HC705RC17GRS/D
REV. 2.0

68HC705RC17

General Release Specification

May 20, 1997

CSIC MCU Design Center
Austin, Texas

MOTOROLA

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

List of Sections

Table of Contents

Section 1. General Description

Section 2. Memory

Section 3. Central Processor Unit (CPU)

Section 4. Interrupts

Section 5. Resets

Section 6. Low-Power Modes

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

Section 8. Core Timer

Section 9. Carrier Modulator Transmitter (CMT)

Section 10. Phase-Locked Loop (PLL) Synthesizer

Section 11. EPROM

Section 12. Instruction Set

Section 13. Electrical Specifications

Section 14. Mechanical Specifications

Section 15. Ordering Information

List of Figure

List of Tables

Section 1. General Description

1.1 Contents

1.2 Introduction

The MC68HC705RC17 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), computer operating properly (COP) watchdog systems, and a programmable phase-lock loop (PLL) synthesizer. 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 18 input-output (I/O) lines (eight having keyscan logic and pullups) and a low-power reset pin.

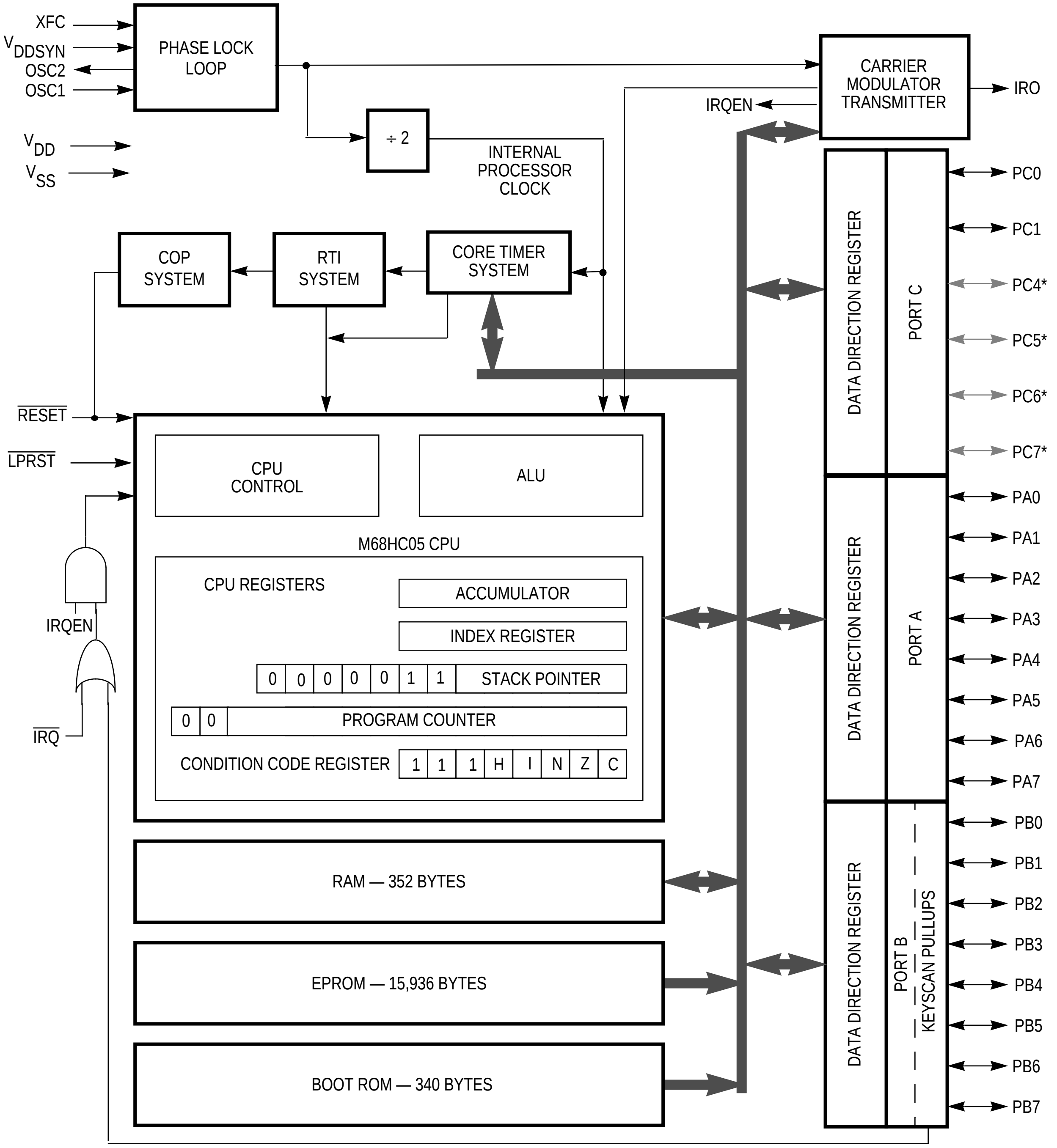
The MC68HC705RC17 is available in 28-pin small outline integrated circuit (SOIC) or plastic dual in-line package (PDIP) packages. Four additional I/O lines are available for bond out in higher pin count packages.

