68HC05RC16
68HC705RC16

SPECIFICATION
(General Release)

CSIC System Design Group
Austin, Texas

Revision History

Change in unit's number indicates changes that can affect design. Changes in tenth's number indicates changes that do not affect design.

Revision	Date	Location of Change
1.0	5/95	
2.0	10/95	1.3.9 PC0 through PC3 (PC4 through PC7) on page 1-8 2.2.3 Port C on page 2-5 Figure 9-5 on page 9-7

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

Section **Title** **Page**

SECTION 11
ELECTRICAL SPECIFICATIONS

SECTION 12
MECHANICAL SPECIFICATIONS

SECTION 13
ORDERING INFORMATION

LIST OF FIGURES

LIST OF TABLES

SECTION 1
GENERAL DESCRIPTION

The MC68HC05RC16 is a general-purpose, low-cost addition to the M68HC05 Family of microcontroller units (MCUs) and is suitable for remote control applications. It contains the HC05 central processing unit (CPU) core, including the 14-stage core timer with real-time interrupt (RTI) and computer operating properly (COP) watchdog systems. On-chip peripherals include a carrier modulator transmitter. The 16-Kbyte memory map has 15,936 bytes of user ROM and 352 bytes of RAM. There are 20 input-output (I/O) lines (eight having keyscan logic and pullups) and a low-power reset pin. The MC68HC05RC16 is available in 28-pin small outline integrated circuit (SOIC) or dual-in-line package (DIP) packages. Four additional I/O lines are available for bond out in higher pin count packages.

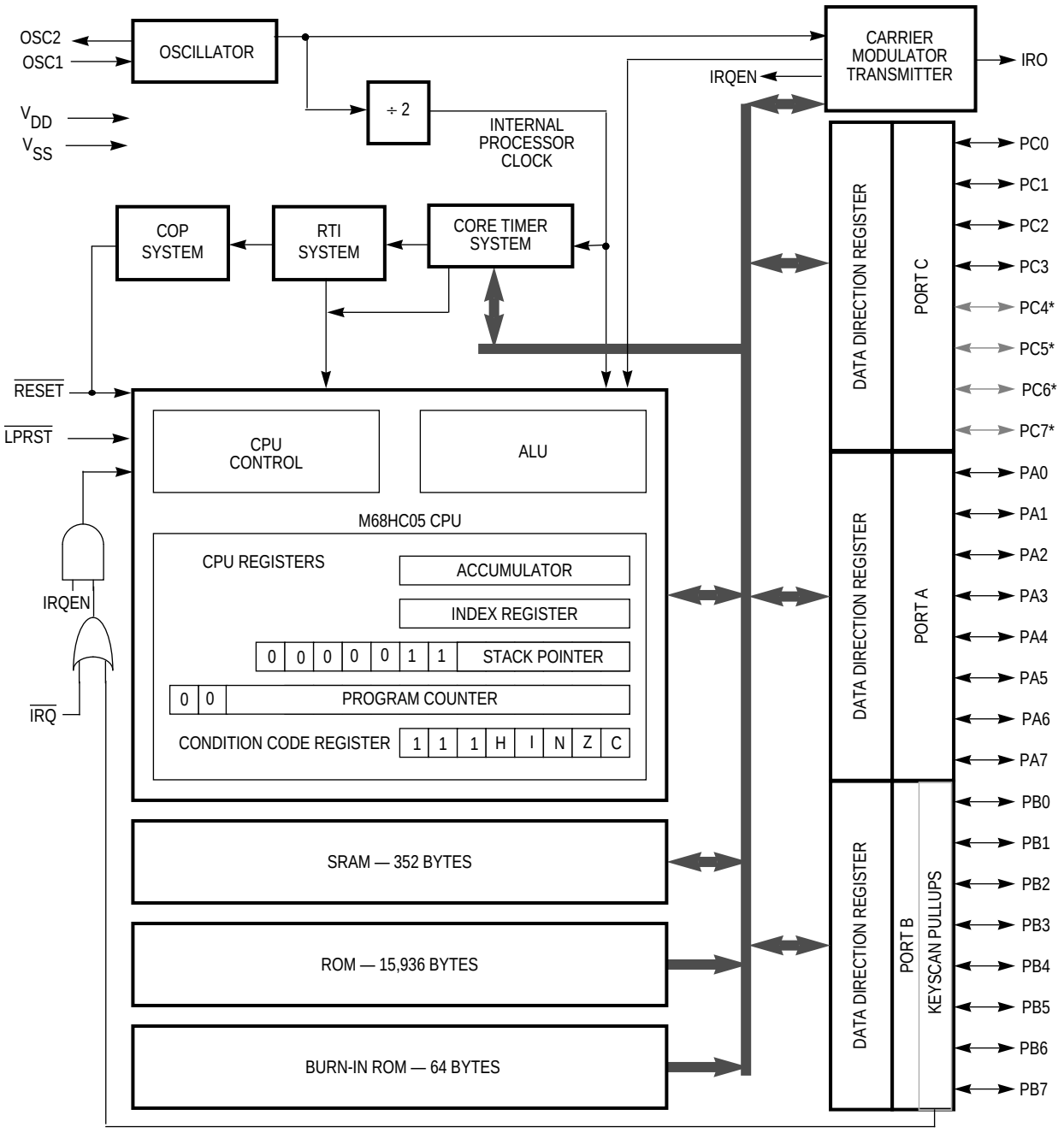
1.1 Features

- Low Cost
- HC05 Core
- 28-Pin SOIC or DIP Packages
- On-Chip Oscillator with Crystal/Ceramic Resonator
- 4 MHz Maximum Oscillator Frequency at 5 V Supply
- 2 MHz Maximum Oscillator Frequency at 2.2 V Supply
- Fully Static Operation
- 15,936 Bytes of User ROM
- 64 Bytes of Burn-In ROM
- 352 Bytes of On-Chip RAM
- 14-Stage Core Timer with Real-Time Interrupt (RTI) and Computer Operating Properly (COP) Watchdog Circuits
- Carrier Modulator Transmitter Supporting Baseband, Pulse Length Modulator (PLM), and Frequency Shift Keying (FSK) Protocols
- Low-Power Reset Pin
- 20 Bidirectional I/O Lines (Four Additional I/O Lines Available for Bond Out in Higher Pin Count Packages)

- Mask Programmable Pullups and Interrupt on Eight Port Pins (PB0 through PB7)
- High Current Infrared (IR) Drive Pin
- High Current Port Pin (PC0)
- Power-Saving Stop and Wait Modes
- Mask Selectable Options:
  - COP Watchdog Timer
  - STOP Instruction Disable
  - Edge-Sensitive or Edge- and Level-Sensitive Interrupt Trigger
  - Port B Pullups for Keyscan
  - Illegal Address Reset
  - ROM Security

NOTE

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



* Marked pins are available only in higher pin count (>28) packages.

Figure 1-1. MC68HC05CRC16 Block Diagram

1.2 Mask Options

There are 11 total mask options on the MC68HC05RC16 including:

- Eight Port B Pullups
- IRQ Sensitivity
- COP Enable/Disable
- STOP Enable/Disable

These are non-programmable options in that they are selected at the time of code submission (when masks are made). These options are as follows:

PB7PU — Port B7 Pullup/Interrupt

This bit enables or disables the pullup/interrupt on port B, bit 7.
1 = Enables the pullup/interrupt
0 = Disables the pullup/interrupt

PB6PU — Port B6 Pullup/Interrupt

This option enables or disables the pullup/interrupt on port B, bit 6.
1 = Enables pullup/interrupt
0 = Disables pullup/interrupt

PB5PU — Port B5 Pullup/Interrupt

This option enables or disables the pullup/interrupt on port B, bit 5.
1 = Enables pullup/interrupt
0 = Disables pullup/interrupt

PB4PU — Port B4 Pullup/Interrupt

This option enables or disables the pullup/interrupt on port B, bit 4.
1 = Enables pullup/interrupt
0 = Disables pullup/interrupt

PB3PU — Port B3 Pullup/Interrupt

This option enables or disables the pullup/interrupt on port B, bit 3.
1 = Enables pullup/interrupt
0 = Disables pullup/interrupt

PB2PU — Port B2 Pullup/Interrupt

This option enables or disables the pullup/interrupt on port B, bit 2.
1 = Enables pullup/interrupt
0 = Disables pullup/interrupt

PB1PU — Port B1 Pullup/Interrupt

This option enables or disables the pullup/interrupt on port B, bit 1.
1 = Enables pullup/interrupt
0 = Disables pullup/interrupt

PB0PU — Port B0 Pullup/Interrupt

This option enables or disables the pullup/interrupt on port B, bit 0.

1 = Enables pullup/interrupt
0 = Disables pullup/interrupt

COPEN — COP Enable

When the COP option is selected (COPEN = 1), the COP watchdog timer is enabled.

When the COP option is deselected (COPEN = 0), the COP watchdog timer is disabled.

STOPEN — STOP Instruction Enable

When the STOP option is selected (STOPEN = 1), the STOP instruction is enabled.

When the STOP option is deselected (STOPEN = 0), the STOP instruction is disabled.

IRQ — IRQ sensitivity

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

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

1.3 Signal Description

Pinout for the 28-pin dual-in-line package is shown in Figure 1-2. The signals are described in the following subsections.

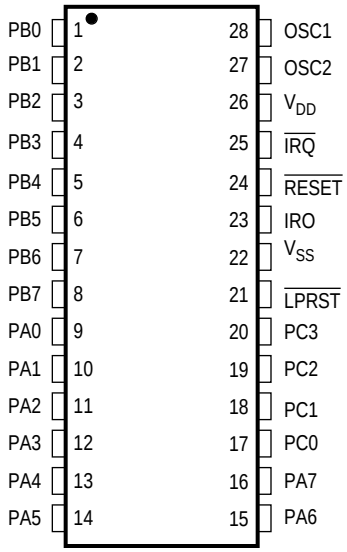


Figure 1-2. 28-Pin DIP Pinout

1.3.1 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.

1.3.2 $\overline{IRQ}$ (Maskable Interrupt Request)

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 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 one 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 one 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 wired-OR operation.

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

Refer to **SECTION 4 INTERRUPTS** for more detail.

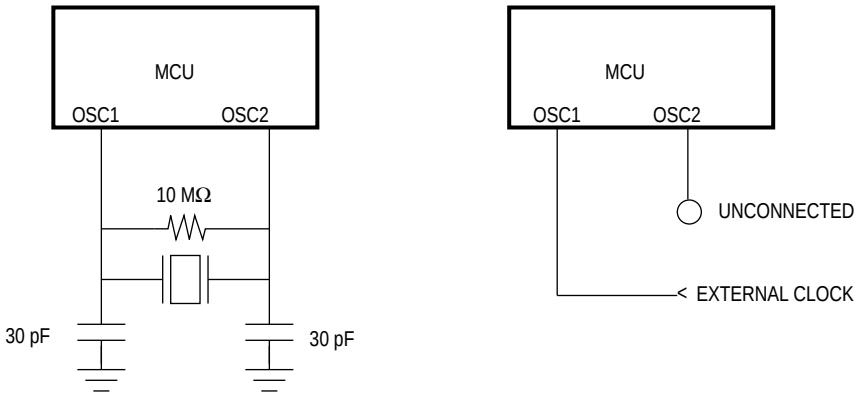
1.3.3 OSC1 and OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, or an external signal connects to these pins to provide a system clock. The oscillator frequency is two times the internal bus rate.

Figure 1-3 shows the recommended circuit when using a crystal. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time.

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. Figure 1-3 (a) shows the recommended circuit for using a ceramic resonator. The manufacturer of the particular ceramic resonator being considered should be consulted for specific information.

An external clock should be applied to the OSC1 input with the OSC2 pin not connected (see Figure 1-3 (b)). This setup can be used if the user does not want to run the CPU with a crystal.



(a) Crystal/Ceramic Resonator Oscillator Connections

(b) External Clock Source Connections

Figure 1-3. Oscillator Connections

1.3.4 $\overline{\text{RESET}}$

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

1.3.5 $\overline{\text{LPRST}}$

The $\overline{\text{LPRST}}$ pin is an active low pin and is used to put the MCU into low-power reset mode. In low-power reset mode the MCU is held in reset with all processor clocks halted. The $\overline{\text{LPRST}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. See **SECTION 5 RESETS**.

1.3.6 IRO

The IRO pin is the high current source and sink output of the carrier modulator transmitter subsystem which is suitable for driving IR LED biasing logic. See **SECTION 8 CARRIER MODULATOR TRANSMITTER (CMT)**.

1.3.7 PA0 through PA7

These eight I/O lines comprise port A. The state of any pin is software programmable and all port A lines are configured as inputs during power-on or reset. For detailed information on I/O programming, see **2.2 Input/Output Programming**.

1.3.8 PB0 through 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 inputs during power-on or reset. Each port B I/O line has a mask optionable pullup/interrupt for keyscan. For detailed information on I/O programming, see **2.2 Input/Output Programming**.

1.3.9 PC0 through PC3 (PC4 through PC7)

These eight I/O lines comprise port C. PC0 is a high current pin. PC4 through PC7 are available only in higher pin count (>28) packages. The state of any pin is software programmable and all port C lines are configured as input during power-on or reset. For detailed information on I/O programming, see **2.2 Input/Output Programming**.

NOTE

Only four bits of port C are bonded out in 28-pin packages for the MC68HC05RC16, although port C is truly an 8-bit port. Since pins PC4–PC7 are unbonded, software should include the code to set their respective data direction register locations to outputs to avoid floating inputs.

NOTE

Any unused inputs and I/O ports should be tied to an appropriate logic level (either V_{DD} or V_{SS}). Although the I/O ports of the MC68HC05RC16 do not require termination, it is recommended to reduce the possibility of static damage.

SECTION 2
MEMORY

This section describes the organization of the on-chip memory.

2.1 Memory Map

The MC68HC05RC16 has a 16-Kbyte memory map consisting of user ROM, RAM, burn-in ROM, control registers, and I/O.

Figure 2-1 is a memory map of the MCU. Figure 2-2 is a more detailed memory map of the I/O register section.

