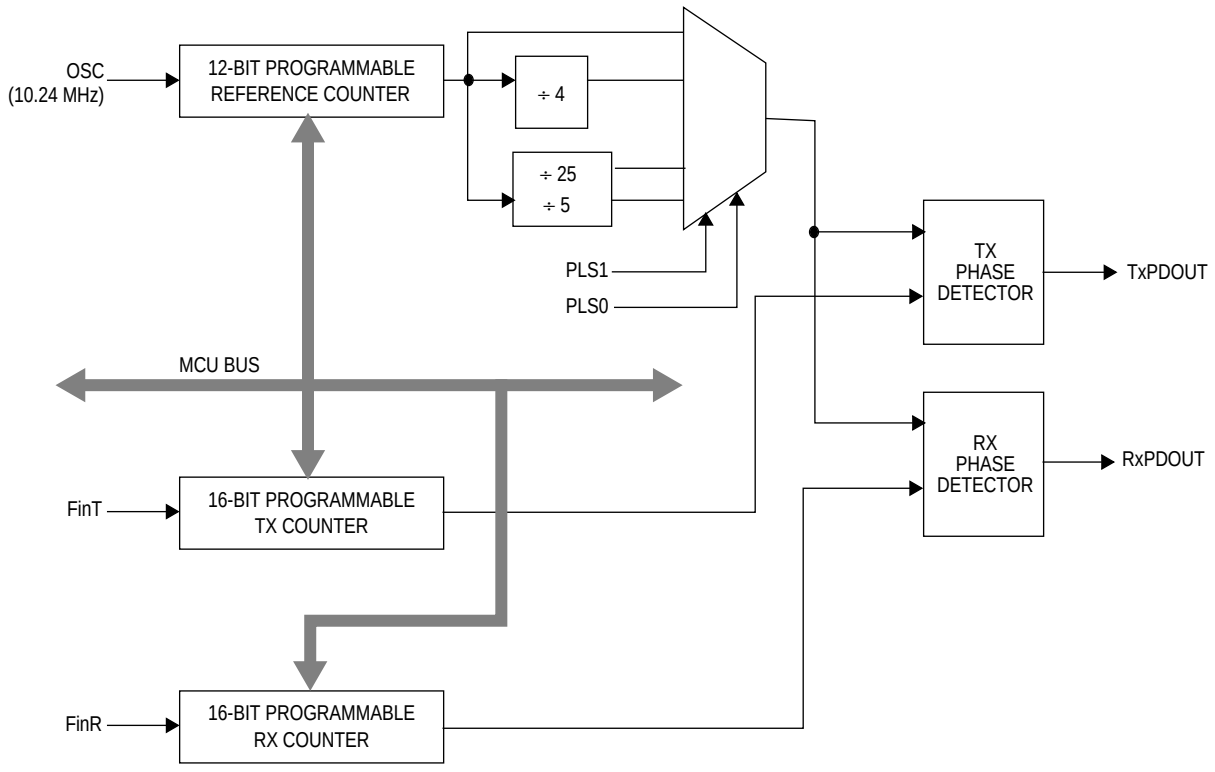


Figure 11-1. Dual PLL Block Diagram

11.3 Registers

The PLL has one 12-bit programmable counter, two 16-bit programmable counters, and one control register.

11.3.1 Dual Control Register

The PLLCR contains bits that affect the operation of the PLL.

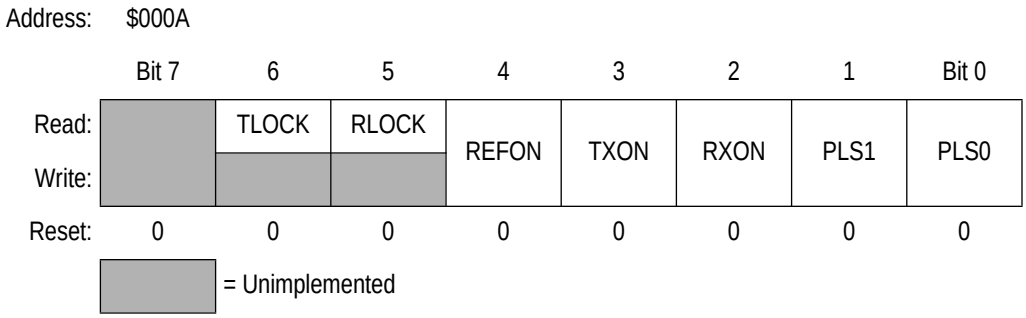


Figure 11-2. Dual PLL Control Register

PLS1 and PLS0 — PLL Reference Counter Select

These bits select between the PLL reference counter MUX outputs. This output signal then drives the phase detectors.

Table 11-1. PLL Reference Counter Select

PLS1 and PLS0	PLL Reference Counter Output
00	÷12-Bit Counter
01	÷4 After 12-Bit Counter
10	÷5 After 12-Bit Counter
11	÷25 After 12-Bit Counter

RXON — RX Counter Enable

When set, this bit enables the PLL receive counter.

When clear, it stops the receive counter in a reset state to save power. RXON also shuts off the associated phase detector and holds it in three-state. Initializing the receive counter before it is enabled is recommended.

TXON — TX Counter Enable

When set, this bit enables the PLL transmit counter.

When clear, it stops the counter in a reset state to save power. TXON also shuts off the associated phase detector and holds it in three-state. Initializing the transmit counter before it is enabled is recommended.

REFON — Reference Counter Enable

When set, this bit enables the PLL reference counter.

When clear, REFON stops the reference counter in a reset state to save power. Initializing the reference counter before it is enabled is recommended.

RLOCK — Receive Lock Detect

This bit is read only and is not latched. When set, this bit is a real-time indication of an active correction pulse from the phase detect in progress. When clear, no correction is made and the output from the phase detect is three-stated. Using multiple reads at known intervals will allow the user to filter and judge a LOCK condition.

TLOCK — Transmit Lock Detect

This bit is read only and is not latched.When set, this bit is a real-time indication of an active correction pulse from the phase detect in progress. When clear, no correction is made and the output from the phase detect is three-stated. Using multiple reads at known intervals will allow the user to filter and judge a LOCK condition.

11.3.2 12-Bit Reference Counter Modulus Register

This 2-byte register holds the count for the 12-bit reference counter. The reference counter is shut off and held in reset when the REFON bit is cleared. For proper operation, this register must not be loaded with a value less than $000F.

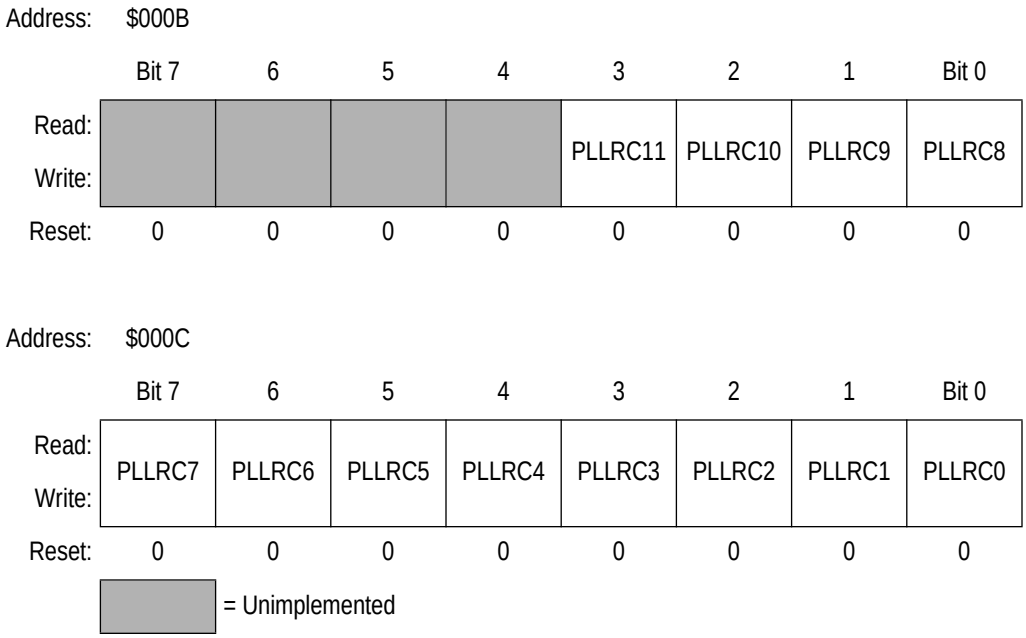


Figure 11-3. 12-Bit Reference Counter (PLLRC)

NOTE: *Bit 4 to bit 7 of $000B are not used but they are physically present. The user may use these four bits for scratch memory.*

11.3.3 16-Bit Transmit Counter Modulus Register

This 2-byte register holds the count for the 16-bit transmit counter. The transmit counter is shut off and held in reset when the TXON bit is cleared. For proper operation, this register must not be loaded with a value less than $000F.