NOTE: *Consult the factory for availability of higher pin count packages.*

1.3 Features

Features of the MC68HC705RC17 include:

- Low Cost
- HC05 Core
- 28-Pin SOIC or DIP Packages
- On-Chip Oscillator with 32.768-kHz Crystal/Ceramic Resonator
- Phase-Locked Loop (PLL) Synthesizer with Programmable Speed
- 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
- 18 Bidirectional I/O Lines (Four Additional I/O Lines Available for Bond Out in Higher Pin Count Packages)
- Programmable Pullups and Interrupt on Eight Port Pins (PB0–PB7)
- High-Current Infrared (IR) Drive Pin
- High-Current Port Pin (PC0)



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

Figure 1-1. MC68HC705RC17 Block Diagram

- Power-Saving Stop and Wait Modes
- Programmable 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.*

1.4 Options

The options on the MC68HC705RC17 are handled with 11 EPROM bits in two separate mask option register (MOR1 and MOR 2). These options are:

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

The factory will program these options in the ROM versions of this device.

1.5 Signal Description

Pinout for the 28-pin small outline integrated circuit (SOIC) package is shown in **Figure 1-2**. Pinout for the 28-pin plastic dual in-line (PDIP) package is shown in **Figure 1-3**. The signals are described in the following subsections.

NOTE: *Consult the factory for availability of higher pin count packages.*

1.5.1 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.

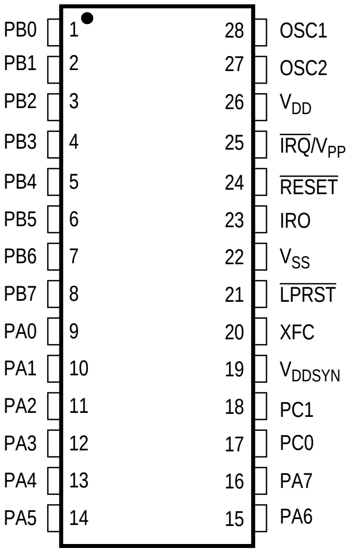


Figure 1-2. 28-Pin SOIC Pinout

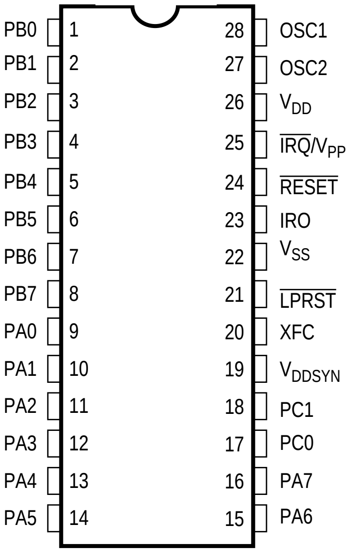


Figure 1-3. 28-Pin PDIP Pinout

1.5.2 $\overline{\text{IRQ}}/V_{PP}$ (Maskable Interrupt Request)

This pin supplies the EPROM with the required programming voltage. In addition, this pin serves as the input for external interrupts. Triggering sensitivity is programmable using MOR1. 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 $\overline{\text{IRQ}}$ goes low for at least one t_{ILIH}, a logic 1 is latched internally to signify that an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch contains a logic 1 and the interrupt mask bit (I bit) in the condition code register is clear, the MCU then begins the interrupt sequence.

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

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

Refer to **Section 4. Interrupts** for more detail.

1.5.3 OSC1 and OSC2

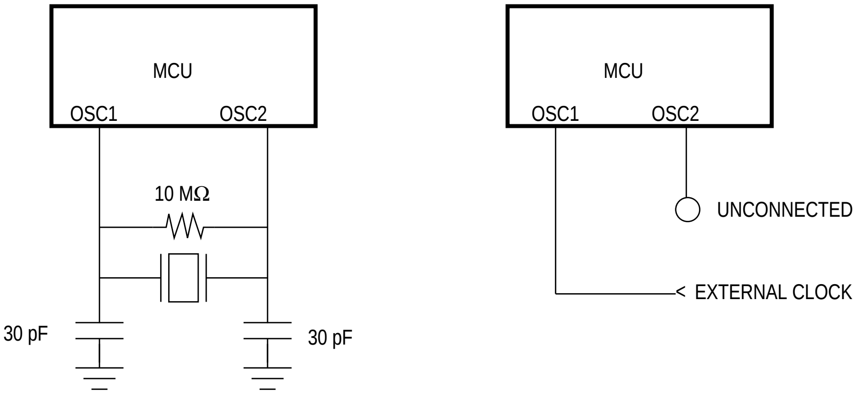
These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, or an external signal connects to these pins to provide a system clock. The oscillator frequency is two times the internal bus rate.