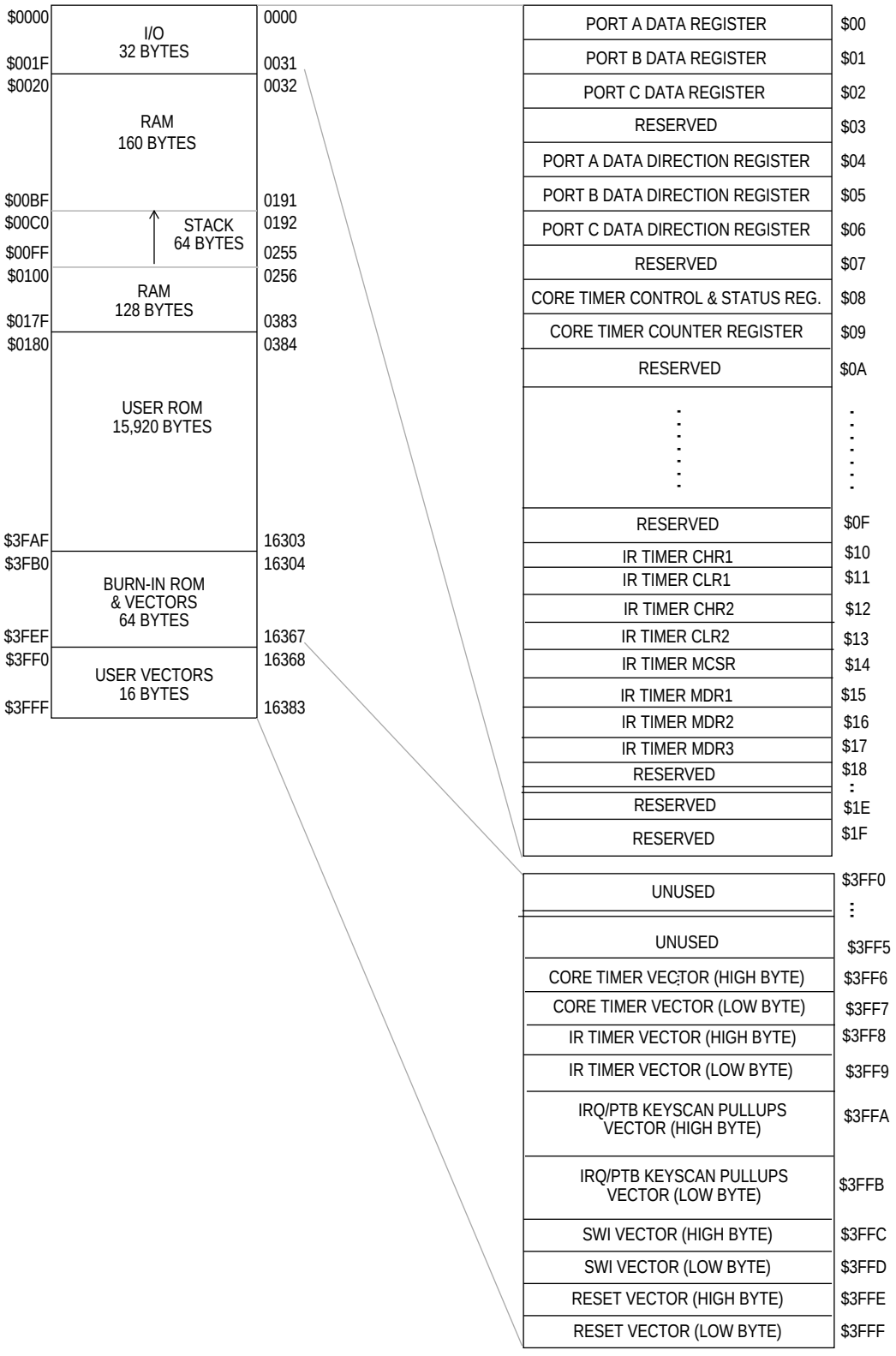


Figure 2-1. 16-KByte Memory Map

ADDR	REGISTER	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
$00	PORT A DATA								
$01	PORT B DATA								
$02	PORT C DATA								
$03	Reserved								
$04	PORT A DDR								
$05	PORT B DDR								
$06	PORT C DDR								
$07	Reserved								
$08	TIMER CONTROL & STATUS REGISTER	CTOF	RTIF	TOFE	RTIE	TOFC	RTFC	RT1	RT0
$09	TIMER COUNTER REGISTER								
$0A	Reserved								
$0B	Reserved								
$0C	Reserved								
$0D	Reserved								
$0E	Reserved								
$0F	Reserved								
$10	IR TIMER CHR1	IROLN	0	PH5	PH4	PH3	PH2	PH1	PH0
$11	IR TIMER CLR1	IROLP	0	PL5	PL4	PL3	PL2	PL1	PL0
$12	IR TIMER CHR2	0	0	SH5	SH4	SH3	SH2	SH1	SH0
$13	IR TIMER CLR2	0	0	SL5	SL4	SL3	SL2	SL1	SL0
$14	IR TIMER MCSR	EOC	0	EIMSK	EXMRK	BASE	MODE	EOCIE	MCGEN
$15	IR TIMER MDR1	MB11	MB10	MB9	MB8	SB11	SB10	SB9	SB8
$16	IR TIMER MDR2	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
$17	IR TIMER MDR3	SB7	SB6	SB5	SB4	SB3	SB2	SB1	SB0
$18	Reserved								
$19	Reserved								
$1A	Reserved								
$1B	Reserved								
$1C	Reserved								
$1D	Reserved								
$1E	Reserved								
$1F	Reserved								

Figure 2-2. I/O Registers

2.1.1 ROM

The user ROM consists of 15,920 bytes of ROM located from $0180 to $3FAF and 16 bytes of user vectors located from $3FF0 to $3FFF.

The burn-in ROM is located from $3FB0 to $3FEF.

Ten of the user vectors, $3FF6 through $3FFF, are dedicated to reset and interrupt vectors. The six remaining locations — $3FF0, $3FF1, $3FF2, $3FF3, $3FF4, and $3FF5 — are general-purpose user ROM locations.

2.1.2 ROM Security

Security has been incorporated into the MC68HC05RC16 to prevent external viewing of the ROM contents. This feature ensures that customer-developed software remains proprietary.

2.1.3 RAM

The user RAM consists of 352 bytes of a shared stack area. The RAM starts at address $0020 and ends at address $017F. 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.

2.2 Input/Output Programming

In user mode, 24 lines are arranged as three 8-bit I/O ports. These ports are programmable as either inputs or outputs under software control of the data direction registers.

NOTE

To avoid a glitch on the output pins, write data to the I/O port data register before writing a one to the corresponding data direction register.

2.2.1 Port A

Port A is an 8-bit bidirectional port which 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 register, but clears the data direction register, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode.

2.2.2 Port B

Port B is an 8-bit bidirectional port which does not share any of its pins with other subsystems. The address of the port B data register is $0001 and the data direction register (DDR) is at address $0005. Reset does not affect the data register, but clears the data direction register, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode. Each of the port B pins has a mask programmable pullup device that can be enabled. When the pullup device is enabled, this pin will also become an interrupt pin. The edge or edge and level sensitivity of the $\overline{\text{IRQ}}$ pin will also pertain to the enabled port B pins. Care needs to be taken when using port B pins that have the pullup enabled. Before switching from an output to an input, the data should be preconditioned to a logic one or the I bit should be set in the condition code register to prevent an interrupt from occurring.

2.2.3 Port C

Port C is an 8-bit bidirectional port (PC0 through PC7) which does not share any of its pins with other subsystems. The port C data register is at $0003 and the data direction register (DDR) is at $0006. Reset does not affect the data register, but clears the data direction register, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode. Port C pins PC4 through PC7 are available only in higher pin count (>28 pin) packages.

NOTE

Only four bits of port C are bonded out in 28-pin packages for the MC68HC05RC16, although port C is truly an 8-bit port. Since pins PC4–PC7 are unbonded, software should include the code to set their respective data direction register locations to outputs to avoid floating inputs.

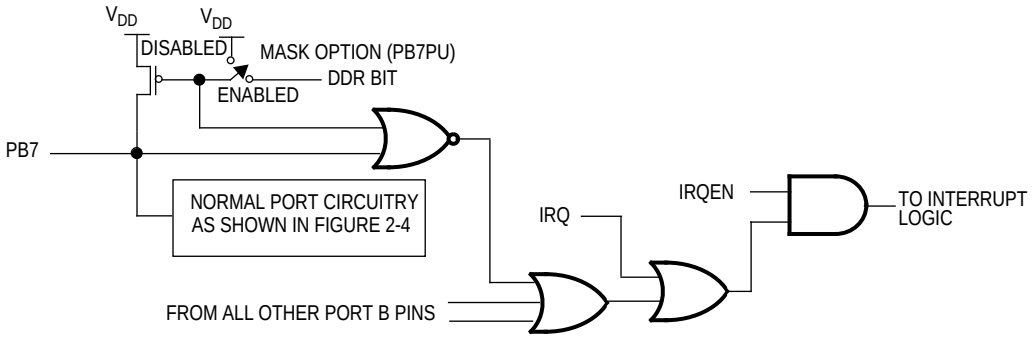


Figure 2-3. Port B Pullup Option

2.2.4 Input/Output 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 zero.

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.

Table 2-1. I/O Pin Functions

Access	DDR	I/O Pin Functions
Write	0	The I/O pin is in input mode. Data is written into the output data latch.
Write	1	Data is written into the output data latch and output to the I/O pin.
Read	0	The state of the I/O pin is read.
Read	1	The I/O pin is in an output mode. The output data latch is read.

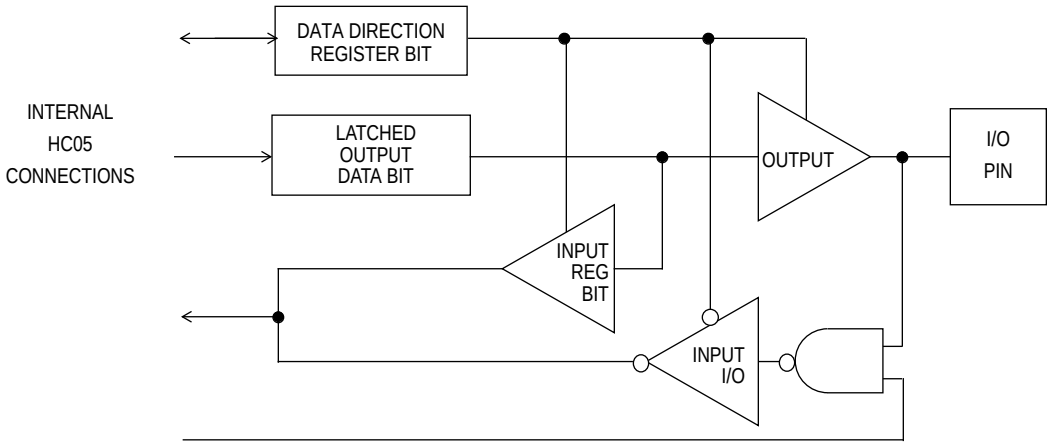


Figure 2-4. I/O Circuitry

SECTION 3
CENTRAL PROCESSING UNIT

This section describes the registers of the MC68HC05RC16 central processor unit (CPU). The MCU contains five registers as shown in Figure 3-1. The interrupt stacking order is shown in Figure 3-2.

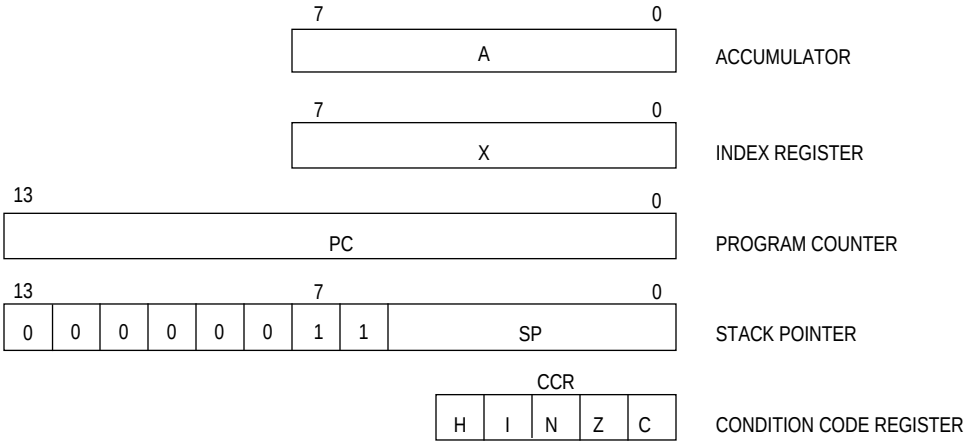
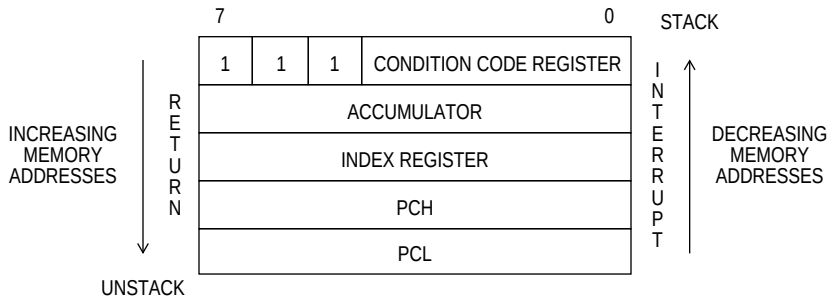


Figure 3-1. Programming Model

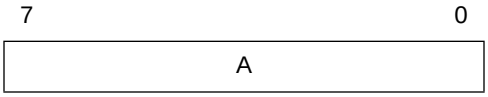


NOTE: Since the stack pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 3-2. Stacking Order

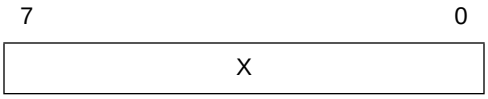
3.1 Accumulator (A)

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



3.2 Index Register (X)

The index register 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.



3.3 Condition Code Register (CCR)

The 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 or disable interrupts. These bits can be individually tested 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

3.3.1 Half Carry (H)

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

3.3.2 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.

3.3.3 Negative (N)

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

3.3.4 Zero (Z)

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

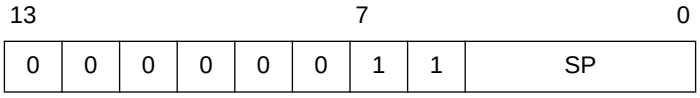
3.3.5 Carry/Borrow (C)

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 is also affected during bit test and branch instructions and during shifts and rotates.

3.4 Stack Pointer (SP)

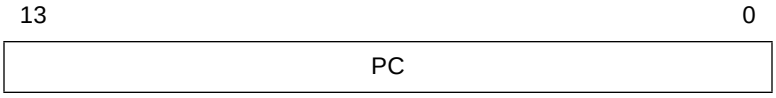
The stack pointer contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the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

When accessing memory, the eight most significant bits are permanently set to 00000011. These eight 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.



3.5 Program Counter (PC)

The program counter 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 a 16-Kbyte memory map.

SECTION 4
INTERRUPTS

The MCU can be interrupted four different ways:

1. Non-maskable Software Interrupt Instruction (SWI)
2. External Asynchronous Interrupt ($\overline{\text{IRQ}}$/Port B Keyscan)
3. Internal Carrier Modulator Transmitter Interrupt
4. Internal Core Timer Interrupt

4.1 CPU Interrupt Processing

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

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

When an interrupt is to be processed, the CPU fetches the address of the appropriate interrupt software service routine from the vector table at locations $3FF6 through $3FFF 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	$3FFE–$3FFF
N/A	N/A	Software Interrupt	SWI	$3FFC–$3FFD
N/A	N/A	External Interrupts*	IRQ	$3FFA–$3FFB
MCSR	EOC	End of Cycle Interrupt	CMT	$3FF8–$3FF9
CTCSR	CTOF, RTIF	Real Time Interrupt Core Timer Overflow	CORE TIMER	$3FF6–$3FF7

*External interrupts include $\overline{\text{IRQ}}$ and port B keyscan 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

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

4.2 Reset Interrupt Sequence

The reset function is not in the strictest sense an interrupt; however, it is acted upon in a similar manner as shown in Figure 4-1. A low-level input on the $\overline{\text{RESET}}$ pin, or an internally generated RST signal, causes the program to vector to its starting address, which is specified by the contents of memory locations $3FFE and $3FFF. The I bit in the condition code register is also set. The MCU is configured to a known state during this type of reset.

4.3 Software Interrupt (SWI)

The SWI is an executable instruction and a non-maskable interrupt since it is executed regardless of the state of the I bit in the CCR. If the I bit is zero (interrupts enabled), the SWI instruction executes after interrupts 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 $3FFC and $3FFD.