Address: $000D

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PLLTX15	PLLTX14	PLLTX13	PLLTX12	PLLTX11	PLLTX10	PLLTX9	PLLTX8
Reset:	0	0	0	0	0	0	0	0

Address: $000E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PLLTX7	PLLTX6	PLLTX5	PLLTX4	PLLTX3	PLLTX2	PLLTX1	PLLTX0
Reset:	0	0	0	0	0	0	0	0

Figure 11-4. 16-Bit Transmit Counter (PLLTX)

11.3.4 16-Bit Receive Counter Modulus Register

This 2-byte register holds the count for the 16-bit receive counter. The receive counter is shut off and held in reset when the RXON bit is cleared. For proper operation, this register must not be loaded with a value less than $000F.

Address: $000F

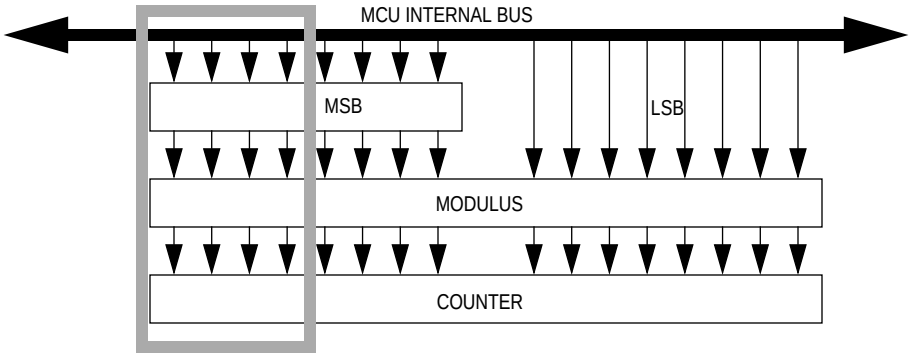
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PLLRX15	PLLRX14	PLLRX13	PLLRX12	PLLRX11	PLLRX10	PLLRX9	PLLRX8
Reset:	0	0	0	0	0	0	0	0

Address: $0010

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PLLRX7	PLLRX6	PLLRX5	PLLRX4	PLLRX3	PLLRX2	PLLRX1	PLLRX0
Reset:	0	0	0	0	0	0	0	0

Figure 11-5. 16-Bit Receive Counter (PLLRX)

The modulus registers mentioned above have specific read/write logic. When the user updates the contents of the modulus registers, the MSB must be written first and is temporarily stored in a temporary buffer. This inhibits the transfer of data from this level to the next level, which is the modulus register. When the LSB is written, all 12 or 16 bits (data from the temporary buffer and the MCU internal bus) are transferred to the modulus register simultaneously. This prevents the loading of bad data from the modulus register. A read of the data registers is the reverse of a write operation. A read of the LSB buffers the MSB. A subsequent read of the MSB reflects this buffered value. Since only one temporary buffer exists, two sequential MSB writes of different registers will result in only the last data value stored in the temporary buffer. The first value will be lost.



NOTE: Not available for the 12-bit reference counter.

Figure 11-6. Counter Structure Block Diagram

11.4 PLL Power Supply Source

The PLL is supplied by V_{DD}, V_{DD2}, V_{SS}, and V_{SS2}. V_{DD} and V_{SS} are reserved for the digital circuitry, while V_{DD2} and V_{SS2} are reserved for the analog circuitry such as the phase detect and the amplifiers which are sensitive to supply noise.

Section 12. Pulse Width Modulator (PWM)

12.1 Contents

12.2 Introduction

The pulse width modulator (PWM) system has one 6-bit channel to enable the correct pulse output. The PWM has a fixed frequency of E/64, where E is the internal bus frequency. For a PWM output frequency of 32 kHz, E must be 2.048 MHz. This corresponds to a 10.24-MHz crystal with the divide-by-five crystal option.

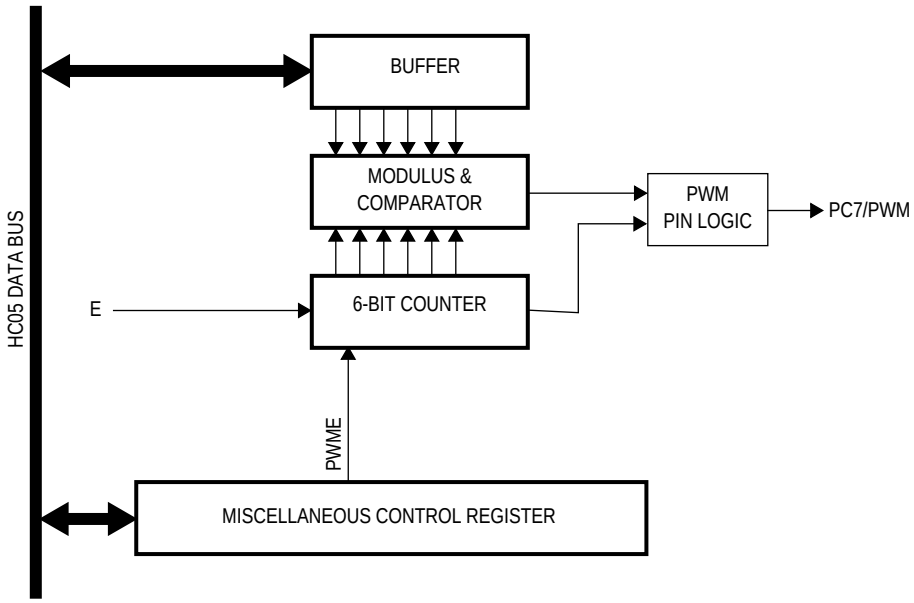


Figure 12-1. PWM Block Diagram

12.3 Functional Description

A $00 in the PWM data register yields an off output (0%), but a $3F yields a duty of 63/64 (98.4%).

When not in use, the PWM system can be shut off to save power by clearing the PWME bit in MISCR.

Writes to the PWM data register can be performed at any time without affecting the current PWM output signal. Updates on the PWM output occur at the end of the PWM period (E x 64). At this time, the new value is loaded into the PWM data register. If a write to the registers is performed while the PWM is disabled, data is transferred directly to the PWM register. A read of the data register reads the active count in progress, not the buffered value.

After the PWM is enabled (PWM = 1) the PWM output remains low for one E cycle. This allows synchronization and will not occur again until the next time the PWM is enabled.