Figure 1-4 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 startup stabilization time.

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. **Figure 1-4** **(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 1-4** **(b)**). This setup can be used if the user does not want to run the CPU with a crystal.

NOTE: *The PLL design requires a 32.768-kHz external frequency for proper operation.*



(a) Crystal/Ceramic Resonator Oscillator Connections

(b) External Clock Source Connections

Figure 1-4. Oscillator Connections

1.5.4 $\overline{\text{RESET}}$

This active-low pin is used to reset the MCU to a known startup 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**.

1.5.5 $\overline{\text{LPRST}}$

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. See **Section 5. Resets**.

1.5.6 IRO

The IRO pin is the high-current source and sink output of the carrier modulator transmitter subsystem which is suitable for driving infrared (IR) LED biasing logic. See **Section 9. Carrier Modulator Transmitter (CMT)**.

1.5.7 Port A (PA0–PA7)

These eight I/O lines comprise port A. The state of any pin is software programmable and all port A lines are configured as inputs during power-on or reset. For detailed information on I/O programming, see **Section 7. Parallel Input/Output (I/O)**.

1.5.8 Port B (PB0–PB7)

These eight I/O lines comprise port B. The state of any pin is software programmable and all port B lines are configured as inputs during power-on or reset. Each port B I/O line has a mask optionable pullup for keyscan. For detailed information on I/O programming, see **Section 7. Parallel Input/Output (I/O)**.

1.5.9 Port C (PC0–PC1 and PC4–PC7)

These six 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 **Section 7. Parallel Input/Output (I/O)**.

NOTE: *Only two bits of port C are bonded out in 28-pin packages for the MC68HC705RC17, although port C is truly a 6-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.*

1.5.10 XFC

This pin provides a means for connecting an external filter capacitor to the synthesizer phase-locked loop filter. (For additional information concerning this capacitor, refer to **Section 10. Phase-Locked Loop (PLL) Synthesizer**.)

1.5.11 V_{DDSYN}

This pin provides a separate power connection to the PLL synthesizer which should be at the same potential as V_{DD}.

NOTE: *Any unused inputs, I/O ports, and no connects should be tied to an appropriate logic level (either* V_{DD} *or* V_{SS}*). Although the I/O ports of the MC68HC705RC17 do not require termination, termination is recommended to reduce the possibility of static damage.*

Section 2. Memory

2.1 Contents

2.2 Introduction

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

2.3 Memory Map

The MC68HC705RC17 has a 16-Kbyte memory map consisting of user EPROM, RAM, bootloader ROM, and input/output (I/O).

Figure 2-1 is a memory map for the MC68HC705RC17. **Figure 2-2** is a more detailed memory map of the I/O register section.