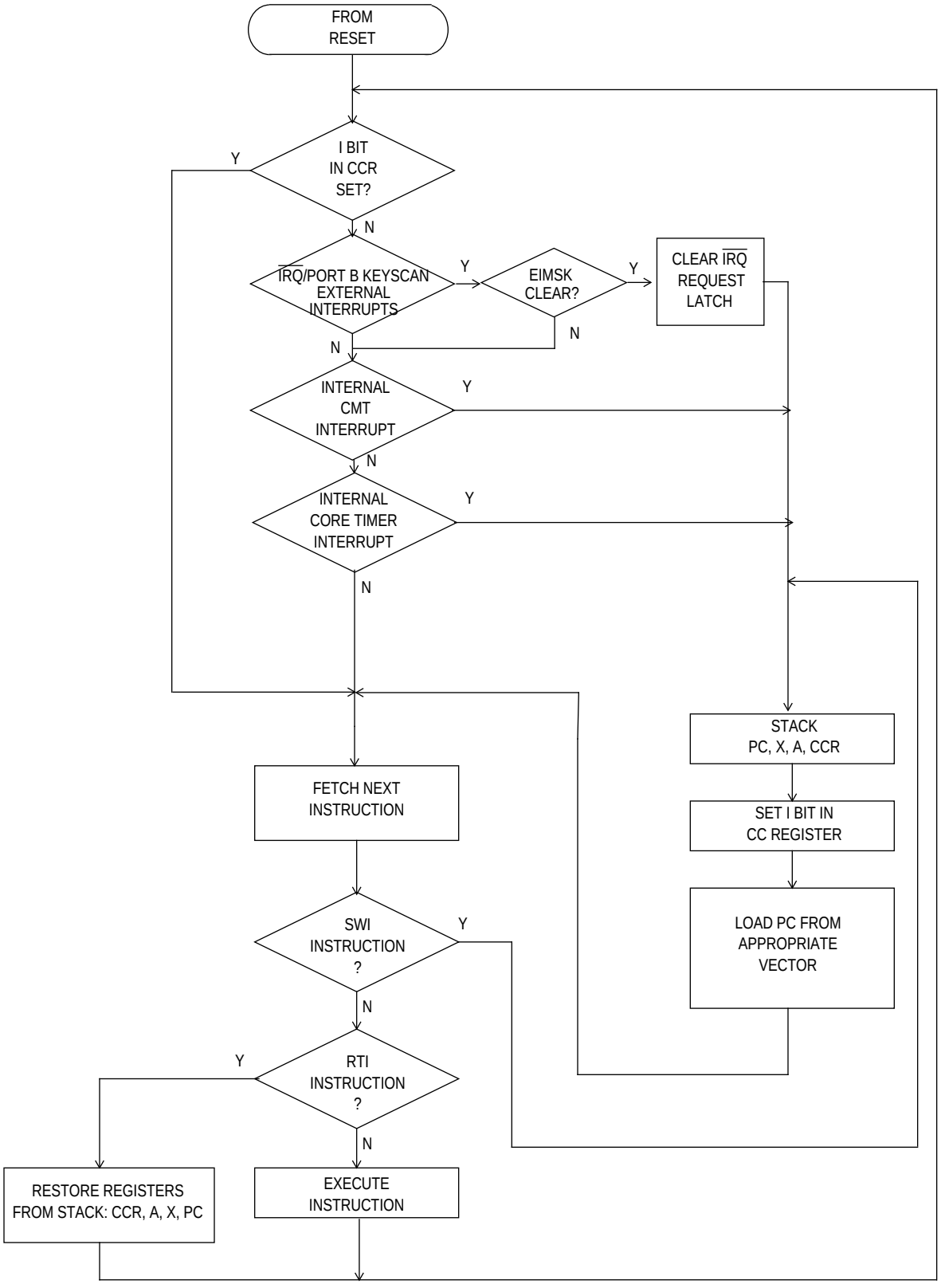


Figure 4-1. Interrupt Processing Flowchart

4.4 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. There are three types of hardware interrupts, which are explained in the following sections.

4.5 External Interrupt ($\overline{\text{IRQ}}$/Port B Keyscan)

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 to the level on the $\overline{\text{IRQ}}$ pin itself and to the output of the logic OR function with the port B IRQ interrupts. The states of the individual port B pins can be checked by reading the appropriate port B pins as inputs.

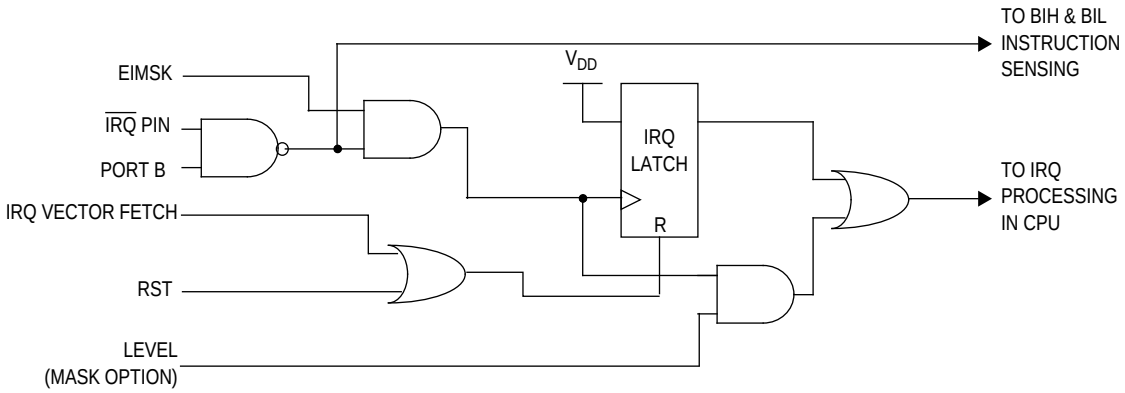


Figure 4-2. IRQ Function Block Diagram

The $\overline{\text{IRQ}}$ pin is one source of an external interrupt. All port B pins (PB0 through PB7) 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 the following cases:

1. Falling edge on the $\overline{\text{IRQ}}$ pin.
2. Falling edge on any port B pin with keyscan enabled.

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

1. Low level on the $\overline{\text{IRQ}}$ pin.
2. Falling edge on the $\overline{\text{IRQ}}$ pin.
3. Falling edge or low level on any port B pin with keyscan enabled.

External interrupts can also be masked by setting the EIMSK bit in the MSCR register of the IR remote timer. See **8.3.4 Modulator Period Data Registers (MDR1, MDR2, and MDR3)** for details.

4.6 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. It is then synchronized internally and serviced as specified by the contents of $3FFA and $3FFB.

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.

4.7 Carrier Modulator Transmitter Interrupt (CMT)

A CMT interrupt occurs when the end of cycle flag (EOC) and the end of cycle interrupt enable (EOCIE) bits are set in the modulator control and status register (MCSR). This interrupt will vector to the interrupt service routine located at the address specified by the contents of memory locations $3FF8 and $3FF9.

4.8 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 vectors to the same interrupt service routine, located at the address specified by the contents of memory locations $3FF6 and $3FF7.

SECTION 5
RESETS

The MCU can be reset from five sources: two external inputs and three internal restart conditions. The $\overline{\text{RESET}}$ and $\overline{\text{LPRST}}$ pins are inputs 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.

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

The $\overline{\text{RESET}}$ pin is one of the two external sources 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.

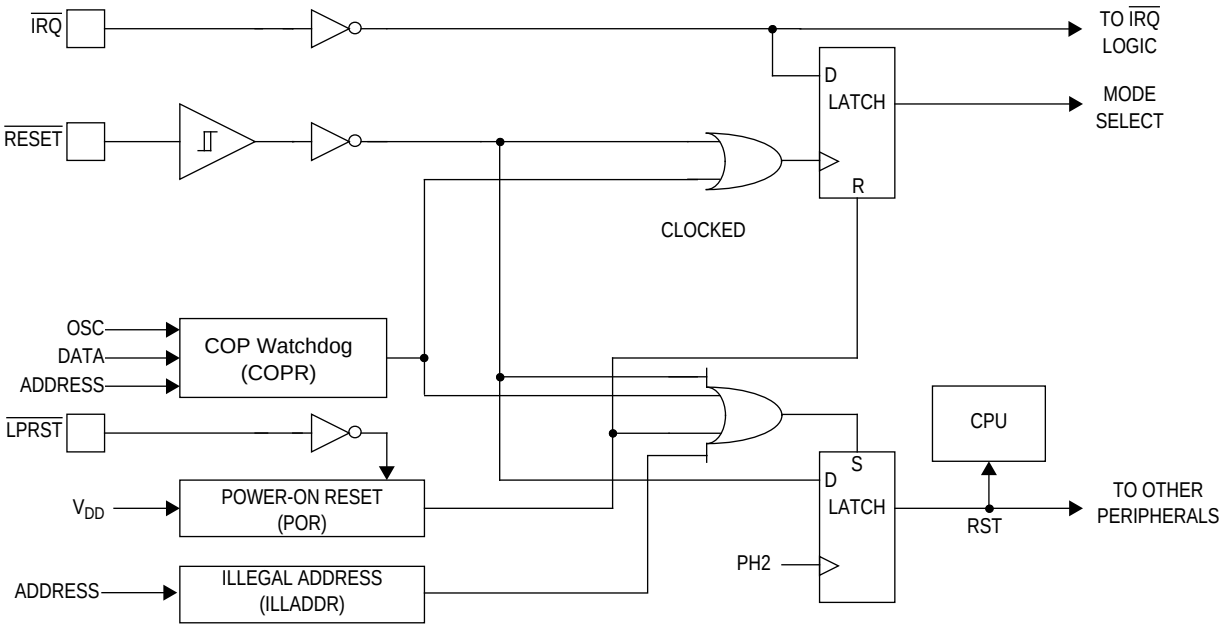


Figure 5-1. Reset Block Diagram

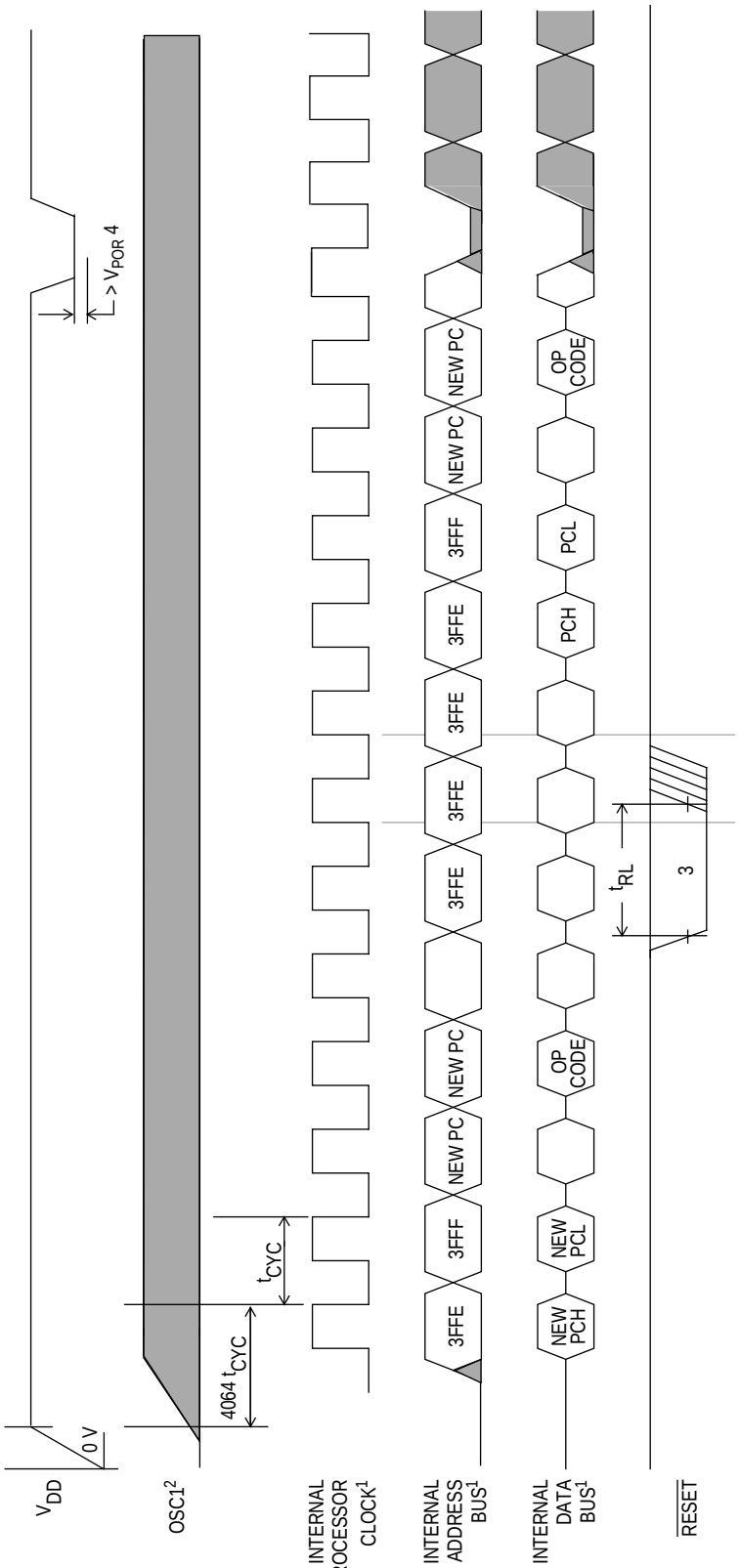


Figure 5-2. Reset and POR Timing Diagram

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.

NOTE

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

5.2 Low-Power External Reset ($\overline{\text{LPRST}}$)

The $\overline{\text{LPRST}}$ pin is one of the two external sources of a reset. This external reset occurs whenever the $\overline{\text{LPRST}}$ pin is pulled below the lower threshold and remains in reset until the $\overline{\text{LPRST}}$ pin rises. This active low input will, in addition to generating the RST signal and resetting the CPU and peripherals, halt all internal processor clocks. The MCU will remain in this low-power reset condition as long as a logic zero remains on LPRST. When a logic one is applied to LPRST, processor clocks will be re-enabled with the MCU remaining in reset until the 4064 internal processor clock cycle (t_{cyc}) oscillator stabilization delay is completed. 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.

5.3 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 reset input, external $\overline{\text{LPRST}}$ 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.

5.3.1 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) ends.

5.3.2 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.

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

5.3.2.1 Resetting the COP

Writing a zero to the COPF bit prevents a COP reset. This action resets the counter and begins the time-out period again. The COPF bit is bit 0 of address $3FF0. A read of address $3FF0 returns user data programmed at that location.

5.3.2.2 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.

5.3.2.3 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 which 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 does count these cycles. It is therefore recommended that the COP is fed before executing the STOP instruction.

5.3.2.4 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 intentionally or inadvertently due to the CPU being disturbed) causes the oscillator to halt and prevents 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 be **disabled** for a system that must have intentional uses of the wait mode for periods longer than the COP time-out 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: Wait Time	THEN the COP Watchdog Timer Should Be as Follows:
Wait Time **Less than** COP Time-Out	Enable or Disable COP by Mask Option
Wait Time **More than** COP Time-Out	Disable COP by Mask Option
Any Length Wait Time	Disable COP by Mask Option

5.3.2.5 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.

		Bit 7	6	5	4	3	2	1	Bit 0
$3FF0	Read:	X	X	X	X	X	X	X	X
	Write:								COPR
	Reset:	—	—	—	—	—	—	—	0

= Unimplemented

Figure 5-3. COP Watchdog Timer Location

5.3.3 Illegal Address

An illegal address reset is generated when the CPU attempts to fetch an instruction from I/O address space ($0000 to $001F).

SECTION 6
LOW-POWER MODES

This section describes the low-power modes.