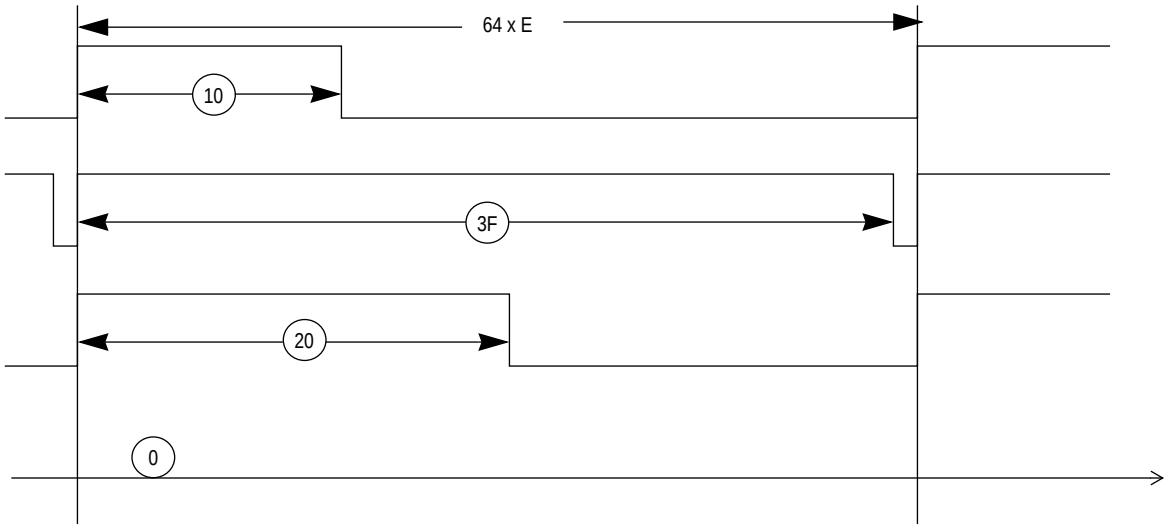


Figure 12-2. PWM Waveforms

12.4 PWM Data Register

One PWM data register (PWMDR), located at $0020, is associated with the PWM system. This 8-bit data register holds the duty cycle for the PWM output; however, only six of these bits are used. PWMDR can be written to and read at any time. Writes to the PWMDR are buffered and are not transferred to the active register until the end of the PWM cycle during which the write was executed.

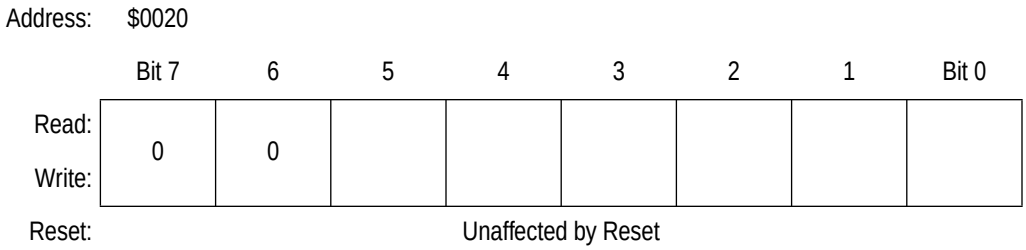


Figure 12-3. PWM Data Register (PWMDR)

Upon RESET the user should write to the data register prior to enabling the PWM system (for example, prior to setting the PWME bit in the miscellaneous control register). This avoids an erroneous duty cycle from being driven.

PWME — Miscellaneous Control Register PWM Enable Bit

When set, this bit enables the PWM subsystem. Its main function is to allow the user to save power when not using the PWM. When enabled, the clocks are active to the module. When clear, this bit shuts off the clocks to the module and relinquishes control of PC7 to the port C logic (the PC7 pullup option should not be selected if the PWM is used). Reset clears this bit. This bit is located in the miscellaneous control register found in **Section 13. Comparators**.

12.5 PWM During Wait Mode

The PWM continues normal operation during wait mode. To decrease power consumption during WAIT, it is recommended that the PWME bit in the miscellaneous control register be cleared if the PWM D/A converter is not being used.

12.6 PWM During Stop Mode

In stop mode, the oscillator is stopped, causing the PWM to cease function. Any signal in process is halted in whatever phase the signal happens to be in. Disabling the PWM before executing the STOP instruction is recommended.

12.7 PWM During Reset

Upon reset the PWME bit is cleared. In effect, this disables the PWM system and configures the PC7/PWM pin as a high impedance port C input pin. The user should write to the data register prior to enabling the PWM system (for instance, prior to setting PWME). This avoids an erroneous duty cycle from being driven.

12.8 PWM Power Supply Source

The PWM is supplied by V_{DD} and V_{SS}, due to the noise generated by the digital circuitry, since this module is not affected by supply noise.

Pulse Width Modulator (PWM)

Section 13. Comparators

13.1 Contents

13.2 Introduction

Port D shares its pins with CMP3+/PD0, CMP1–/PD1 and CMP12+/PD2, CMP2–/PD3 of the comparators. The output of the comparators is a status bit internal to the MCU. This circuitry is used for the comparison of analog signals, with a digital output. The comparator circuitry may be powered up by setting the CEN1, CEN2, and the CEN3 bits in the comparator control/status register. This register is located at $0022 and is cleared by reset. The state of the comparator output bit upon $\overline{\text{RESET}}$ is logic 0. For further electrical information see **Section 16. Electrical Specifications**.

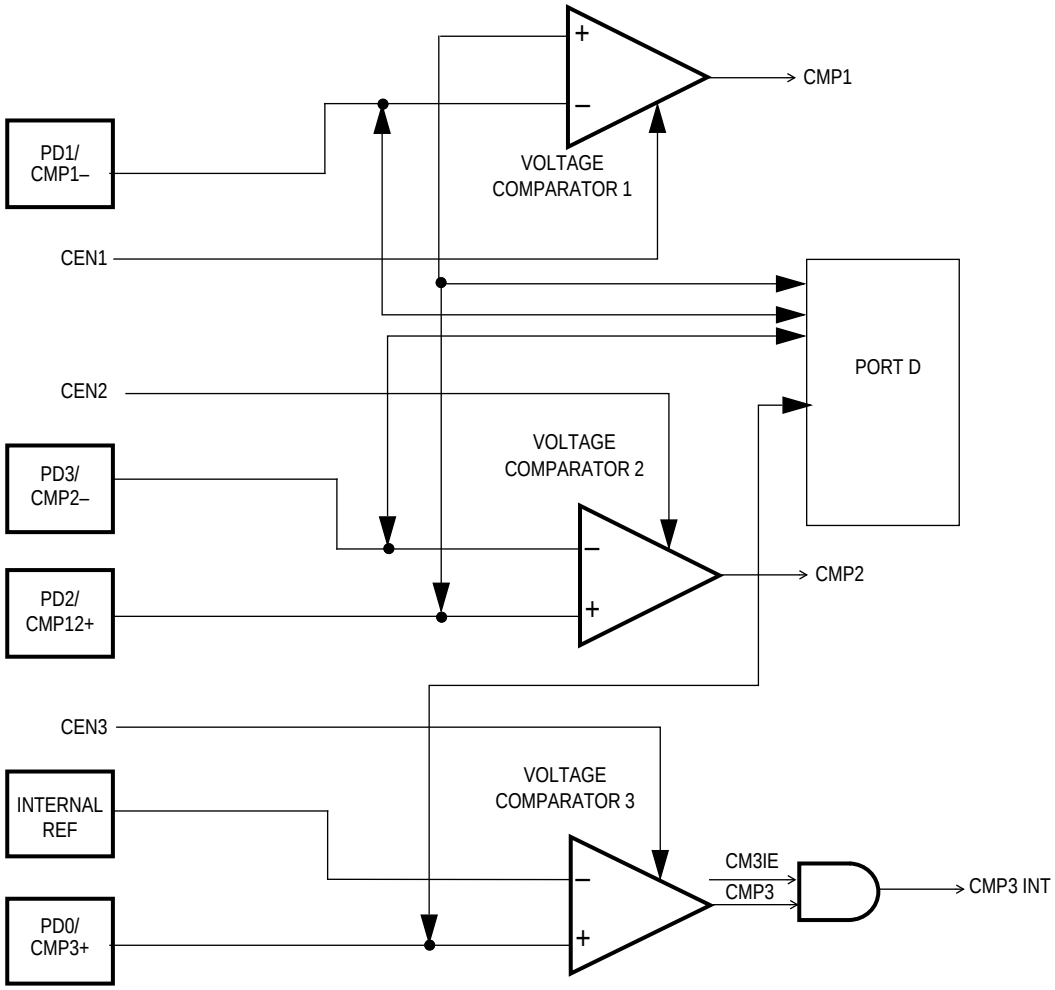


Figure 13-1. Comparator Block Diagram

13.3 Comparator Control/Status Register

Address: $0022

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CMP3	CMP2	CMP1	CM3IE	0	CEN3	CEN2	CEN1
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 13-2. Comparator Control/Status Register (CMPCSR)

CMP3 — Comparator 3 Output

This bit is implemented in two ways: as a real-time comparator output or as a latched interrupt source. The state of CM3IE determines which implementation is selected.