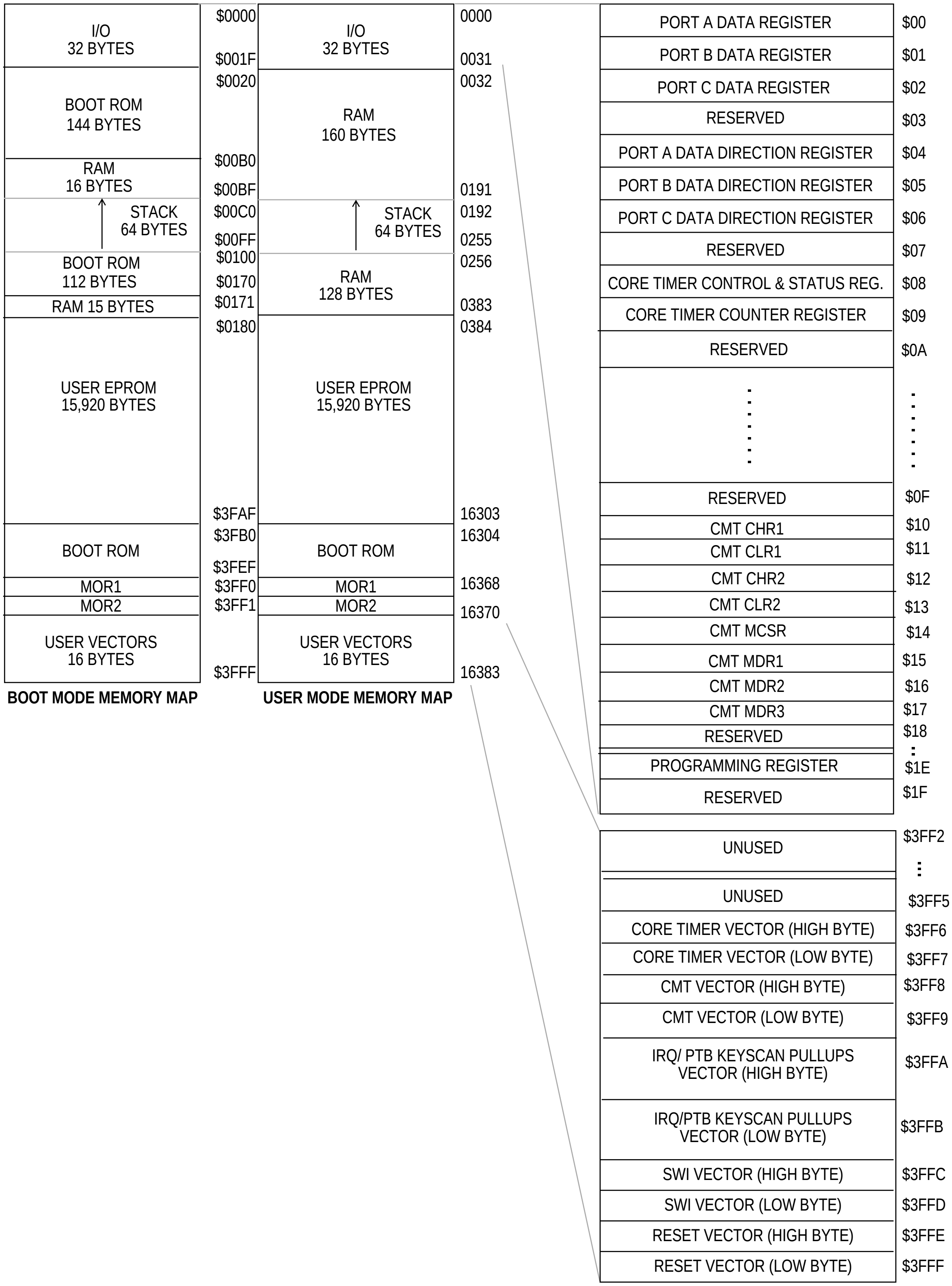


Figure 2-1. MC68HC705RC17 Memory Map

Addr	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$00	Port A Data Register	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
$01	Port B Data Register	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
$02	Port C Data Register	PC7	PC6	PC5	PC4	▨	▨	PC1	PC0
$03	Reserved	R	R	R	R	R	R	R	R
$04	Port A Data Direction Register	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
$05	Port B Data Direction Register	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
$06	Port C Data Direction Register	DDRC7	DDRC6	DDRC5	DDRC4	▨	▨	DDRC1	DDRC0
$07	PLL Control Register	0	BCS	0	BWC	PLLON	VCOTST	PS1	PS0
$08	Timer Control and Status Register	CTOF	RTIF	TOFE	RTIE	TOFC	RTFC	RT1	RT0
$09	Timer Counter Register	D7	D6	D5	D4	D3	D2	D1	D0
$0A	Reserved	R	R	R	R	R	R	R	R
$0B	Reserved	R	R	R	R	R	R	R	R
$0C	Reserved	R	R	R	R	R	R	R	R
$0D	Reserved	R	R	R	R	R	R	R	R
$0E	Reserved	R	R	R	R	R	R	R	R
$0F	Reserved	R	R	R	R	R	R	R	R
$10	CMT Timer CHR1	IROLN	CMTPOL	PH5	PH4	PH3	PH2	PH1	PH0
$11	CMT Timer CLR1	IROLP	0	PL5	PL4	PL3	PL2	PL1	PL0
$12	CMT Timer CHR2	0	0	SH5	SH4	SH3	SH2	SH1	SH0
$13	CMT Timer CLR2	0	0	SL5	SL4	SL3	SL2	SL1	SL0
$14	CMT Timer MCSR	EOC	DIV2	EIMSK	EXSPC	BASE	MODE	IE	MCGEN
$15	CMT Timer MDR1	MB11	MB10	MB9	MB8	SB11	SB10	SB9	SB8
$16	CMT Timer MDR2	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
$17	CMT Timer MDR3	SB7	SB6	SB5	SB4	SB3	SB2	SB1	SB0

▨ = Unimplemented R = Reserved

Figure 2-2. I/O Registers (Sheet 1 of 2)

Addr	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
$18	Reserved	R	R	R	R	R	R	R	R
$19	Reserved	R	R	R	R	R	R	R	R
$1A	Reserved	R	R	R	R	R	R	R	R
$1B	Reserved	R	R	R	R	R	R	R	R
$1C	Reserved	R	R	R	R	R	R	R	R
$1D	Reserved	R	R	R	R	R	R	R	R
$1E	Reserved	R	R	R	R	R	R	R	R
$1F	Reserved	R	R	R	R	R	R	R	R

R = Reserved

Figure 2-2. I/O Registers (Sheet 2 of 2)

2.3.1 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 $0020 to $00AF and $0100 to $170 in boot mode only, and from $3FB0 to $3FEF in both boot mode and burn-in mode.

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

2.3.2 EPROM Security

The MC68HC705RC17 contains special circuitry to prevent accessing the EPROM in non-user mode. Emulation will not be affected by this change.