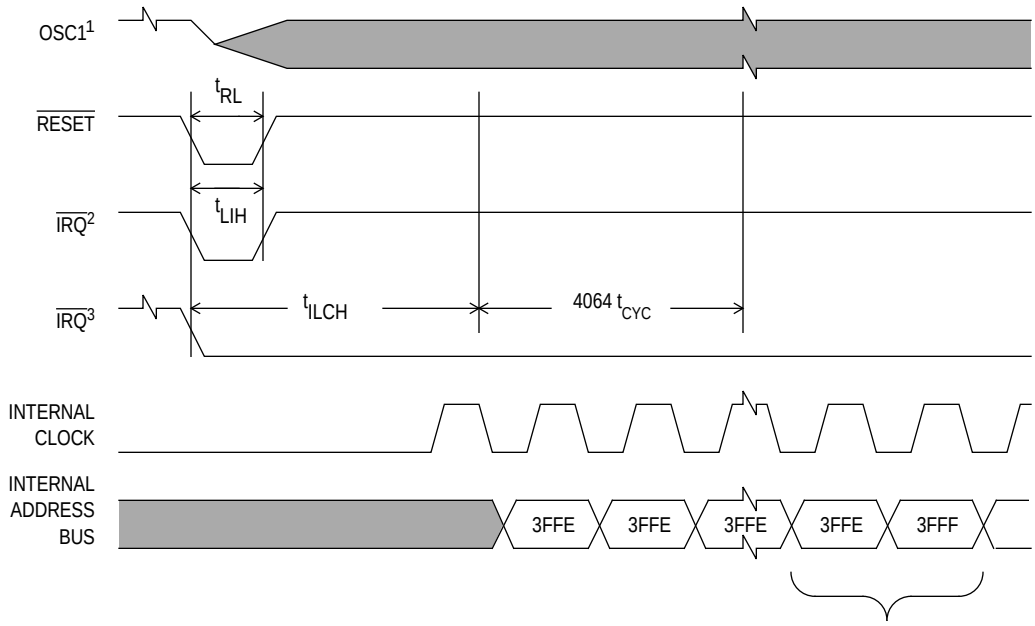
6.1 Stop Mode

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 the 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.

6.2 Stop Recovery

The processor can be brought out of the stop mode only by an external interrupt or $\overline{\text{RESET}}$. Refer to Figure 6-1.



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 6-1. Stop Recovery Timing Diagram

6.3 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.

During the 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 states. The timer may be enabled to allow a periodic exit from the wait mode.

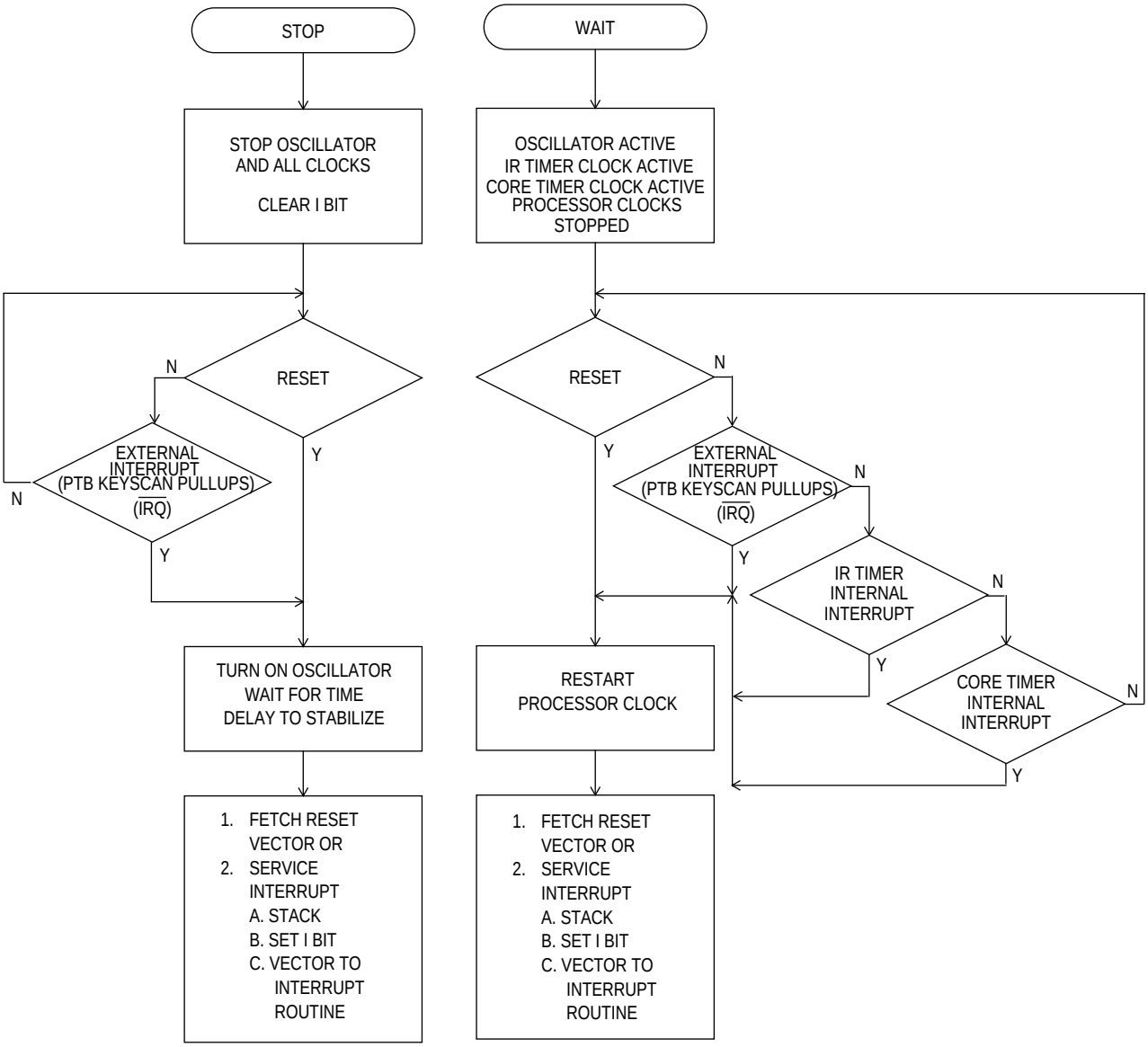


Figure 6-2. Stop/Wait Flowchart

NOTE

The EIMSK bit is not cleared automatically by the execution of a STOP instruction. Care should be taken to clear this bit before entering stop mode.

6.4 Low-Power Reset

Low-power reset mode is entered when a logic zero is detected on the $\overline{\text{LPRST}}$ pin. When in this mode (as long as $\overline{\text{LPRST}}$ is held low), the MCU is held in reset and all internal clocks are halted. Applying a logic one to $\overline{\text{LPRST}}$ will cause the part to exit low-power reset mode and begin counting out the 4064-cycle oscillator stabilization period. Once this time has elapsed, the MCU will begin operation from the reset vectors ($3FFE–$3FFF).

SECTION 7
CORE TIMER

The core timer for this device is a 14-stage multifunctional ripple counter. Features include timer overflow, power-on reset (POR), real-time interrupt (RTI), and COP watchdog timer.

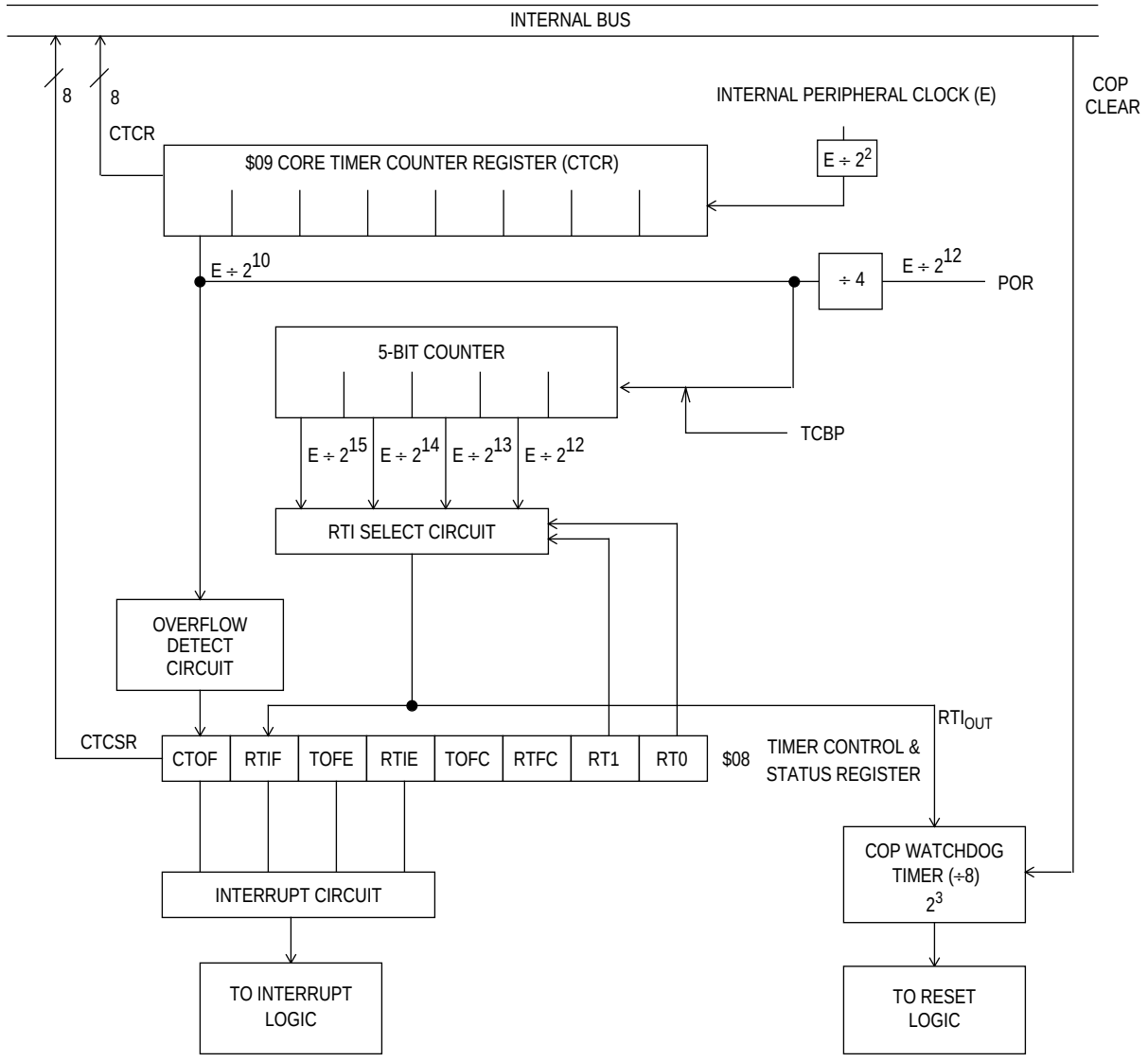


Figure 7-1. Core Timer Block Diagram

As seen in Figure 7-1, 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/4096) 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.

7.1 Core Timer Control and Status Register (CTCSR)

The CTCSR contains the timer interrupt flag, the timer interrupt enable bits, and the real-time interrupt rate select bits. Figure 7-2 shows the value of each bit in the CTCSR when coming out of reset.

		Bit 7	6	5	4	3	2	1	Bit 0
$08	Read:	CTOF	RTIF	TOFE	RTIE	0	0	RT1	RT0
	Write:					TOFC	RTFC		
	Reset:	0	0	0	0	0	0	1	1

= Unimplemented

Figure 7-2. Core Timer Control and Status Register

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 the CTOF is done by writing a one to TOFC. Writing to this bit has no effect. Reset clears CTOF.

RTIF — Real-Time Interrupt Flag

The real-time interrupt circuit consists of a three-stage divider and a one-of-four selector. The clock frequency that drives the RTI circuit is $E/2^{12}$ (or $E \div 4096$ with three additional divider stages giving a maximum interrupt period of 16 milliseconds at a bus rate of 2.024 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 the RTIF is done by writing a one to RTFC. Writing has no effect on this bit. 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 one is written to this bit, CTOF is cleared. Writing a zero has no effect on the CTOF bit. This bit always reads as zero.

RTFC — Real-Time Interrupt Flag Clear

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

RT1:RT0 — Real-Time Interrupt Rate Select

These two bits select one of four taps from the real-time interrupt circuit. Refer to Table 7-1. Reset sets these two bits which selects the lowest 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 time-out period is imminent or uncertain. If the selected tap is modified during a cycle in which the counter is switching, an RTIF could be missed or an additional one could be generated. To avoid problems, the COP should be cleared before changing RTI taps.

Table 7-1. RTI and COP Rates at 4.096 MHz Oscillator

RTI RATE 2.048 MHz Bus		RT1:RT0	MINIMUM COP RATES 2.048 MHz Bus	
2 ms	$2^{12} \div E$	00	$(2^{15}–2^{12})/E$	14 ms
4 ms	$2^{13} \div E$	01	$(2^{16}–2^{13})/E$	28 ms
8 ms	$2^{14} \div E$	10	$(2^{17}–2^{14})/E$	56 ms
16 ms	$2^{15} \div E$	11	$(2^{18}–2^{15})/E$	112 ms

7.2 Core Timer Counter Register (CTCR)

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.

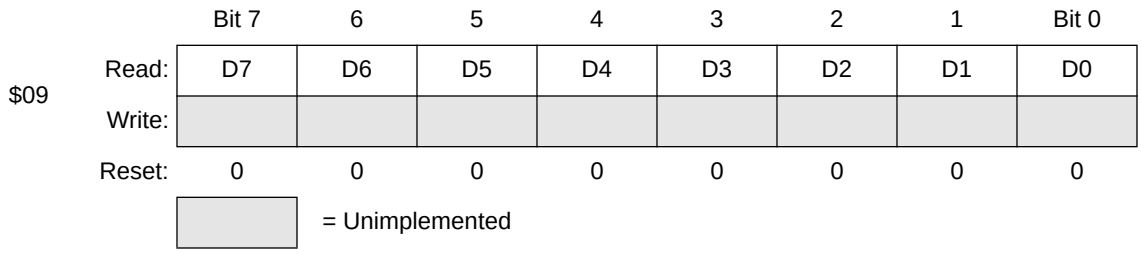


Figure 7-3. Timer Counter Register

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 and low-power reset), the counter chain is cleared.

7.3 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 Figure 7-1. If the COP circuit times out, an internal reset is generated and the normal reset vector is fetched. Preventing a COP time-out, or clearing the COP, is accomplished by writing a zero to bit 0 of address $3FF0. 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.

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 no operation (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.

7.4 Timer During Wait Mode

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

SECTION 8
CARRIER MODULATOR TRANSMITTER (CMT)

The carrier modulator transmitter (CMT) module provides a means to generate the protocol timing and carrier signals for a wide variety of encoding schemes. It incorporates hardware to off-load the critical and/or lengthy timing requirements associated with code generation from the CPU, releasing much of its bandwidth to handle other tasks such as code data generation, data decompression or keyboard scanning. The CMT does not include dedicated hardware configurations for specific protocols but is intended to be sufficiently programmable in its function to handle the timing requirements of most protocols with minimal CPU intervention. When disabled, certain CMT registers can be used to change the state of the infrared out pin (IRO) directly. This feature allows for the generation of future protocols not readily producible by the current architecture.

8.1 Overview

The module consists of carrier generator, modulator, and transmitter output blocks. The block diagram is shown in Figure 8-1.