CM3IE = 0 (nonlatched mode) **—** This status bit is cleared only by two means:

1. Clearing CM3IE while the voltage at CMP3+ is less than CMP3–
2. By an external reset

CM3IE = 1 (latched mode) — This latched status bit is set when the voltage at CMP3+ (PD0) is larger than that of CMP3– (internal voltage reference ~ 2 V_{DD}/3). An interrupt is then initiated ($1FFA–$1FFB).

CMP2 — Comparator 2 Output

This status bit is set when the voltage at CMP12+ is larger than that of CMP2–, otherwise, it is cleared. Reset clears this bit because the comparator is disabled.

CMP1 — Comparator 1 Output

This status bit is set when the voltage at CMP12+ is larger than that of CMP1–, otherwise, it is cleared. Reset clears this bit because the comparator is disabled.

CM3IE — Comparator 3 Interrupt Enable

This control bit, when set, allows CMP3 to generate an interrupt when it goes high.

CEN3 — Comparator 3 Enable

This control bit, when set, powers up the voltage comparator 3 on PD0. The user must set this bit to allow any functionality of the comparator. After enabling the comparator, the user should delay for t_{CEN} before reading the state of the output or enabling the interrupt. Reset clears this bit.

PD0 remains under control of its DDR bit to enable the user to drive CMP3+ to a desired level. When CEN3 is cleared, the comparator is disabled and consumes negligible power.

CEN2 — Comparator 2 Enable

This control bit, when set, powers up the voltage comparator 2 on PD2 and PD3. The user must set this bit to allow any functionality of the comparator. After enabling the comparator, the user should delay for t_{CEN} before reading the state of the output. Reset clears this bit.

PD2 and PD3 remain under control of their DDR bits to enable the user to drive CMP2– and CMP12+ to desired levels. When CEN2 is cleared, the comparator is disabled and consumes negligible power.

CEN1 — Comparator 1 Enable

This control bit, when set, powers up voltage comparator 1 on PD1 and PD2. The user must set this bit to allow any functionality of the comparator. After enabling the comparator, the user should delay for t_{CEN} before reading the state of the output. Reset clears this bit.

PD1 and PD2 remain under control of their DDR bits to enable the user to drive CMP1– and CMP12+ to desired levels. When CEN1 is cleared, the comparator is disabled and consumes negligible power.

13.4 Reading Comparator Outputs

The state of each comparator output bit is internally readable from the CMPCSR.

13.5 Comparator During Wait Mode

The comparator operates normally in wait mode. If the user wishes to save power during the wait mode, the CEN1, CEN2, and CEN3 bits should be cleared before the WAIT instruction.

13.6 Comparator During Stop Mode

Stop mode does not affect the comparator circuit. If the user needs to save power, the CEN1, CEN2, and CEN3 bits should be cleared before the STOP instruction.

13.7 Comparator Power Supply Source

The comparators are supplied by V_{DD2} and V_{SS2}. These power lines are reserved for noise-susceptible circuitry, such as the circuitry found in the comparators.

Comparators

Section 14. Miscellaneous Register

14.1 Contents

14.2 Introduction

The miscellaneous register exists to hold various system control bits. It is located at address $0021.

14.3 Miscellaneous Control Register

Address: $0021

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	0	0	0	0	SPEED	COE	PWME	0
Reset:	0	0	0	0	0	0	0	0

Figure 14-1. Miscellaneous Control Register (MISCR)

SPEED — CPU Speed Select

This control bit, when set, selects an additional divide-by-eight after the already divide-by-five from the oscillator clock to the internal bus clock. This makes the internal CPU clock a divide-by-40 from the oscillator input.

When clear, the internal bus clock is a divide-by-five from the oscillator input. Reset clears this bit.

COE — 16-Bit Timer Output Compare Enable

This control bit, when set, enables the 16-bit output compare function to come out on PD6/TCMP. When clear, PD6 returns to a general-purpose I/O pin.

PWME — PWM Enable

This control bit, when set, enables the PWM function to come out on PC7/PWM. When clear, PC7 returns to a general-purpose I/O pin.

14.4 Miscellaneous Register Power Supply Source

The miscellaneous control register is supplied by V_{DD} and V_{SS}. V_{DD2} and V_{SS2} are reserved for noise-susceptible circuitry.

Section 15. Instruction Set

15.1 Contents

15.2 Introduction

The MCU instruction set has 62 instructions and uses eight addressing modes. The instructions include all those of the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. The MUL instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is stored in the index register, and the low-order product is stored in the accumulator.

15.3 Addressing Modes

The CPU uses eight addressing modes for flexibility in accessing data. The addressing modes provide eight different ways for the CPU to find the data required to execute an instruction. The eight addressing modes are:

- Inherent
- Immediate
- Direct
- Extended
- Indexed, no offset
- Indexed, 8-bit offset
- Indexed, 16-bit offset
- Relative

15.3.1 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 operand address and are one byte long.

15.3.2 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 operand address and are two bytes long. The opcode is the first byte, and the immediate data value is the second byte.

15.3.3 Direct

Direct instructions can access any of the first 256 memory locations 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.

15.3.4 Extended

Extended instructions use three bytes and can 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.

15.3.5 Indexed, No Offset

Indexed instructions with no offset are 1-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the effective 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.

15.3.6 Indexed, 8-Bit Offset

Indexed, 8-bit offset instructions are 2-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 effective 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.

15.3.7 Indexed,16-Bit Offset

Indexed, 16-bit offset instructions are 3-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 effective 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.

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.

15.3.8 Relative

Relative addressing is only for branch instructions. If the branch condition is true, the CPU finds the effective 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.

15.4 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

15.4.1 Register/Memory Instructions

These instructions operate on CPU registers and memory locations. Most of them use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory.

Table 15-1. 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

15.4.2 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.

NOTE: *Do not use read-modify-write operations on write-only registers.*

Table 15-2. Read-Modify-Write Instructions

Instruction	Mnemonic
Arithmetic Shift Left (Same as LSL)	ASL
Arithmetic Shift Right	ASR
Bit Clear	BCLR(1)
Bit Set	BSET(1)
Clear Register	CLR
Complement (One's Complement)	COM
Decrement	DEC
Increment	INC
Logical Shift Left (Same as ASL)	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(2)

1. Unlike other read-modify-write instructions, BCLR and BSET use only direct addressing.
2. TST is an exception to the read-modify-write sequence because it does not write a replacement value.

15.4.3 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.

The BRCLR and BRSET instructions cause a branch based on the state of any readable bit in the first 256 memory locations. These 3-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 effective 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 15-3. 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

15.4.4 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory, which includes I/O registers and on-chip RAM locations. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations.

Table 15-4. Bit Manipulation Instructions

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

15.4.5 Control Instructions

These instructions act on CPU registers and control CPU operation during program execution.