Security is controlled by a security bit in the MOR register. It is intended to be programmed while the users are programming their code. This will inhibit reading of the EPROM in all modes other than user mode.

2.3.3 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.*

2.3.4 Bootloader ROM

Addresses \$0020 to \$00AF, \$0100 to \$0170, and \$3FB0 to \$3FEF are reserved ROM addresses that contain the instructions for the bootloader functions. (See **Section 11. EPROM**.)

2.4 Input/Output Programming

In user mode, 18 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. For detailed information, refer to **Section 7. Parallel Input/Output (I/O)**.

Section 3. Central Processor Unit (CPU)

3.1 Contents

3.2 Introduction

This section describes the registers of the MC68HC705RC17's central processor unit (CPU).

3.3 CPU Registers

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

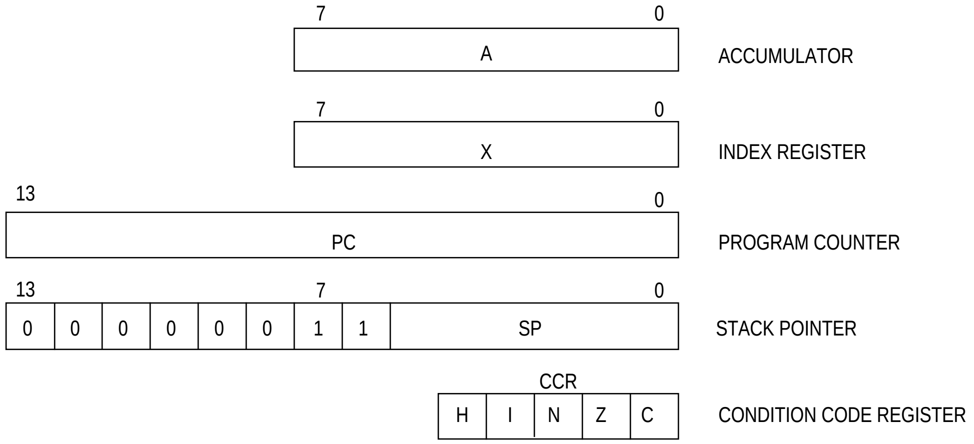
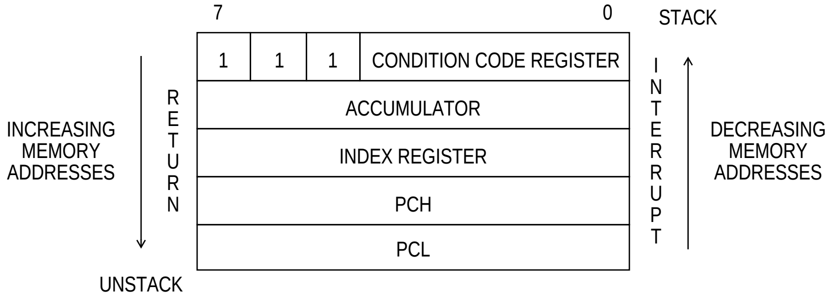


Figure 3-1. Programming Model

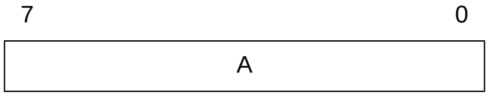


NOTE: Since the stack pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 3-2. Stacking Order

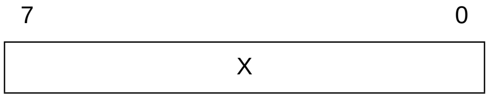
3.4 Accumulator

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



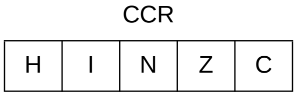
3.5 Index Register

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



3.6 Condition Code Register

The condition code register (CCR) is a 5-bit register in which the H, N, Z, and C bits are used to indicate the results of the instruction just executed, and the I bit is used to enable 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.



Half Carry (H)

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

Interrupt (I)

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

Negative (N)

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

Zero (Z)

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

Carry/Borrow (C)

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

3.7 Stack Pointer

The stack pointer (SP) 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.

13						7		0
0	0	0	0	0	0	1	1	SP

3.8 Program Counter

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

13	0
PC	

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

4.1 Contents

4.2 Introduction

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

4.3 CPU Interrupt Processing

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

If interrupts are not masked (I bit in the CCR is clear) and the corresponding interrupt enable bit is set, the processor will proceed with interrupt processing. Otherwise, the next instruction is fetched and executed. If an interrupt occurs, the processor completes the current instruction, stacks the current CPU register 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 return from interrupt (RTI) instruction is used to signify when the interrupt software service routine is completed. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume at the next instruction that was to be executed when the interrupt took place. **Figure 4-1** shows the sequence of events that occurs during interrupt processing.

4.4 Reset Interrupt Sequence

The reset function is not in the strictest sense an interrupt; however, it is acted upon in a similar manner as shown in **Figure 4-1**. A low-level input on the $\overline{\text{RESET}}$ pin or an internally generated RST signal causes the program to vector to its starting address, which is specified by the contents of memory locations $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.