The carrier generator has a resolution of 500 ns with a 2-MHz oscillator. The user may independently define the high and low times of the carrier signal to determine both period and duty cycle. The carrier generator can generate signals with periods between 1 μs (1 MHz) and 64 μs (15.6 kHz) in steps of 500 ns. The possible duty cycle options will depend upon the number of counts required to complete the carrier period. For example, a 400-kHz signal has a period of 2.5 μs and will therefore require 5 x 500 ns counts to generate. These counts may be split between high and low times so the duty cycles available will be 20% (one high, four low), 40% (two high, three low), 60% (three high, two low) and 80% (four high, one low). For lower frequency signals with larger periods, higher resolution (as a percentage of the total period) duty cycles are possible. The carrier generator may select between two sets of high and low times. When operating in normal mode (subsequently referred to as time mode), just one set will be used. When operating in FSK (frequency shift key) mode, the generator will toggle between the two sets when instructed to do so by the modulator, allowing the user to dynamically switch between two carrier frequencies without CPU intervention. When the BASE bit in the modulator control and status register (MCSR) is set, the carrier output to the modulator is held high continuously to allow for the generation of baseband protocols. See **8.2 Carrier Generator**.

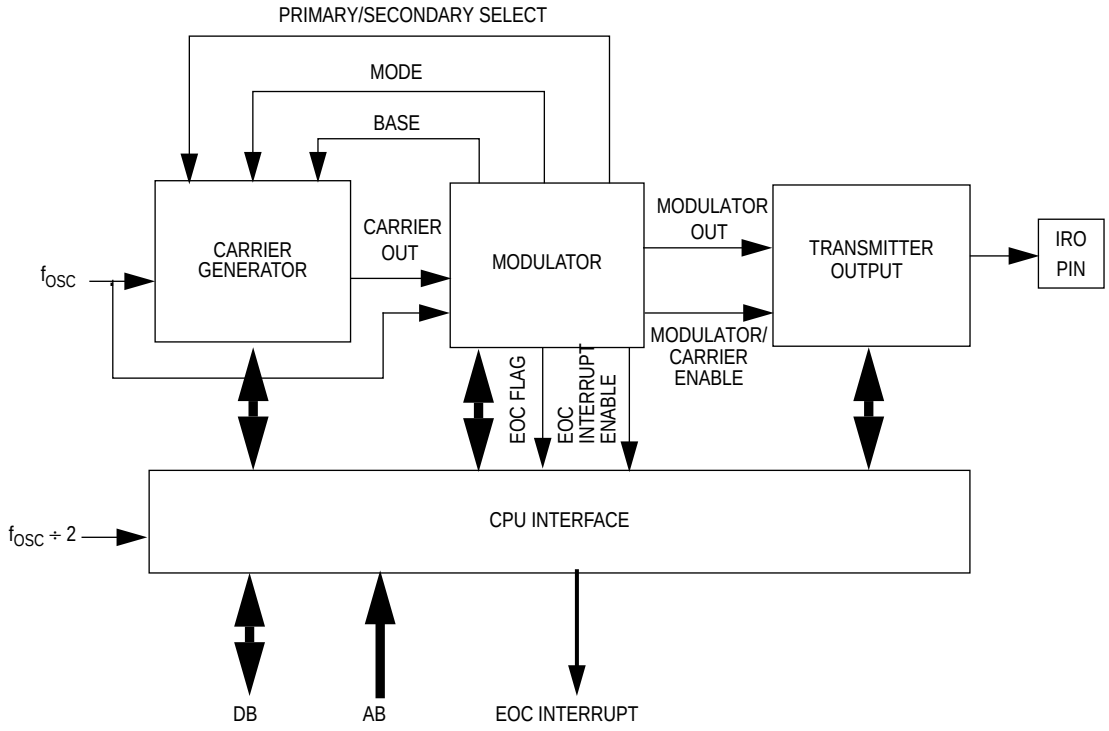


Figure 8-1. Carrier Modulator Transmitter Module Block Diagram

The modulator provides a simple method to control protocol timing. The modulator has a resolution of 4 μs with a 2-MHz oscillator. It can count system clocks to provide real-time control or it can count carrier clocks for self-clocked protocols. It can either gate the carrier onto the modulator output (TIME), control the logic level of the modulator output (baseband) or directly route the carrier to the modulator output while providing a signal to switch the carrier generator between high/low time register buffers (FSK). See **8.3 Modulator**.

The transmitter output block controls the state of the infrared out pin (IRO). The modulator output is gated on to the IRO pin when the modulator/carrier generator is enabled. Otherwise, the IRO pin is controlled by the state of the IRO latch, which is directly accessible to the CPU by means of bit 7 of the carrier generator data registers CHR1 and CLR1. The IRO latch can be written to on either edge of the internal bus clock ($f_{osc}/2$), allowing for IR waveforms which have a resolution of twice the bus clock frequency (f_{osc}). See **8.2.2 Carrier Generator Data Registers (CHR1, CLR1, CHR2, and CLR2)**.

8.2 Carrier Generator

The carrier signal is generated by counting a predetermined number of input clocks (500 ns for a 2-MHz oscillator) for both the carrier high time and the carrier low time. The period is determined by the total number of clocks counted. The duty cycle is determined by the ratio of high time clocks to total clocks counted. The high and low time values are user programmable and are held in two registers. An alternate set of high/low count values is held in another set of registers to allow the generation of dual frequency FSK (frequency shift keying) protocols without CPU intervention. The MCGEN bit in the MCSR must be set and the BASE bit in the MCSR must be cleared to enable carrier generator clocks. The block diagram is shown in Figure 8-2.

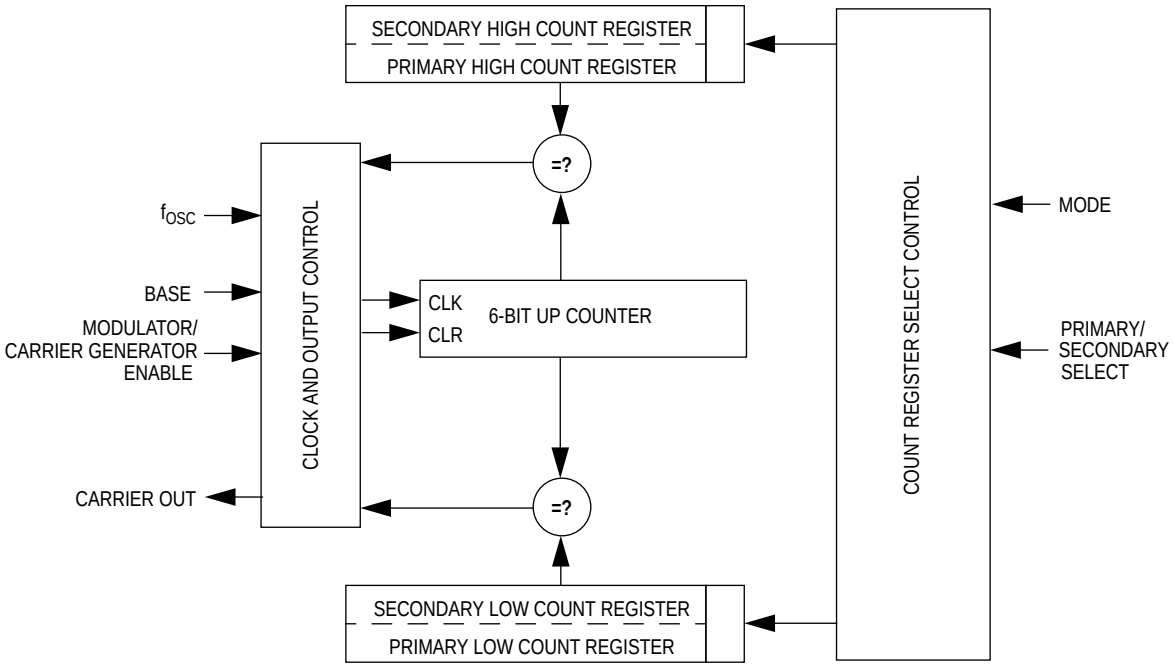


Figure 8-2. Carrier Generator Block Diagram

8.2.1 Time Counter

The high/low time counter is a 6-bit up counter. After each increment, the contents of the counter are compared with the appropriate high or low count value register. When this value is reached, the counter is reset and the compare is redirected to the other count value register. Assuming that the high time count compare register is currently active, a valid compare will cause the carrier output to be driven low. The counter will continue to increment and when reaching the value stored in the selected low count value register, it will be cleared and will cause the carrier output to be driven high. The cycle repeats, automatically generating a periodic signal

which is directed to the modulator. The lowest frequency (maximum period) and highest frequency (minimum period) which can be generated are defined below.

$$f_{min} = f_{osc} \div (2 \times (2^6 - 1)) \text{ Hz}$$

$$f_{max} = f_{osc} \div (2 \times 1) \text{ Hz}$$

In the general case, the carrier generator output frequency is:

$$f_{out} = f_{osc} \div (\text{Highcount} + \text{Lowcount}) \text{ Hz}$$

Where: $0 < \textit{Highcount} < 64$ and
$0 < \textit{Lowcount} < 64$

NOTE

These equations assume the DIV2 bit (bit 6) of the MCSR is clear. When the DIV2 bit is set, the carrier generator frequency will be half of what is shown in these equations.

The duty cycle of the carrier signal is controlled by varying the ratio of high time to low + high time. As the input clock period is fixed, the duty cycle resolution will be proportional to the number of counts required to generate the desired carrier period.

$$\text{Duty Cycle} = \frac{\text{Highcount}}{\text{Highcount} + \text{Lowcount}}$$

8.2.2 Carrier Generator Data Registers (CHR1, CLR1, CHR2, and CLR2)

The carrier generator contains two, 7-bit data registers: primary high time (CHR1), primary low time (CLR1); and two, 6-bit data registers: secondary high time (CHR2) and secondary low time (CLR2). Bit 7 of CHR1 and CHR2 is used to read and write the IRO latch.

		Bit 7	6	5	4	3	2	1	Bit 0
CHR1 $10	Read: / Write:	IROLN	0	PH5	PH4	PH3	PH2	PH1	PH0
	Reset:	0	0	U	U	U	U	U	U

U = Unaffected

		Bit 7	6	5	4	3	2	1	Bit 0
CLR1 $11	Read: / Write:	IROLP	0	PL5	PL4	PL3	PL2	PL1	PL0
	Reset:	0	0	U	U	U	U	U	U

U = Unaffected

		Bit 7	6	5	4	3	2	1	Bit 0
CHR2 $12	Read: / Write:	0	0	SH5	SH4	SH3	SH2	SH1	SH0
	Reset:	0	0	U	U	U	U	U	U

U = Unaffected

		Bit 7	6	5	4	3	2	1	Bit 0
CLR2 $13	Read: / Write:	0	0	SL5	SL4	SL3	SL2	SL1	SL0
	Reset:	0	0	U	U	U	U	U	U

U = Unaffected

Figure 8-3. Carrier Data Registers

PH0-PH5 and PL0-PL5 — Primary Carrier High and Low Time Data Values

When selected, these bits contain the number of input clocks required to generate the carrier high and low time periods. When operating in time mode (see **8.3.1 Time Mode**), this register pair is always selected. When operating in FSK mode (see **8.3.2 FSK Mode**), this register pair and the secondary register pair are alternately selected under control of the modulator. The primary carrier high and low time values are undefined out of reset. These bits must be written to non-zero values before the carrier generator is enabled to avoid spurious results.

NOTE

Writing to CHR1 to update PH0–PH5 or to CLR1 to update PL0–PL5 will also update the IRO latch. When MCGEN (bit 0 in the MCSR) is clear, the IRO latch value appears on the IRO output pin. Care should be taken that bit 7 of the data to be written to CHR1 or CHL1 should contain the desired state of the IRO latch.

SH0-SH5 and SL0-SL5 — Secondary Carrier High and Low Time Data Values

When selected, these bits contain the number of input clocks required to generate the carrier high and low time periods. When operating in time mode (see **8.3.1 Time Mode**), this register pair is never selected. When operating in FSK mode (see **8.3.2 FSK Mode**), this register pair and the secondary register pair are alternately selected under control of the modulator. The secondary carrier high and low time values are undefined out of reset. These bits must be written to non-zero values before the carrier generator is enabled when operating in FSK mode.

IROLN and IROLP — IRO Latch Control

Reading IROLN or IROLP reads the state of the IRO latch. Writing IROLN updates the IRO latch with the data being written on the negative edge of the internal processor clock ($f_{osc}/2$). Writing IROLP updates the IRO latch on the positive edge of the internal processor clock; for example, one f_{osc} period later. The IRO latch is clear out of reset.

NOTE

Writing to CHR1 to update IROLN or to CLR1 to update IROLP will also update the primary carrier high and low data values. Care should be taken that bits 5–0 of the data to be written to CHR1 or CHL1 should contain the desired values for the primary carrier high or low data.

8.3 Modulator

The modulator consists of a 12-bit down counter with underflow detection which is loaded from the modulation mark period from the mark buffer register, MBUFF. When this counter underflows, the modulator gate is closed and a 12-bit comparator is enabled which continually compares the logical complement of the contents of the (still) decrementing counter with the contents of the modulation space period register, SREG. When a match is obtained, the modulator control gate is opened again. Should SREG = 0, the match will be immediate and no space period will be generated (for instance, for FSK protocols which require successive bursts of different frequencies). When the match occurs, the counter is reloaded with the contents of MBUFF, SREG is reloaded with the contents of its buffer, SBUFF, and the cycle repeats. The MCGEN bit in the MCSR must be set to enable the modulator timer. The 12-bit MBUFF and SBUFF registers are accessed through three 8-bit modulator period registers, MDR1, MDR2, and MDR3.

The modulator can operate in two modes, time or FSK. In time mode the modulator counts clocks derived from the system oscillator and modulates a single-carrier frequency or no carrier (baseband). In FSK mode, the modulator counts carrier periods and instructs the carrier generator to alternate between two carrier frequencies whenever a modulation period (mark + space counts) expires.

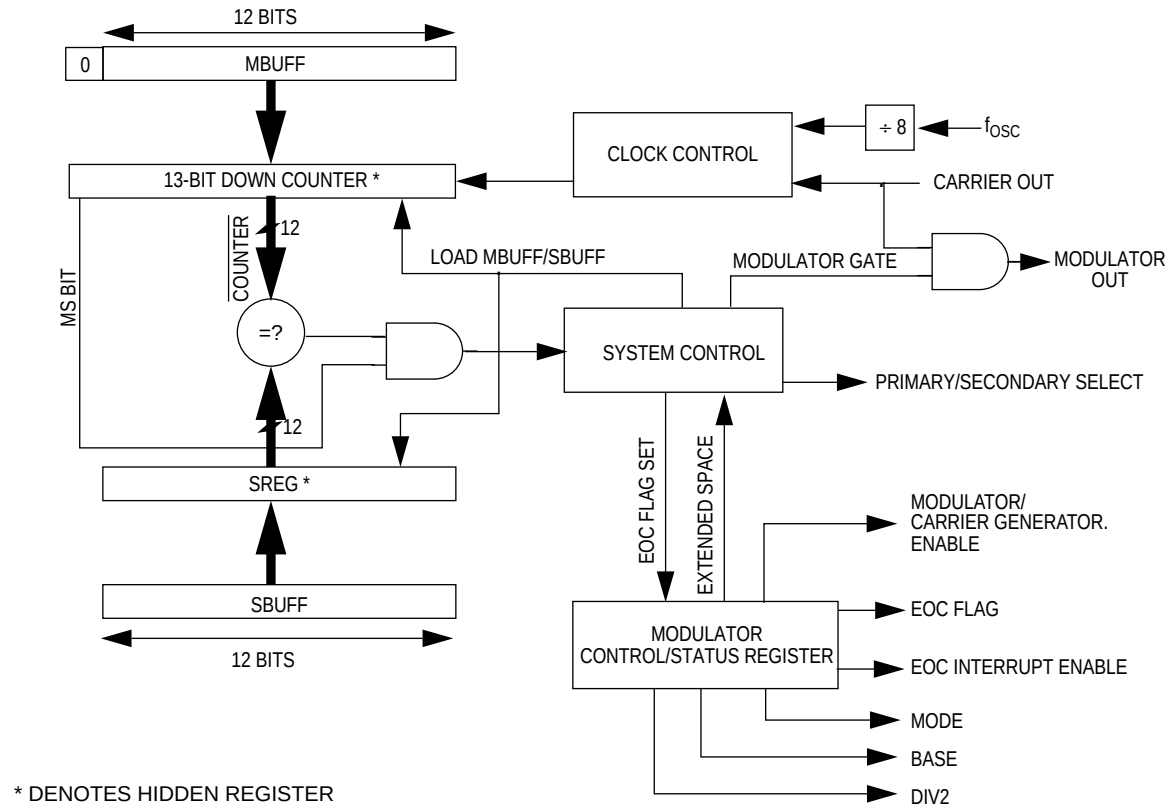


Figure 8-4. Modulator Block Diagram

8.3.1 Time Mode

When the modulator operates in time mode, the modulation mark and space periods consist of zero or an integer number of $f_{osc} \div 8$ clocks (= 250 kHz @ 2 MHz osc). This provides a modulator resolution of 4 μs and a maximum mark and space periods of about 16 ms (each). However, to prevent carrier glitches which could affect carrier spectral purity, the modulator control gate and carrier clock are synchronized. The carrier signal is activated when the modulator gate opens. The modulator gate can only close when the carrier signal is low (the output logic level during space periods is low). If the carrier generator is in baseband mode (BASE bit in MCSR is high), the modulator output will be at a logic one for the duration of the mark period and at a logic zero for the duration of a space period. See Figure 8-5.