Table 15-5. 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

15.5 Instruction Set Summary

Table 15-6. Instruction Set Summary

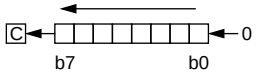
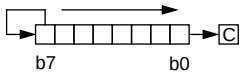
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr*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

Table 15-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BIH *rel*	Branch if IRQ Pin High	PC ← (PC) + 2 + *rel* ? IRQ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if IRQ Pin Low	PC ← (PC) + 2 + *rel* ? IRQ = 1	—	—	—	—	—	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	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
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3
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
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

Table 15-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
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}$) = $FF – (A) X ← ($\overline{X}$) = $FF – (X) M ← ($\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)	—	—	↕×	↕	↕	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
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

Table 15-6. Instruction Set Summary (Continued)

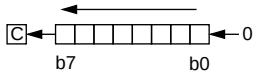
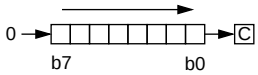
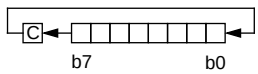
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Effective Address	—	—	—	—	—	DIR EXT IX2 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	dd ff 	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2
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

Table 15-6. Instruction Set Summary (Continued)

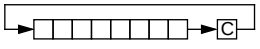
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕×	↕	↕	DIR INH INH IX1 IX	36 46 56 66 76	dd ff	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return 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		9
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	—	—	—	—	—	INH	81		6
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 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
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

Table 15-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
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		—	0×	—	—	—	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

15.6 Opcode Map

See **Table 15-7**.

Table 15-7. 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	5 BRSET0 3 DIR	5 BSET0 2 DIR	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0
1	5 BRCLR0 3 DIR	5 BCLR0 2 DIR	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1
2	5 BRSET1 3 DIR	5 BSET1 2 DIR	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2
3	5 BRCLR1 3 DIR	5 BCLR1 2 DIR	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3
4	5 BRSET2 3 DIR	5 BSET2 2 DIR	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4
5	5 BRCLR2 3 DIR	5 BCLR2 2 DIR	3 BCS/BLO 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5
6	5 BRSET3 3 DIR	5 BSET3 2 DIR	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6
7	5 BRCLR3 3 DIR	5 BCLR3 2 DIR	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7
8	5 BRSET4 3 DIR	5 BSET4 2 DIR	3 BHCC 2 REL	5 ASL/LSL 2 DIR	3 ASLA/LSLA 1 INH	3 ASLX/LSLX 1 INH	6 ASL/LSL 2 IX1	5 ASL/LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8
9	5 BRCLR4 3 DIR	5 BCLR4 2 DIR	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9
A	5 BRSET5 3 DIR	5 BSET5 2 DIR	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A
B	5 BRCLR5 3 DIR	5 BCLR5 2 DIR	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B
C	5 BRSET6 3 DIR	5 BSET6 2 DIR	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C
D	5 BRCLR6 3 DIR	5 BCLR6 2 DIR	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D
E	5 BRSET7 3 DIR	5 BSET7 2 DIR	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E
F	5 BRCLR7 3 DIR	5 BCLR7 2 DIR	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 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	5 BRSET0 3 DIR

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

Section 16. Electrical Specifications

16.1 Contents

16.2 Introduction

This section contains the electrical and timing specifications.

16.3 Maximum Ratings

Maximum ratings are the extreme limits to which the MCU can be exposed without permanently damaging it.

The MCU contains circuitry to protect the inputs against damage from high static voltages; however, do not apply voltages higher than those shown in the table below. Keep V_{IN} and V_{OUT} within the range $V_{SS} \le (V_{IN}$ or $V_{OUT}) \le V_{DD}$. Connect unused inputs to the appropriate voltage level, either V_{SS} or V_{DD}.

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	–0.3 to +7.0	V
Input Voltage	V_{IN}	V_{SS} –0.3 to V_{DD} + 0.3	V
Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{IN}	V_{SS} –0.3 to 2 x V_{DD} + 0.3	
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Junction Temperature	T_J	+150	°C
Storage Temperature Range	T_{stg}	–65 to +150	°C

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to* ***16.6 5.0 V DC Electrical Characteristics*** *and* ***16.7 3.3 V DC Electrical Characteristics*** *for guaranteed operating conditions.*

16.4 Operating Range

Characteristic	Symbol	Value	Unit
Operating Temperature Range MC68HC05CT4 (Standard)	T_A	T_L to T_H 0 to +70	°C

16.5 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Dual In-Line Package	θ_{JA}	70	°C/W

16.6 5.0 V DC Electrical Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage I_{Load} = 10.0 µA I_{Load} = –10.0 µA	 V_{OL} V_{OH}	 — V_{DD} –0.1	 — —	 0.1 —	V
Output High Voltage (I_{Load} -–0.8 mA) Port A, Port B, Port C, Port D	V_{OH}	V_{DD} –0.8	—	—	V
Output Low Voltage (I_{Load} = 1.6 mA) Port A, Port B, Port C, Port D	V_{OL}	—	—	0.4	V
Input High Voltage Port A, Port B, Port C, Port D, $\overline{IRQ}$, $\overline{RE3.5SET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage Port A, Port B, Port C, Port D, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
Input Hysteresis TCAP, $\overline{RESET}$, $\overline{IRQ}$, PD4/SCK	V_{HYST}	0.8	0.9	1	V
Supply Current (see Notes) Run Wait Stop 25 °C 0 °C to +70 °C	 I_{DD}	 — — — —	 3 0.3 10 20	 10 1 25 30	 mA mA µA µA
Supply Current (see Notes) Run Wait Stop 25 °C 0 °C to +70 °C	 I_{DD2}	 — — — —	 1.5 0.5 200 250	 3 1.3 300 350	 mA mA µA µA
I/O Ports Hi-Z Leakage Current Port A, Port B, Port D	I_{OZ}	—	—	5	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1 PC0–PC7 with Pullups Enabled	I_{IN}	— –20	— –125	10 –250	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$	 C_{OUT} C_{INT}	 — —	 — —	 12 8	 pF

NOTES:

1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = 0 °C to +70 °C, unless otherwise noted
2. All values shown reflect average measurements.
3. Typical values at midpoint of voltage range, 25 °C only
4. Wait I_{DD} and I_{DD2} : Only timer system active.
5. Run (operating) I_{DD}, I_{DD2}, Wait I_{DD}, and I_{DD2}: Measured using external square wave clock source (f_{OSC} = 10.24 MHz); all inputs 0.2 V from rail; no dc loads; less than 50 pF on all outputs; C_L = 20 pF on OSC2
6. Wait, stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V
7. Stop I_{DD} and I_{DD2} are measured with OSC1 = V_{SS}.
8. Wait I_{DD} and I_{DD2} are affected linearly by the OSC2 capacitance.
9. Run, wait, and stop I_{DD2} are measured directly off the V_{DD2} power supply.

16.7 3.3 V DC Electrical Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage I_{Load} = 10.0 µA I_{Load} = –10.0 µA	 V_{OL} V_{OH}	 — V_{DD} –0.1	 — —	 0.1 —	 V
Output High Voltage (I_{Load} –0.2 mA) Port A, Port B, Port C, Port D	V_{OH}	V_{DD} –0.3	—	—	V
Output Low Voltage (I_{Load} = 1.6 mA) Port A, Port B, Port C, Port D	V_{OL}	—	—	0.3	V
Input High Voltage Port A, Port B, Port C, Port D, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage Port A, Port B, Port C, Port D, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
Supply Current (see Notes) Run Wait Stop 25 °C 0 °C to +70 °C	 I_{DD}	 — — — —	 1.5 0.15 5 10	 5 0.5 12 15	 mA mA µA µA
Supply Current (see Notes) Run Wait Stop 25 °C 0 °C to +70 °C	 I_{DD2}	 — — — —	 0.5 0.3 100 125	 2 0.8 150 175	 mA mA µA µA
I/O Ports Hi-Z Leakage Current Port A, Port B, Port D	I_{OZ}	—	—	5	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1 PC0–PC7 with Pullups Enabled	I_{IN}	— –20	— –50	10 –125	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$	 C_{OUT} C_{INT}	 — —	 — —	 12 8	 pF

NOTES:
1. V_{DD} = 3.3 Vdc ±10%, V_{SS} = 0 Vdc, T_A = 0 °C to +70 °C, unless otherwise noted
2. All values shown reflect average measurements.
3. Typical values at midpoint of voltage range, 25 °C only
4. Wait I_{DD} and I_{DD2} : Only timer system active.
5. Run (operating) I_{DD}, I_{DD2}, Wait I_{DD}, and I_{DD2}: Measured using external square wave clock source (f_{OSC} = 10.24 MHz); all inputs 0.2 V from rail; no dc loads; less than 50 pF on all outputs; C_L = 20 pF on OSC2
6. Wait, stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V
7. Stop I_{DD} and I_{DD2} are measured with OSC1 = V_{SS}.
8. Wait I_{DD} and I_{DD2} are affected linearly by the OSC2 capacitance.
9. Run, wait, and stop I_{DD2} are measured directly off the V_{DD2} power supply.