4.5 Software Interrupt (SWI)

The SWI is an executable instruction and a nonmaskable interrupt since it is executed regardless of the state of the I bit in the CCR. If the I bit is zero (interrupts enabled), the SWI instruction executes after interrupts that were pending before the SWI was fetched or before interrupts generated after the SWI was fetched. The interrupt service routine address is specified by the contents of memory locations $3FFC and $3FFD.

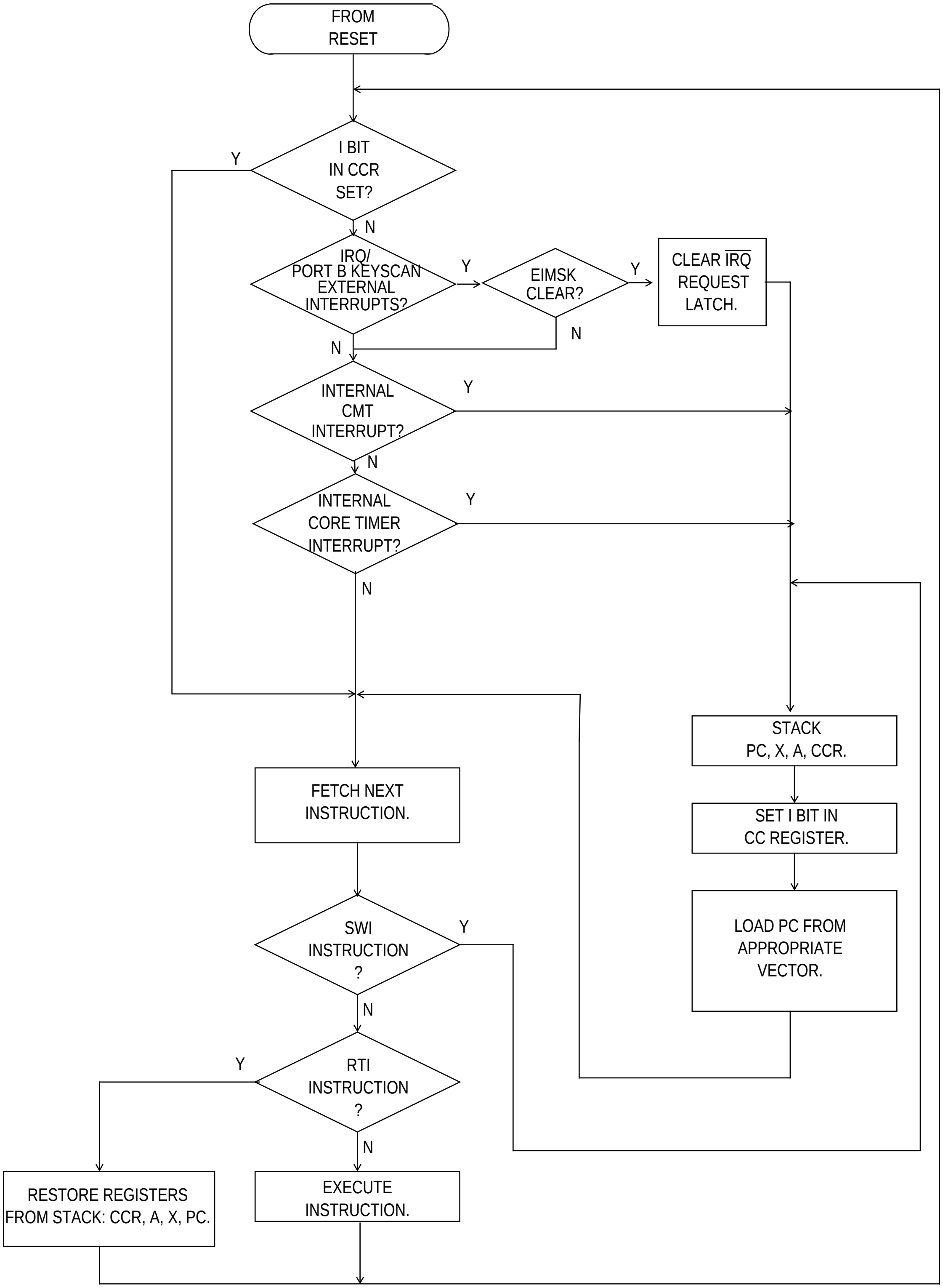


Figure 4-1. Interrupt Processing Flowchart

4.6 Hardware Interrupts

All hardware interrupts except $\overline{\text{RESET}}$ are maskable by the I bit in the CCR. If the I bit is set, all hardware interrupts (internal and external) are disabled. Clearing the I bit enables the hardware interrupts. The three types of hardware interrupts are explained in the following sections.

4.7 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.*

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 pullup/interrupt feature is enabled as specified by the user.