The mark and space time equations are:

$$t_{mark} = \frac{(MBUFF + 1) \times 8}{f_{osc}} secs$$

$$t_{space} = \frac{SBUFF \times 8}{f_{osc}} secs$$

Setting the DIV2 bit in the MCSR will double mark and space times.

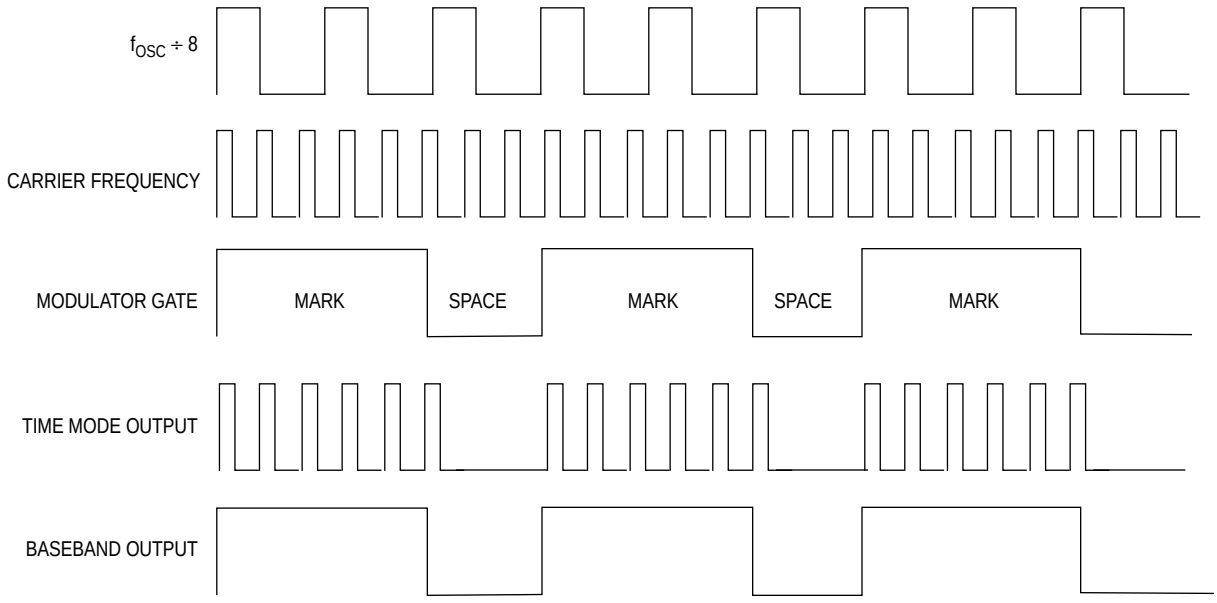


Figure 8-5. CMT Operation in Time Mode

8.3.2 FSK Mode

When the modulator operates in FSK mode, the modulation mark and space periods consist of an integer number of carrier clocks (space period can be zero). When the mark period expires, the space period is transparently started (as in time mode); however, in FSK mode the carrier switches between data registers in preparation for the next mark period. The carrier generator toggles between primary and secondary data register values whenever the modulator mark period expires. The space period provides an interpulse gap (no carrier), but if SBUFF = 0, then the modulator and carrier generator will switch between carrier frequencies without a gap or any carrier glitches (zero space).

Using timing data for carrier burst and interpulse gap length calculated by the CPU, FSK mode can automatically generate a phase-coherent, dual-frequency FSK signal with programmable burst and interburst gaps.

The mark and space time equations for FSK mode are:

$$t_{mark} = \frac{MBUFF + 1}{f_{cg}} secs$$

$$t_{space} = \frac{SBUFF}{f_{cg}} secs$$

Where f_{cg} is the frequency output from the carrier generator, setting the DIV2 bit in the MCSR will double mark and space times.

8.3.3 Extended Space Operation

In either time or FSK mode, the space period can be made longer than the maximum possible value of SBUFF. Setting the EXSPC bit in the MCSR will force the modulator to treat the next modulation period (beginning with the next load of MBUFF/SBUFF) as a space period equal in length to the mark and space counts combined. Subsequent modulation periods will consist entirely of these extended space periods with no mark periods. Clearing EXSPC will return the modulator to standard operation at the beginning of the next modulation period. To calculate the length of an extended space in time mode, use the equation:

$$t_{exspace} = \frac{((SBUFF_1)+(MBUFF_2+1+SBUFF_2) + \ldots (MBUFF_n+1+SBUFF_n)) \times 8}{f_{osc}} \text{ secs}$$

Where the subscripts 1, 2, ... n refer to the modulation periods that elapsed while the EXSPC bit was set.

Similarly, to calculate the length of an extended space in FSK mode, use the equation:

$$t_{exspace} = \frac{((SBUFF_1)+(MBUFF_2+1+SBUFF_2)+\ldots\ (MBUFF_n+1+SBUFF_n))}{\mathbf{f_{cg}}}\ \textbf{secs}$$

Where $\mathbf{f_{cg}}$ is the frequency output from the carrier generator. For an example of extended space operation, see Figure 8-6.

NOTE

The EXSPC feature can be used to emulate a zero mark event.

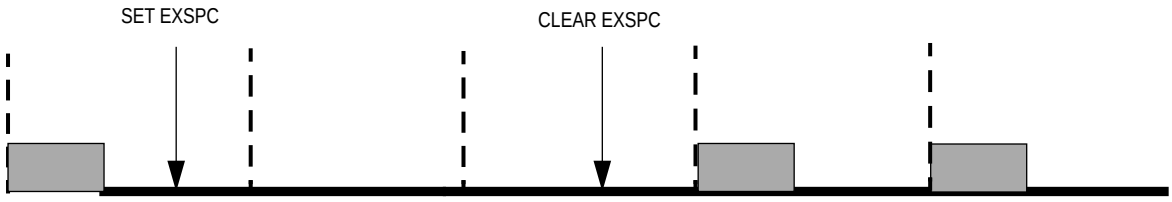


Figure 8-6. Extended Space Operation

8.3.3.1 End Of Cycle (EOC) Interrupt

At the end of each cycle (when the counter is reloaded from MBUFF), the end of cycle (EOC) flag is set. If the interrupt enable bit was previously set, an interrupt will also be issued to the CPU. The EOC interrupt provides a means for the user to reload new mark/space values into the MBUFF and SBUFF registers. As the EOC interrupt is coincident with reloading the counter, MBUFF does not require additional buffering and may be updated with a new value for the next period from within the EOC interrupt service routine (ISR). To allow both mark and space period values to be updated from within the same ISR, SREG is buffered by SBUFF. The contents written to SBUFF are transferred to the active register SREG at the end of every cycle irrespective of the state of the EOC flag. The EOC flag is cleared by a read of the modulator control and status register (MCSR) followed by an access of MDR2 or MDR3. The EOC flag must be cleared within the ISR to prevent another interrupt being generated after exiting the ISR. If the EOC interrupt is not being used (IE = 0), the EOC flag need not be cleared.

8.3.3.2 Modulator Control and Status Register (MCSR)

The modulator control and status register (MCSR) contains the modulator and carrier generator enable (MCGEN), interrupt enable (IE), mode select (MODE), baseband enable (BASE), extended space (EXSPC), and external interrupt mask (EIMSK) control bits, divide-by-two prescaler (DIV2) bit, and the end of cycle (EOC) status bit.

		Bit 7	6	5	4	3	2	1	Bit 0
MCSR $14	Read:	EOC	DIV2	EIMSK	EXSPC	BASE	MODE	IE	MCGEN
	Write:								
	Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 8-7. Modulator Control and Status Register (MCSR)

EOC — End Of Cycle Status Flag

1 = End of modulator cycle (counter = SBUFF) has occurred
0 = Current modulation cycle in progress

EOC is set when a match occurs between the contents of the space period register, SREG, and the down counter. This is recognized as the end of the modulation cycle. At this time, the counter is initialized with the (possibly new) contents of the mark period buffer, MBUFF, and the space period register, SREG, is loaded with the (possibly new) contents of the space period buffer, SBUFF. This flag is cleared by a read of the MCSR followed by an access of MDR2 or MDR3. The EOC flag is cleared by reset.

DIV2 — Divide-by-two prescaler

1 = Divide-by-two prescaler enabled
0 = Divide-by-two prescaler disabled

The divide-by-two prescaler causes the CMT to be clocked at the bus rate when enabled; 2 x the bus rate when disabled (f_{osc}). This bit is not double buffered and so should not be set during a transmission.

EIMSK — External Interrupt Mask

1 = IRQ and keyscan interrupts masked
0 = IRQ and keyscan interrupts enabled

The external interrupt mask bit is used to mask IRQ and keyscan interrupts. This bit is cleared by reset.

EXSPC — Extended Space Enable

1 = Extended space enabled
0 = Extended space disabled

For a description of the extended space enable bit, see **8.3.3 Extended Space Operation**. This bit is cleared by reset.

BASE — Baseband Enable

1 = Baseband enabled
0 = Baseband disabled

When set, the BASE bit disables the carrier generator and forces the carrier output high for generation of baseband protocols. When BASE is clear, the carrier generator is enabled and the carrier output toggles at the frequency determined by values stored in the carrier data registers. See **8.3.1 Time Mode**. This bit is cleared by reset. This bit is not double buffered and should not be written to during a transmission.

MODE — Mode Select

1 = CMT operates in FSK mode.
0 = CMT operates in Time mode.

For a description of CMT operation in time mode, see **8.3.1 Time Mode**. For a description of CMT operation in FSK mode, see **8.3.2 FSK Mode**. This bit is cleared by reset. This bit is not double buffered and should not be written to during a transmission.

IE — Interrupt Enable

1 = CPU interrupt enabled
0 = CPU interrupt disabled

A CPU interrupt will be requested when EOC is set if IE was previously set. If IE is clear, EOC will not request a CPU interrupt.

MCGEN — Modulator and Carrier Generator Enable

1 = Modulator and carrier generator enabled
0 = Modulator and carrier generator disabled

Setting MCGEN will initialize the carrier generator and modulator and will enable all clocks. Once enabled, the carrier generator and modulator will function continuously. When MCGEN is cleared, the current modulator cycle will be allowed to expire before all carrier and modulator clocks are disabled (to save power) and the modulator output is forced low. The user should initialize all data and control registers before enabling the system to prevent spurious operation. This bit is cleared by reset.

8.3.4 Modulator Period Data Registers (MDR1, MDR2, and MDR3)

The 12-bit MBUFF and SBUFF registers are accessed through three 8-bit registers, MDR1, MDR2, and MDR3. MDR2 and MDR3 contain the least significant eight bits of MBUFF and SBUFF respectively. MDR1 contains the two most significant nibbles of MBUFF and SBUFF. In many applications, periods greater than those obtained by eight bits will not be required. Splitting the registers up in this manner allows the user to clear MDR1 and generate 8-bit periods with just two data writes.

		Bit 7	6	5	4	3	2	1	Bit 0
MDR1 $15	Read: / Write:	MB11	MB10	MB9	MB8	SB11	SB10	SB9	SB8
	Reset:	U	U	U	U	U	U	U	U

U = Unaffected

		Bit 7	6	5	4	3	2	1	Bit 0
MDR2 $16	Read: / Write:	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
	Reset:	U	U	U	U	U	U	U	U

U = Unaffected

		Bit 7	6	5	4	3	2	1	Bit 0
MDR3 $17	Read: / Write:	SB7	SB6	SB5	SB4	SB3	SB2	SB1	SB0
	Reset:	U	U	U	U	U	U	U	U

U = Unaffected

Figure 8-8. Modulator Data Registers (MDR1, MDR2, and MDR3)

SECTION 9
MC68HC705RC16

The following section describes user mode differences between the MC68HC705RC16 and the MC68HC05RC16.

The MC68HC705RC16 is a low-cost addition to the M68HC05 Family of microcontrollers (MCUs) and is suitable for remote control applications. It contains the HC05 CPU core, including the 14-stage core timer with RTI and COP watchdog systems. On-chip peripherals include a carrier modulator transmitter. The 16 Kbyte memory map has 15,936 bytes of user EPROM, 340 bytes of boot ROM, and 352 bytes of RAM. There are 20 I/O lines (8 having keyscan logic and pullups) and a low-power reset pin. The MC68HC705RC16 is available in 28 pin SOIC or DIP packages. There are four additional I/O lines available for bond out in higher pin count packages.

9.1 Mask Options

The mask options on the MC68HC705RC16 are handled with 11 EPROM bits in two separate MOR registers (MOR1 and MOR2). These options are:

- Eight Port B Pullups
- IRQ Sensitivity, **Section 1.3.2IRQ (Maskable Interrupt Request)**
- STOP Enable/Disable, and
- COP Enable/Disable

ROM versions of this device will have these options programmed by the factory.

9.2 $\overline{IRQ}/V_{PP}$ (Maskable Interrupt Request)

In addition to the functions described in **1.3.2 IRQ (Maskable Interrupt Request)**, this pin is also used to supply the EPROM with the required programming voltage.

9.3 Memory

The MC68HC705RC16 has a 16-Kbyte memory map consisting of user EPROM, RAM, bootloader ROM, and I/O.

9.3.1 EPROM

The user EPROM consists of 15,936 bytes of EPROM from $0180 to $3FAF and 14 bytes of user vectors from $3FF2 to $3FFF.

The bootloader ROM and vectors are located from $3FB0 to $3FEF.

Ten of the user vectors, $3FF6 thorough $3FFF, are dedicated to reset and interrupt vectors. The four remaining locations, $3FF2–$3FF5 are general-purpose user EPROM locations. The mask option registers (MOR1 and MOR2) are located at $3FF0 and $3FF1.

9.3.2 EPROM Memory Map

Figure 9-1 shows the memory map in user mode.

9.4 EPROM Security

The MC68HC705RC16 contains special circuitry to prevent accessing the EPROM in non-user mode. Emulation is not affected by security.