16.8 3.3 V and 5.0 V Control Timing

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal External Clock	f_{OSC}	— dc	10.24 10.24	MHz
Internal Operating Frequency Crystal (f_{OSC} /5) External Clock (f_{OSC} /5)	f_{OP}	— dc	2.048 2.048	MHz
Cycle Time	t_{CYC}	488	—	ns
Crystal Oscillator Startup Time	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{CYC}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	125	—	ns
Interrupt Pulse Period	t_{ILIL}	(Note 2)	—	t_{CYC}
OSC1 Pulse Width	t_{OH},t_{OL}	90	—	ns

NOTES:
1. V_{DD} = 3.0 to 5.5 Vdc, V_{SS} = 0 Vdc, T_A = 0 ^{o}C to +70 ^{o}C, unless otherwise noted
2. The minimum period t_{ILIL}, should not be less than the number of cycle times it takes to execute the interrupt service routine plus 2T t_{cyc}.

16.9 Comparators

Characteristic	Symbol	Min	Max	Unit
Input Voltage Range (see Note 2)	V_{INT}	V_{SS} –0.5	V_{DD} +0.5	V
Voltage Comparator Propagation Time Measured at 100 mV Overdrive (see Note 3)	t_{COMP}	—	10	µs
Common Mode Range (see Note 4)	V_{COM}	V_{SS} +1.5	V_{DD} –0.5	V
Offset (see Note 5)	V_{OFF}	—	± 20	mV
Comparator Enable Time (see Note 6)	t_{CEN}	—	100	ms
Input Current $V_{SS} \le V_{IN} \le V_{DD}$	I_{IN}	—	1	µA
Input Current $V_{IN} \le V_{SS}$	I_{IN}	—	4.0	mA
Gain Bandwidth Product		—	10	kHz
Slew Rate	SR	5	—	V/µs
Saturation Voltage (I_{OUT} = –5 mA)	V_{SAT}	—	0.5	V
Internal Voltage Reference CMP–	V_{REF}	x 90%	x 110%	(2/3) V_{DD}

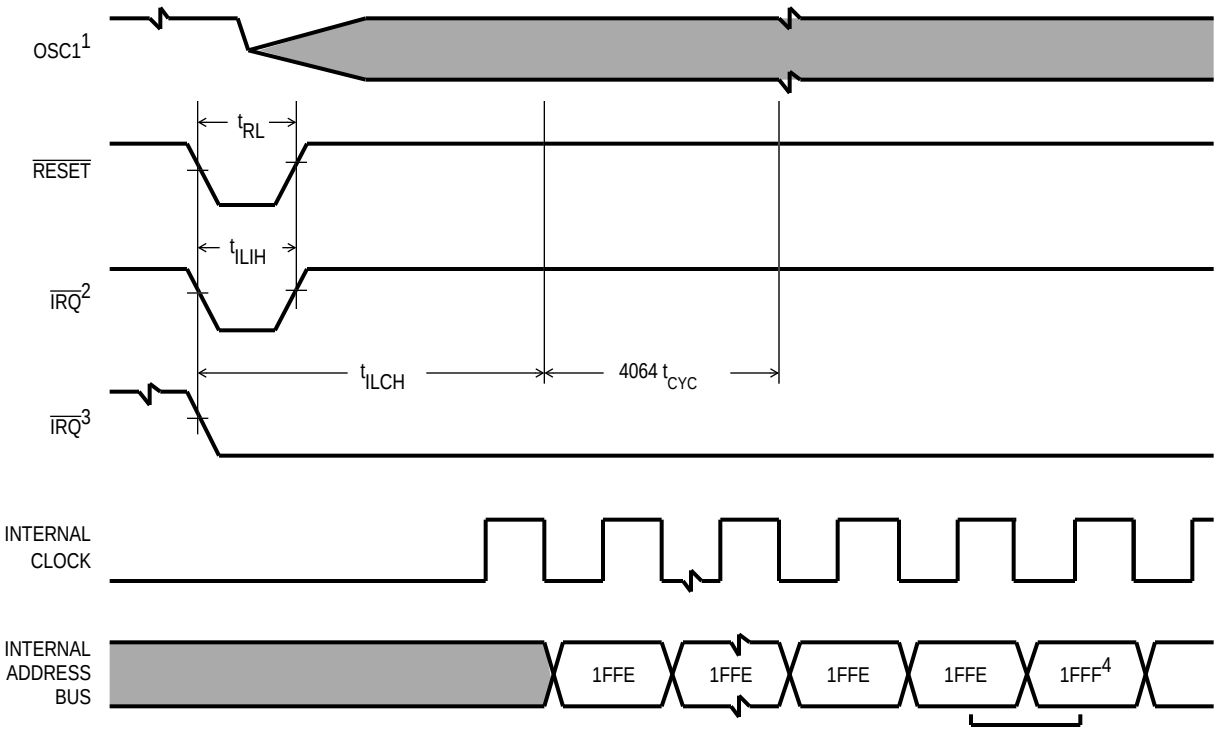
NOTES:
1. V_{DD} = 3.3 Vdc ±10%, V_{SS} = 0 Vdc, T_A = 0 °C to +70 °C, unless otherwise noted
2. The comparator is guaranteed to function over the specified input voltage range with no erroneous outputs.
3. Signal propagation time through the comparators measured with 100 mv of overdrive.
4. Comparator is guaranteed to meet specifications; for example, SR, t_{COMP}, and V_{OFF}.
5. Input offset voltage is guaranteed over the temperature range.
6. Enable time is the time from enabling the comparator with the CEN bit until the comparator is fully functional.

16.10 PWM Timing

Characteristic	Symbol	Min	Max	Unit
PWM Rise Time	t_{PWMR}	15	35	ns
PWM Fall Time	t_{PWMR}	15	35	ns

16.11 PLL Signal Input

Characteristic	Symbol	Min	Max	Unit
Input Voltage on FINTX and FINRX (peak-to-peak)	V_{VCO}	200	—	mV_{p-p}
Input Frequency on FINTX and FINRX	f_{VCO}	—	60	MHz



NOTES:
1. Represents the internal gating of the OSC1 pin
2. IRQ pin edge-sensitive mask option
3. IRQ pin level- and edge-sensitive mask option
4. RESET vector address shown for timing example