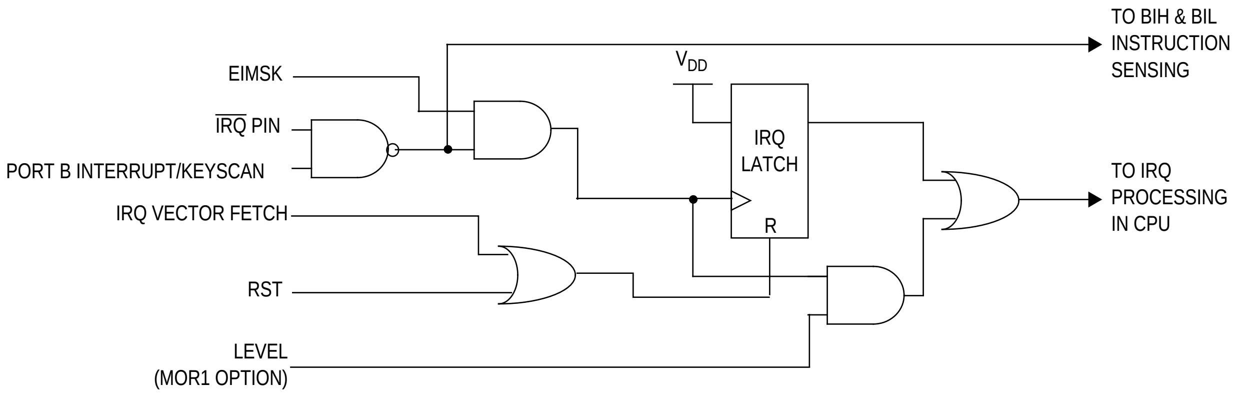


Figure 4-2. IRQ Function Block Diagram

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

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

When edge and level sensitivity is selected for the IRQ interrupt, it is sensitive to these 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 pullup/interrupt enabled

External interrupts also can be masked by setting the EIMSK bit in the MSCR register of the IR remote timer. See **9.5.4 Modulator Period Data Registers** for details.

4.8 External Interrupt Timing

If the interrupt mask bit (I bit) of the CCR is set, all maskable interrupts (internal and external) are disabled. Clearing the I bit enables interrupts. The interrupt request is latched immediately following the falling edge of the $\overline{\text{IRQ}}$ source. 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.

4.9 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.

4.10 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

5.1 Contents

5.2 Introduction

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.

5.3 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.