Security is controlled by a security bit in the MOR1 register. It is intended to be programmed while the user is programming their code. When set, this will inhibit reading of the EPROM in all modes other than user mode.

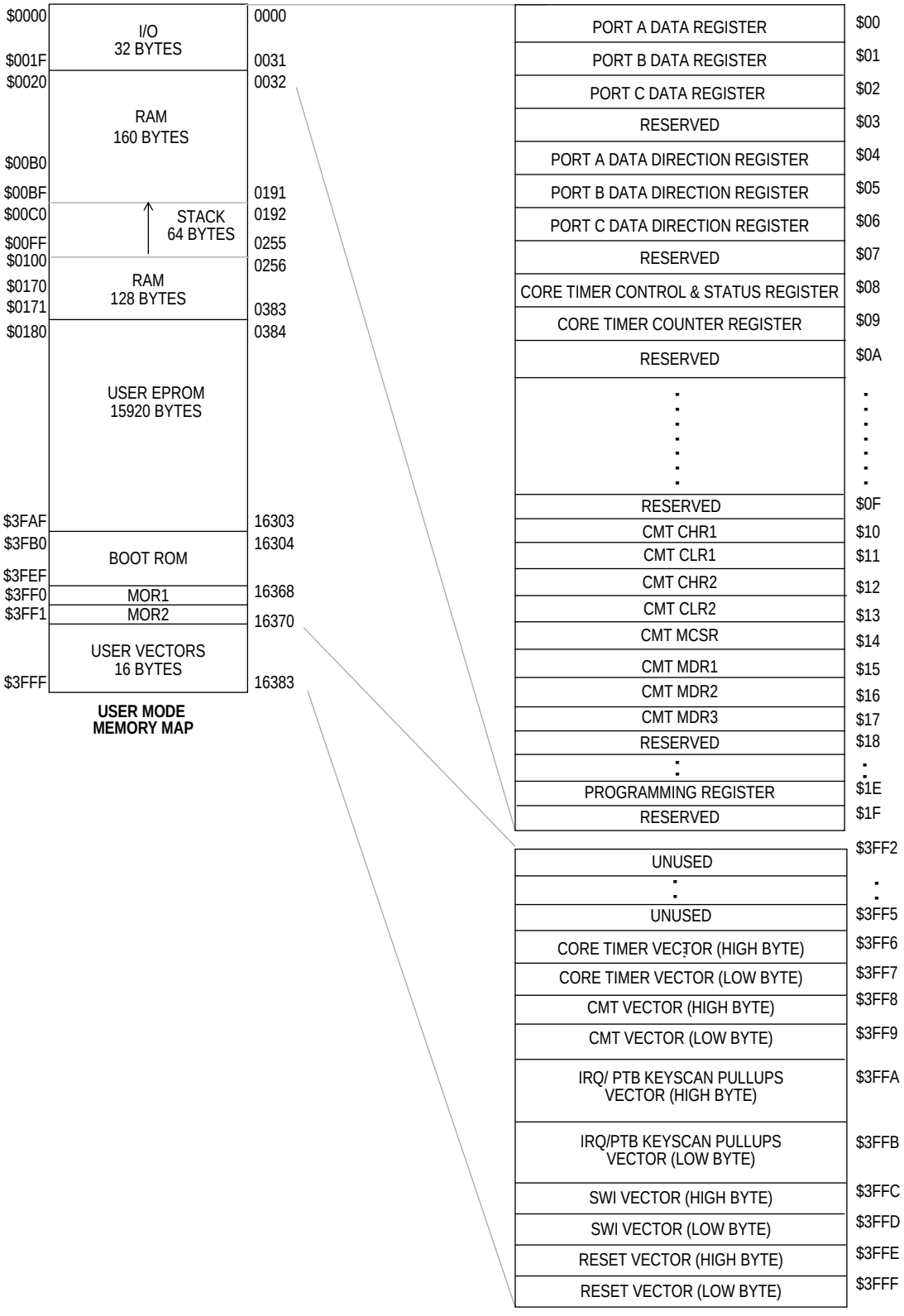


Figure 9-1. MC68HC705RC16 16K Memory Map

9.5 Bootloader

This program (contained in an on-chip BOOTROM) handles copying of user code from an external EPROM into the on-chip EPROM. The bootload function does not have to be done from an external EPROM, it may be done from a host.

9.5.1 Bootloader Functions

Two pins are used to select the bootloader function. These pins are PC1 and PB5. PC1 is normally a SYNC pin, which is used to synchronize the MCU to an off-chip source driving EPROM data into the MCU. The programmer/host interface is shown in Figure 9-2.

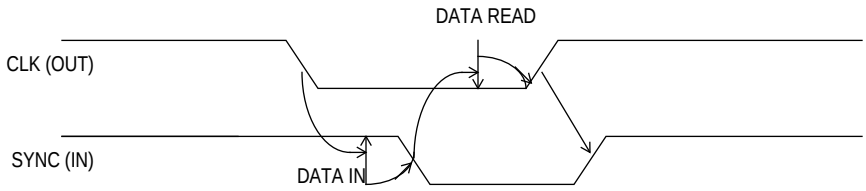


Figure 9-2. Programmer Interface to Host

If an external EPROM is used, this pin (PC1) must be connected to V_{SS}. PB5 is used to select between program/verify or verify only modes. Two other pins, PB2 and IRO, are used to drive the VERF LED and the PROG LED respectively. The programming modes are shown in Table 9-1.

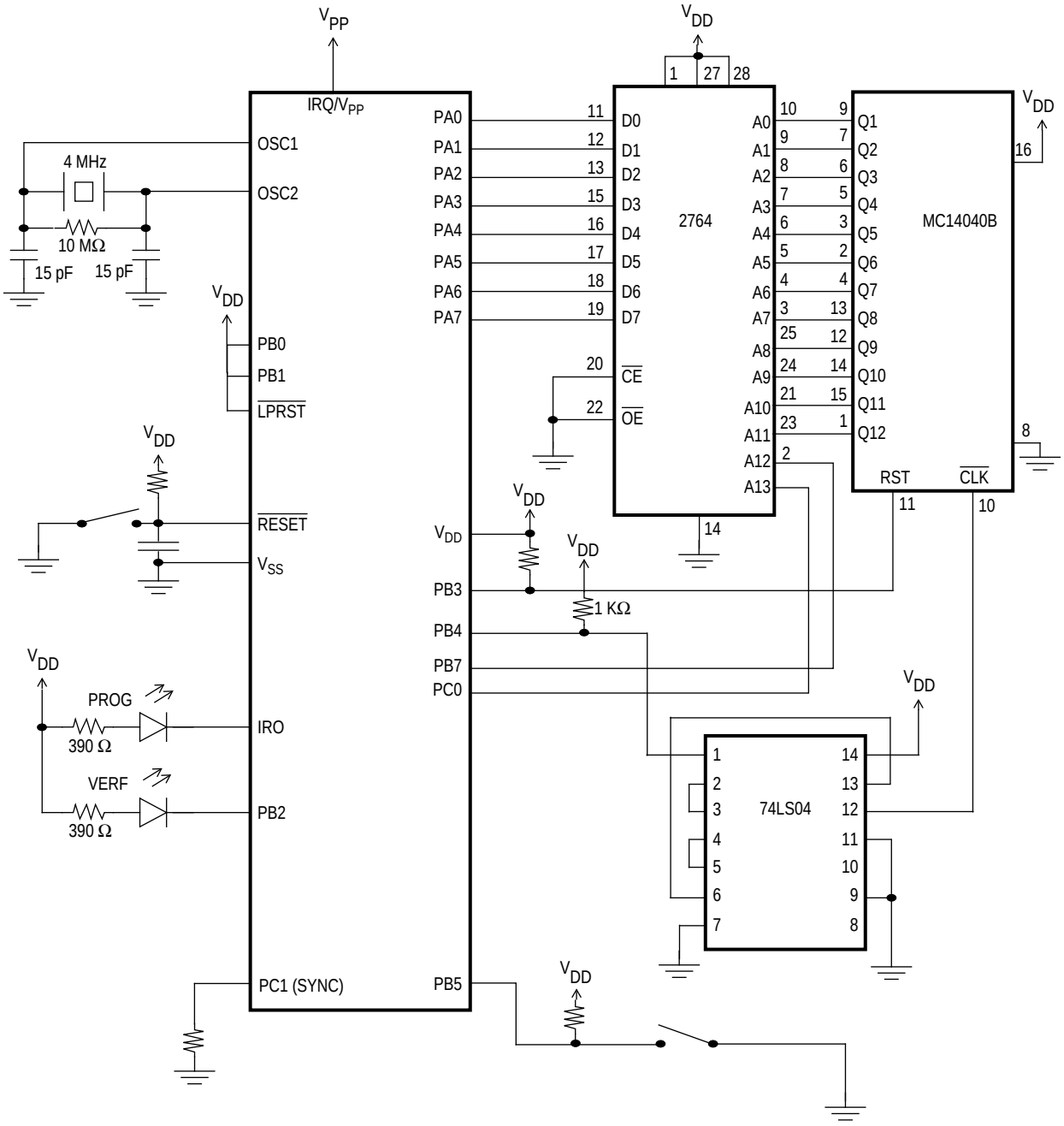
Table 9-1. Bootloader Functions

PC1	PB5	Mode
SYNC	1	Program/Verify
SYNC	0	Verify Only

The bootloader programming board shown in Figure 9-3 uses an external 12-bit counter to address the memory device containing the code to be copied. This counter requires a clock and a reset function. The 12-bit counter can address up to 4 Kbytes of memory, which means that two port pins have to be used to address the extra memory space.

NOTE

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



All Resistors are 10 KΩ unless specified otherwise

Figure 9-3. MC68HC705RC16 Programming Circuit

9.5.2 Programming Register (PROG)

This register is used to program the EPROM array. Only the LATCH and EPGM bits are available in User mode. To program a byte of EPROM, set LATCH, then write data to the desired address, then set EPGM for $t_{EPGM.}$

	7	6	5	4	3	2	1	0
$1E	R	R	R	R	R	LATCH	R	EPGM
RESET:	0	0	0	0	0	0	0	0

R = Reserved for Test

Figure 9-4. Programming Register

LATCH — EPROM Latch Control

READ: Any time.

WRITE: Any time.

1 = EPROM address and data bus configured for programming. Causes address and data bus to be latched when a write to EPROM is done. EPROM cannot be read if LATCH = 1.

0 = EPROM address and data bus configured for normal reads.

EPGM — EPROM Program Control

READ: Any time

WRITE: Any time security is not set.

1 = V_{PP} switched on to the EPROM array. If LATCH = 1, EPGM switches programming power to the EPROM array.

0 = Programming power switched off the EPROM array.

9.5.3 Mask Option Registers (MOR1 and MOR2)

The mask option registers contain programmable EPROM bits to control mask options. The MOR register is latched upon reset going away and refreshed periodically defined by how often the EPROM is read.

	7	6	5	4	3	2	1	0
MOR1 $3FF0	U	U	U	U	SECUR	IRQ	STOP	COP
MOR2 $3FF1	PB7PU	PB6PU	PB5PU	PB4PU	PB3PU	PB2PU	PB1PU	PB0PU

Figure 9-5. Mask Option Registers

PBXPU — Port B Pullup (X is 7–0)

When set, the PBPU bit enables the pullup on the corresponding port B pin. If the PBPU bit is cleared, the pullup devices are disabled. The erased state of the PBPU bit is to be cleared, thereby disabling the pullup devices.

NOTE

The MOR registers are reset to zero during reset. This causes the port B pullup devices to become inactive until reset is completed. This is a difference between the MC68HC705RC16 and the MC58HC05RC16 with the pullup options enabled. If the pullup options on the MC68HC05RC16 are enabled, the pullup devices will be active.

SECUR — SECURITY Enable

1 = This bit secures the EPROM by disabling a read of the EPROM in all modes other than user. This bit also disables writes to the MOR registers and the programming register.
0 = The EPROM can be read in all modes.

STOP — STOP Enable

1 = This bit enables the STOP instruction.
0 = A STOP instruction is equivalent to a WAIT instruction.

COP — COP Enable

1 = This bit enables the COP watchdog timer.
0 = The COP is disabled.

IRQ — IRQ sensitivity

1 = This bit selects the edge and level sensitive IRQ.
0 = IRQ is edge-only sensitive.

9.6 EPROM Electrical Specifications

The electrical specifications for the MC68HC705RC16 are the same as in the MC68HC05RC16 except for the following:

9.6.1 Maximum Ratings

Table 9-2. Maximum Ratings

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

NOTE: Voltages referenced by V_{SS}

9.6.2 DC Electrical Characteristics (5.0 Vdc and 3.3 Vdc)

Table 9-3. DC Electrical Characteristics (5.0 Vdc)

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

Characteristic	Symbol	Min	Typ	Max	Unit
EPROM Programming Voltage	V	14.5	—	15.5	V
Supply Current (see Notes)					
Run	I_{DD}	—	TBD	TBD	mA
Wait	I_{DD}	—	TBD	TBD	mA
Stop					
25 °C	I_{DD}	—	TBD	TBD	µA
0 °C to +70 °C	I_{DD}	—	TBD	TBD	µA

Table 9-4. DC Electrical Characteristics (3.3 Vdc)

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

Characteristic	Symbol	Min	Typ	Max	Unit
Supply Current (see Notes)					
Run	I_{DD}	—	TBD	TBD	mA
Wait	I_{DD}	—	TBD	TBD	mA
Stop					
25 °C	I_{DD}	—	TBD	TBD	µA
0 °C to +70 °C	I_{DD}	—	TBD	TBD	µA

NOTES:
1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25 ^{o}C only
3. Wait I_{DD}: only timer system active
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{OSC} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2
5. Wait, Stop I_{DD}: all ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V
6. Stop I_{DD} is measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

9.7 EPROM Control Timing

Table 9-5. EPROM Control Timing (3.3 and 5.0 Vdc)

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

SECTION 10
INSTRUCTION SET

This section describes the MC68HC05RC16 addressing modes and instruction types.

10.1 Addressing Modes

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

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

10.2 Inherent

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

10.2.1 Immediate

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

10.2.2 Direct

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

10.2.3 Extended

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

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

10.2.4 Indexed, No Offset

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

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

10.2.5 Indexed, 8-Bit Offset

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

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

10.2.6 Indexed, 16-Bit Offset

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

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

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

10.2.7 Relative

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

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

10.3 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

10.3.1 Register/Memory Instructions

Most of these instructions use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory. Table 10-1 lists the register/memory instructions.

Table 10-1. 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

10.3.2 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. The test for negative or zero instruction (TST) is an exception to the read-modify-write sequence because it does not write a replacement value. Table 10-2 lists the read-modify-write instructions.

Table 10-2. Read-Modify-Write Instructions

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

10.3.3 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. All branch instructions use relative addressing.

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

its condition (set or clear) is part of the opcode. The span of branching is from –128 to +127 from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register. Table 10-3 lists the jump and branch instructions.

Table 10-3. 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

10.3.4 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory. Port registers, port data direction registers, timer registers, and on-chip RAM locations are in the first 256 bytes of memory. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations. Bit manipulation instructions use direct addressing. Table 10-4 lists these instructions.

Table 10-4. Bit Manipulation Instructions

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

10.3.5 Control Instructions

These register reference instructions control CPU operation during program execution. Control instructions, listed in Table 10-5, use inherent addressing.

Table 10-5. 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

10.4 Instruction Set Summary

Table 10-6 is an alphabetical list of all M68HC05 instructions and shows the effect of each instruction on the condition code register.