Figure 16-1. Stop Recovery Timing Diagram

Section 17. Mechanical Specifications

17.1 Contents

17.2 Introduction

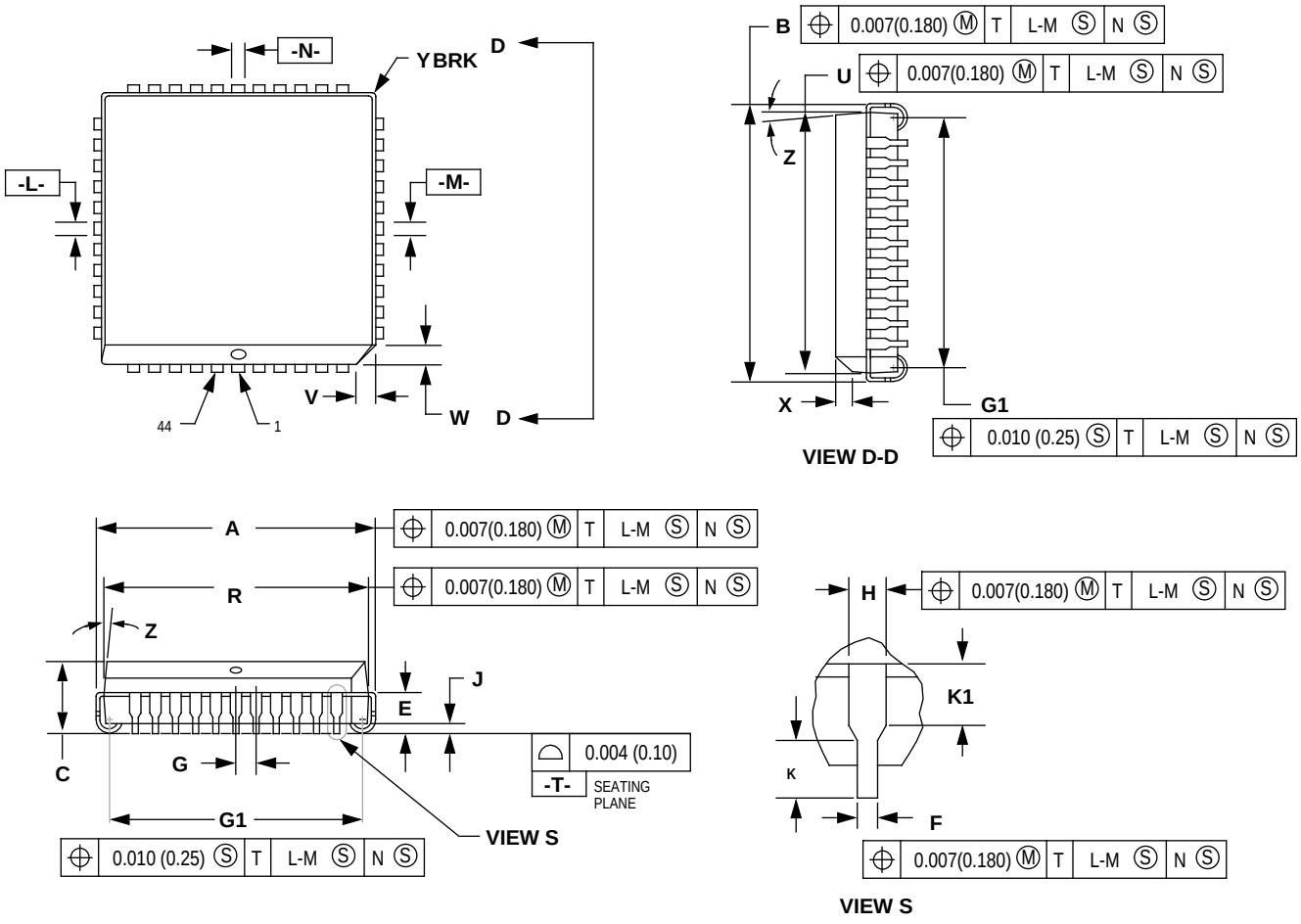
The MC68HC05CT4 is available in a 44-pin plastic-leaded chip carrier (PLCC) and a 44-pin quad flat pack (QFP). Package dimensions are provided in this section.

The following figures show the latest package information at the time of this publication. To make sure that you have the latest package specifications, contact one of the following:

- Local Motorola Sales Office
- Motorola Fax Back System (Mfax™)
  - Phone 1-602-244-6609
  - EMAIL RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/
- Worldwide Web (wwweb) home page at http://motorola.com/sps/

Follow Mfax or wwweb on-line instructions to retrieve the current mechanical specifications.

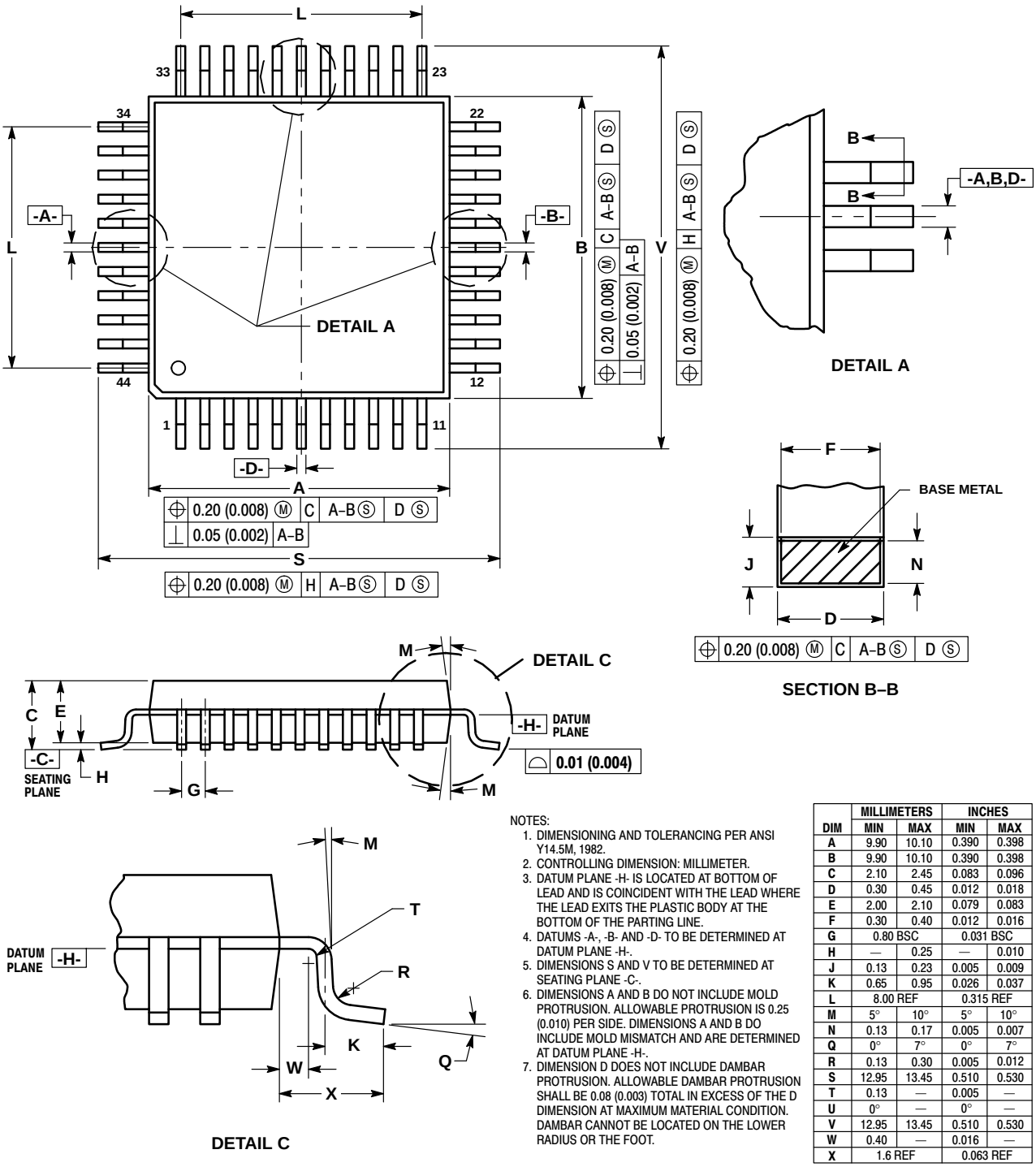
17.3 44-Lead Plastic-Leaded Chip Carrier (Case 777-02)