Table 10-6. Instruction Set Summary

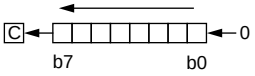
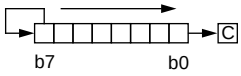
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr*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

Table 10-6. Instruction Set Summary (Continued)

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

Table 10-6. Instruction Set Summary (Continued)

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

Table 10-6. Instruction Set Summary (Continued)

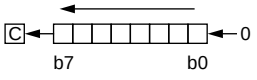
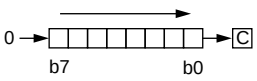
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
INC *opr* INCA INCX INC *opr*,X INC ,X	Increment Byte	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1	—	—	↕	↕	—	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff 	5 3 3 6 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Unconditional Jump	PC ← Jump Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BC C CC D C EC FC	dd hh ll ee ff ff 	2 3 4 3 2
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Conditional Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BD C CD D D D 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

Table 10-6. Instruction Set Summary (Continued)

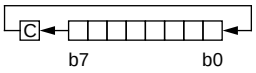
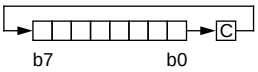
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	—	—	↕	↕	↕	DIR INH INH IX1 IX	30 40 50 60 70	ii ff 	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff 	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	39 49 59 69 79	dd ff 	5 3 3 6 5
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	36 46 56 66 76	dd ff 	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	INH	80		6
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)						INH			
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff 	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X	Store Accumulator in Memory	M ← (A)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff 	4 5 6 5 4
STOP	Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin		—	0	—	—	—	INH	8E		2

Table 10-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
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
TST *opr* TSTA TSTX TST *opr*,X TST ,X	Test Memory Byte for Negative or Zero	(M) – $00	—	—	—	—	—	DIR INH INH IX1 IX	3D 4D 5D 6D 7D	dd ff	4 3 3 5 4
TXA	Transfer Index Register to Accumulator	A ← (X)	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	↕	—	—	—	INH	8F		2

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

Table 10-7. Opcode Map

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

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

LSB of Opcode in Hexadecimal

MSB / LSB	0
0	BRSET0 5 3 DIR

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

SECTION 11
ELECTRICAL SPECIFICATIONS

This section contains MCU electrical specifications and timing information.

11.1 Maximum Ratings

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
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC05RC16 (Standard)	T_A	T_L to T_H 0 to +70	^oC
Storage Temperature Range	T_{STG}	-65 to +150	^oC

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

Maximum ratings are the extreme limits the device can be exposed to without causing permanent damage to the chip. The device is NOT intended to operate at these conditions.

11.2 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic DIP SOIC	θ_{JA}	 60 60	$^oC/W$

11.3 DC Electrical Characteristics (5.0 Vdc)

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

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} –2.0 mA) Port A, Port B, Port C (1–7) (I_{LOAD} –8.0 mA) IRO (I_{LOAD} –4.0 mA) Port C (Bit 0)	V_{OH}	 V_{DD} –0.8 V_{DD} –0.8 V_{DD} –0.8	 V_{DD} –0.2 V_{DD} –0.2 V_{DD} –0.2	 — — —	V
Output Low Voltage (I_{LOAD} = 3.0 mA) Port A, Port B, Port C (1–7) (I_{LOAD} = 25.0 mA) IRO (I_{LOAD} = 20.0 mA) Port C (Bit 0)	V_{OL}	— — —	0.2 0.2 0.2	0.4 0.4 0.4	V
Input High Voltage Port A, Port B, Port C, $\overline{IRQ}$, $\overline{RESET}$, $\overline{LPRST}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage Port A, Port B, Port C, $\overline{IRQ}$, $\overline{RESET}$, $\overline{LPRST}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Supply Current (see Notes) Run Wait Stop 25 °C 0 °C to +70 °C	I_{DD}	 — — — —	 2.3 0.5 0.3 0.3	 4.0 1.0 10.0 20.0	 mA mA µA µA
I/O Ports Hi-Z Leakage Current Port A, Port B, Port C	I_{OZ}	–10	—	10	µA
Input Current $\overline{RESET}$, $\overline{LPRST}$, $\overline{IRQ}$, OSC1 PB0–PB7 with Pullups Enabled (V_{IN} = 0.2 x V_{DD})[8] PB0–PB7 with Pullups Enabled (V_{IN} = 0.7 x V_{DD})	I_{IN}	 –1 –100 –50	 — –330 –120	 1 –700 –300	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{LPRST}$, $\overline{IRQ}$	 C_{OUT} C_{INT}	 — —	 — —	 12 8	pF

NOTES:

1. All values shown reflect average measuremen.ts
2. Typical values at midpoint of voltage range, 25 °C only
3. Wait I_{DD}: only core timer active
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{OSC} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2
5. Wait, Stop I_{DD}: all ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} -0.2 V
6. Stop I_{DD} is measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.
8. Pullups are designed to be capable of pulling to V_{IH} within 1 µs for a 100 pF, 4-kΩ load

11.4 DC Electrical Characteristics (3.3 Vdc)

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

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage					V
I_{LOAD} = 10.0 µA	V_{OL}	—	—	0.1	
I_{LOAD} = -10.0 µA	V_{OH}	V_{DD}– 0.1	—	—	
Output High Voltage	V_{OH}				V
(I_{LOAD} –0.8 mA) Port A, Port B, Port C (1-7)		V_{DD} –0.3	V_{DD} –0.1	—	
(I_{LOAD} –4.0 mA) IRO		V_{DD} –0.3	V_{DD} –0.1	—	
(I_{LOAD} –2.0 mA) Port C (Bit 0)		V_{DD} –0.3	V_{DD} –0.1	—	
Output Low Voltage	V_{OL}				V
(I_{LOAD} = 1.2 mA) Port A, Port B, Port C (1-7)		—	0.1	0.3	
(I_{LOAD} = 9.0 mA) IRO		—	0.1	0.3	
(I_{LOAD} = 8.0 mA) Port C (Bit 0)		—	0.1	0.3	
Input High Voltage	V_{IH}	0.7 x	—	V_{DD}	V
Port A, Port B, Port C, $\overline{IRQ}$, $\overline{RESET}$, $\overline{LPRST}$, OSC1		V_{DD}			
Input Low Voltage				0.3 x	V
Port A, Port B, Port C, $\overline{IRQ}$, $\overline{RESET}$, $\overline{LPRST}$, OSC1	V_{IL}	V_{SS}	—	V_{DD}	
Supply Current (see Notes)	I_{DD}				
Run		—	0.7	1.5	mA
Wait		—	0.25	0.5	mA
Stop					
25 °C		—	0.2	4.0	µA
0 °C to +70 °C		—	0.2	8.0	µA
I/O Ports Hi-Z Leakage Current	I_{OZ}	–6	—	6	µA
Port A, Port B, Port C					
Input Current	I_{IN}				µA
$\overline{RESET}$, $\overline{LPRST}$, $\overline{IRQ}$, OSC1		–0.6	—	0.6	
PB0-PB7 with Pullups Enabled		–50	–140	–330	
(V_{IN} = 0.3 x V_{DD}) [8]					
PB0-PB7 with Pullups Enabled		–25	–76	–180	
(V_{IN} = 0.7 x V_{DD})					
Capacitance					pF
Ports (as Input or Output)	C_{OUT}	—	—	12	
$\overline{RESET}$, $\overline{LPRST}$, $\overline{IRQ}$	C_{INT}	—	—	8	

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25 °C only
3. Wait I_{DD}: only core timer active
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{OSC} = 2.0 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2
5. Wait, Stop I_{DD}: all ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} -0.2 V
6. Stop I_{DD} is measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.
8. Pullups are designed to be capable of pulling to V_{IH} within 10 µs for a 100 pF, 4-kΩ load.

11.5 DC Electrical Characteristics (2.2 Vdc)

(V_{DD} = 2.2 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0 °C to +70 °C, unless otherwise noted)

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} –0.6mA) Port A, Port B, Port C (1–7) (I_{LOAD} –2.5 mA) IRO (I_{LOAD} –1.2 mA) Port C (Bit 0)	V_{OH}	 V_{DD}– 0.3 V_{DD}– 0.3 V_{DD}– 0.3	 V_{DD}– 0.1 V_{DD}– 0.1 V_{DD}– 0.1	 — — —	V
Output Low Voltage (I_{LOAD} = 1.0 mA) Port A, Port B, Port C (1–7) (I_{LOAD} = 8.0 mA) IRO (I_{LOAD} = 7.0 mA) Port C (Bit 0)	V_{OL}	 — — —	 0.1 0.1 0.1	 0.3 0.3 0.3	V
Input High Voltage Port A, Port B, Port C, $\overline{IRQ}$, $\overline{RESET}$, $\overline{LPRST}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage Port A, Port B, Port C, $\overline{IRQ}$, $\overline{RESET}$, $\overline{LPRST}$, OSC1	V_{IL}	V_{SS}	—	0.4 x V_{DD}	V
Supply Current (see Notes) Run Wait Stop 25 °C 0 °C to +70 °C	 I_{DD} I_{DD} I_{DD} I_{DD}	 — — — —	 0.3 0.15 0.1 0.1	 1.0 0.3 1.0 4.0	 mA mA µA µA
I/O Ports Hi-Z Leakage Current Port A, Port B, Port C	I_{OZ}	–4	—	4	µA
Input Current $\overline{RESET}$, $\overline{LPRST}$, $\overline{IRQ}$, OSC1 PB0-PB7 with pullups enabled (V_{IN} = 0.4 x V_{DD}) [8] PB0-PB7 with pullups enabled (V_{IN} = 0.7 x V_{DD})	I_{IN}	 –0.4 –50 –25	 — –140 –76	 0.4 –300 –150	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{LPRST}$, $\overline{IRQ}$	 C_{OUT} C_{INT}	 — —	 — —	 12 8	pF

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25 °C only
3. Wait I_{DD}: only core timer active
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{OSC} = 2.0 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2
5. Wait, Stop I_{DD}: all ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} -0.2 V
6. Stop I_{DD} is measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.
8. Pullups are designed to be capable of pulling to V_{IH} within 25 µs for a 100 pF, 4-kΩ load.

11.6 Control Timing (3.3 Vdc and 5.0 Vdc)

(V_{DD} = 3.0 to 5.5 Vdc, V_{SS} = 0 Vdc, T_A = 0 °C to +70 °C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal External Clock	 f_{OSC} f_{OSC}	 — dc	 4.2 4.2	 MHz MHz
Internal Operating Frequency Crystal (f_{OSC} /4) External Clock (f_{OSC} /4)	 f_{OP} f_{OP}	 — dc	 2.1 2.1	 MHz MHz
Cycle Time	t_{CYC}	480	—	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}	*	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

* The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{CYC}.

11.7 Control Timing (2.2 Vdc)

(V_{DD} = 2.2 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0 °C to +70 °C, unless otherwise noted)

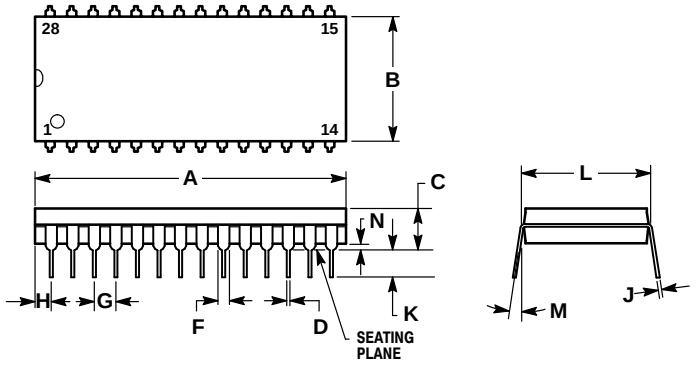
Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal External Clock	 f_{OSC} f_{OSC}	 — dc	 2.0 2.0	 MHz MHz
Internal Operating Frequency Crystal (f_{OSC} /4) External Clock (f_{OSC} /4)	 f_{OP} f_{OP}	 — dc	 1.0 1.0	 MHz MHz
Cycle Time	t_{CYC}	1000	—	ns
Crystal Oscillator Startup Time	t_{OXOV}	—	200	ms
Stop Recovery Startup Time (Crystal Oscillator)	t_{ILCH}	—	200	ms
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{CYC}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	250	—	ns
Interrupt Pulse Period	t_{ILIL}	*	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	180	—	ns

* The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{CYC}.

SECTION 12
MECHANICAL SPECIFICATIONS

This section describes the dimensions of the dual-in-line package (DIP) and small outline integrated circuit (SOIC) MCU packages.

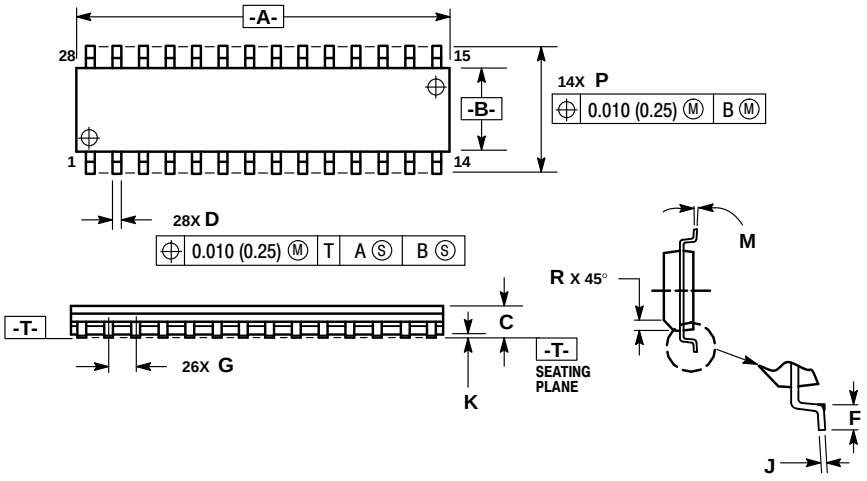
12.1 28-Pin Plastic Dual-In-Line Package (Case 710-02)



NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

12.2 28-Pin Small Outline Integrated Circuit Package (Case 751F-04)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 (0.005) TOTAL IN EXCESS OF D DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	17.80	18.05	0.701	0.711
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.41	0.90	0.016	0.035
G	1.27 BSC		0.050 BSC	
J	0.23	0.32	0.009	0.013
K	0.13	0.29	0.005	0.011
M	0°	8°	0°	8°
P	10.05	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

SECTION 13
ORDERING INFORMATION

13.1 Introduction

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

13.2 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 **13.3 Application Program Media 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 lowercase letters. Then press the return key to start the BBS software.

13.3 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 360 K or double-sided high-density 1.2 M)

---

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.

Use positive logic for data and addresses.

When submitting the application program on a diskette, clearly label the diskette with the following 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.

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 non-user ROM locations or leave all non-user ROM locations blank**. Refer to the current MCU ordering form for additional requirements. Motorola may request pattern re-submission if non-user areas contain any non-zero code.

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

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 filename of the source code.

13.4 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 non-user 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 the listing verify form to Motorola. The signed listing verify form constitutes the contractual agreement for the creation of the custom mask.

13.5 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 ten 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 ten 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.

13.6 MC Order Numbers

Table 13-1 shows the MC order numbers for the available package types.

Table 13-1. MC Order Numbers

Package Type	Operating Temperature Range	MC Order Number
28-Pin Plastic Dual In-Line Package (DIP)	0 °C to 70 °C	MC68HC05RC16P
28-Pin Small Outline Integrated Circuit Package (SOIC)	0 °C to 70 °C	MC68HC05RC16DW

How to reach us:

USA/EUROPE: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447

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

INTERNET: http://Design-NET.com

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

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

MOTOROLA

HC05RC16GRS/D