NOTES:
1. DATUMS -L-, -M-, AND -N- ARE DETERMINED WHERE TOP OF LEAD SHOLDERS EXITS PLASTIC BODY AT MOLD PARTING LINE.
2. DIMENSION G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
3. DIMENSION R AND U DO NOT INCLUDE MOLD FLASH. ALLOWABLE MOLD FLASH IS 0.010 (0.25) PER SIDE.
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
5. CONTROLLING DIMENSION: INCH.
6. THE PACKAGE TOP MAY BE SMALLER THAN THE PACKAGE BOTTOM BY UP TO 0.012 (0.300). DIMENSIONS R AND U ARE DETERMINED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY EXCLUSIVE OF THE MOLD FLASH, TIE BAR BURRS, GATE BURRS AND INTERLEAD FLASH, BUT INCLUDING ANY MISMATCH BETWEEN THE TOP AND BOTTOM OF THE PLASTIC BODY.
7. DIMINSION H DOES NOT INCLUDE DAMBAR PROTRUSION OR INTRUSION. THE DAMBAR PROTUSION(S) SHALL NOT CAUSE THE H DIMINSION TO BE GREATER THAN 0.037 (0.940136). THE DAMBAR INTRUSION(S) SHALL NOT CAUSE THE H DIMINISION TO SMALLER THAN 0.025 (0.635).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.685	0.695	17.40	17.65
B	0.685	0.695	17.40	17.65
C	0.165	0.180	4.20	4.57
E	0.090	0.110	2.29	2.79
F	0.013	0.019	0.33	0.48
G	0.050 BSC		1.27 BSC	
H	0.026	0.032	0.66	0.81
J	0.020	—	0.51	—
K	0.025	—	0.64	—
R	0.650	0.656	16.51	16.66
U	0.650	0.656	16.51	16.66
V	0.042	0.048	1.07	1.21
W	0.042	0.048	1.07	1.21
X	0.042	0.056	1.07	1.42
Y	—	0.020	—	0.50
Z	2°	10°	2°	10°
G1	0.610	0.630	15.50	16.00
K1	0.040	—	1.02	—

17.4 44-Lead Quad Flat Pack (Case 824A-01)



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	9.90	10.10	0.390	0.398
B	9.90	10.10	0.390	0.398
C	2.10	2.45	0.083	0.096
D	0.30	0.45	0.012	0.018
E	2.00	2.10	0.079	0.083
F	0.30	0.40	0.012	0.016
G	0.80 BSC		0.031 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	8.00 REF		0.315 REF	
M	5°	10°	5°	10°
N	0.13	0.17	0.005	0.007
Q	0°	7°	0°	7°
R	0.13	0.30	0.005	0.012
S	12.95	13.45	0.510	0.530
T	0.13	—	0.005	—
U	0°	—	0°	—
V	12.95	13.45	0.510	0.530
W	0.40	—	0.016	—
X	1.6 REF		0.063 REF	

Mechanical Specifications

Section 18. Ordering Information

18.1 Contents

18.2 Introduction

This section contains instructions for ordering custom-masked ROM MCUs.

18.3 MCU Ordering Forms

To initiate an order for a ROM-based MCU, first obtain the current ordering form for the MCU from a Motorola representative. Submit the following items when ordering MCUs:

- A current MCU ordering form that is **completely filled out** (Contact your Motorola sales office for assistance.)
- A copy of the customer specification if the customer specification deviates from the Motorola specification for the MCU
- Customer's application program on one of the media listed in **18.4 Application Program Media**

The current MCU ordering form is also available through the Motorola Freeware Bulletin Board Service (BBS). The telephone number is (512) 891-FREE. After making the connection, type bbs in lower-case letters. Then press the return key to start the BBS software.

18.4 Application Program Media

Please deliver the application program to Motorola in one of the following media:

- Macintosh®[1] 3 1/2 inch diskette (double-sided 800 K or double-sided high-density 1.4 M)
- MS-DOS®[2] or PC-DOS™[3] 3 1/2 inch diskette (double-sided 720 K or double-sided high-density 1.44 M)
- MS-DOS® or PC-DOS™ 5 1/4 inch diskette (double-sided double- density 360K or double-sided high-density 1.2M)

Use positive logic for data and addresses.

When submitting the application program on a diskette, clearly label the diskette with this information:

- Customer name
- Customer part number
- Project or product name
- File name of object code
- Date
- Name of operating system that formatted diskette
- Formatted capacity of diskette

On diskettes, the application program must be in Motorola's S-record format (S1 and S9 records), a character-based object file format generated by M6805 cross assemblers and linkers.

1. Macintosh is a registered trademark of Apple Computer, Inc.
2. MS-DOS is a registered trademark of Microsoft Corporation.
3. PC-DOS is a trademark of International Business Machines Corporation.

NOTE: *Begin the application program at the first user ROM location. Program addresses must correspond exactly to the available on-chip user ROM addresses as shown in the memory map. Write $00 in all nonuser ROM locations or leave all nonuser ROM locations blank. Refer to the current MCU ordering form for additional requirements. Motorola may request pattern re-submission if nonuser areas contain any nonzero code.*

If the memory map has two user ROM areas with the same address, then write the two areas in separate files on the diskette. Label the diskette with both file names.

In addition to the object code, a file containing the source code can be included. Motorola keeps this code confidential and uses it only to expedite ROM pattern generation in case of any difficulty with the object code. Label the diskette with the file name of the source code.

18.5 ROM Program Verification

The primary use for the on-chip ROM is to hold the customer's application program. The customer develops and debugs the application program and then submits the MCU order along with the application program.

Motorola inputs the customer's application program code into a computer program that generates a listing verify file. The listing verify file represents the memory map of the MCU. The listing verify file contains the user ROM code and may also contain nonuser ROM code, such as self-check code. Motorola sends the customer a computer printout of the listing verify file along with a listing verify form.

To aid the customer in checking the listing verify file, Motorola will program the listing verify file into customer-supplied blank preformatted Macintosh or DOS disks. All original pattern media are filed for contractual purposes and are not returned.

Check the listing verify file thoroughly, then complete and sign the listing verify form and return it to Motorola. The signed listing verify form constitutes the contractual agreement for the creation of the custom mask.

18.6 ROM Verification Units (RVUs)

After receiving the signed listing verify form, Motorola manufactures a custom photographic mask. The mask contains the customer's application program and is used to process silicon wafers. The application program cannot be changed after the manufacture of the mask begins. Motorola then produces 10 MCUs, called RVUs, and sends the RVUs to the customer. RVUs are usually packaged in unmarked ceramic and tested to 5 Vdc at room temperature. RVUs are not tested to environmental extremes because their sole purpose is to demonstrate that the customer's user ROM pattern was properly implemented. The 10 RVUs are free of charge with the minimum order quantity. These units are not to be used for qualification or production. RVUs are not guaranteed by Motorola Quality Assurance.

18.7 MC Order Numbers

Table 18-1 shows the MC order numbers for the available package types.

Table 18-1. MC Order Numbers

MC Order Number	Operating Temperature Range
MC68HC05CT4FN	0 °C to 70 °C
MC68HC05CT4FB	0 °C to 70 °C

NOTE: FN = 44-Lead Plastic-Leaded Chip Carrier
FB = 44-Lead Quad Flat Pack

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution, P.O. Box 5405, Denver, Colorado 80217, 1-800-441-2447 or 1-303-675-2140. Customer Focus Center, 1-800-521-6274

JAPAN: Nippon Motorola Ltd. SPD, Strategic Planning Office 4-32-1, Nishi-Gotanda Shinagawa-ku, Tokyo 141, Japan, 81-3-5487-8488

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

Mfax™, Motorola Fax Back System: RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/; TOUCHTONE, 1-602-244-6609; US and Canada ONLY, 1-800-774-1848

HOME PAGE: http://motorola.com/sps/

Mfax is a trademark of Motorola, Inc.

MOTOROLA

© Motorola, Inc., 1997

HC05CT4GRS/D