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

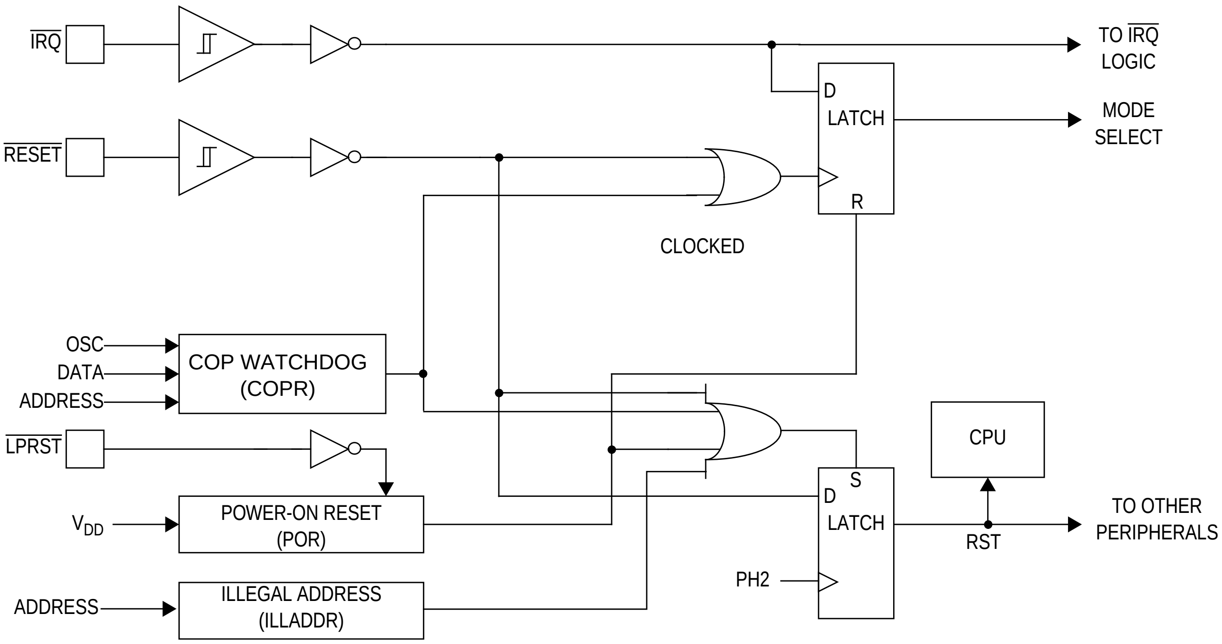
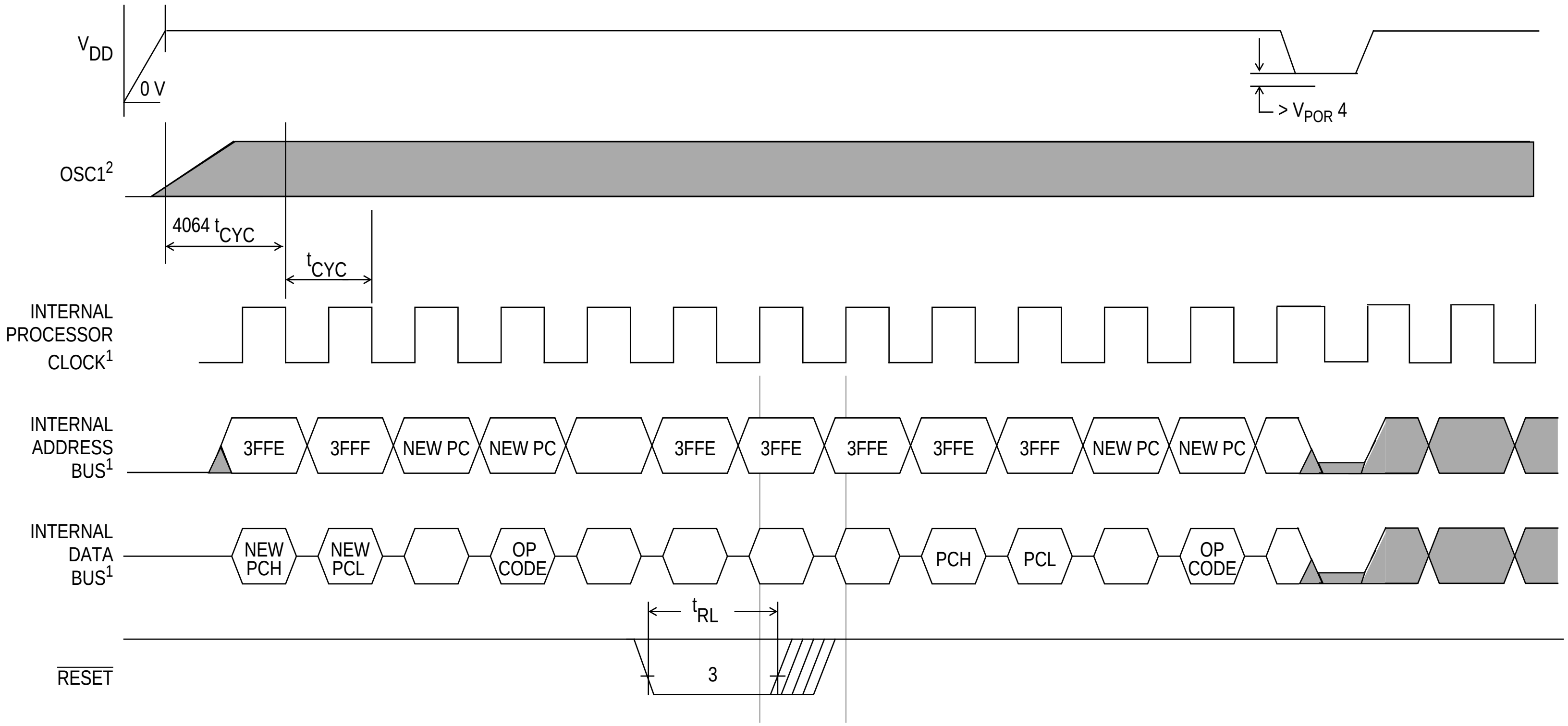


Figure 5-1. Reset Block Diagram



NOTES:

1. Internal timing signal and bus information are not available externally.
2. OSC1 line is not meant to represent frequency. It is only used to represent time.
3. The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.
4. V_{DD} must fall to a level lower than V_{POR} to be recognized as a power-on reset.
5. The $\overline{\text{LPRST}}$ pin resets the CPU like the $\overline{\text{RESET}}$ pin. However, 4064 POR cycles are executed first, before the reset vector address appears on the internal address bus. (See **5.3 External Reset ($\overline{\text{RESET}}$)**.)

Figure 5-2. Reset and POR Timing Diagram

5.4 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 0 remains on LPRST. When a logic 1 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.

5.5 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.

5.5.1 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.

5.5.2 Computer Operating Properly Reset (COPR)

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

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

5.5.2.1 Resetting the COP

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

5.5.2.2 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.

5.5.2.3 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. Therefore, it is recommended that the COP is fed before executing the STOP instruction.

5.5.2.4 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:	THEN the COP Watchdog Timer Should Be:
Wait Time	
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

5.5.3 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

6.1 Contents

6.2 Introduction

This section describes the low-power modes.

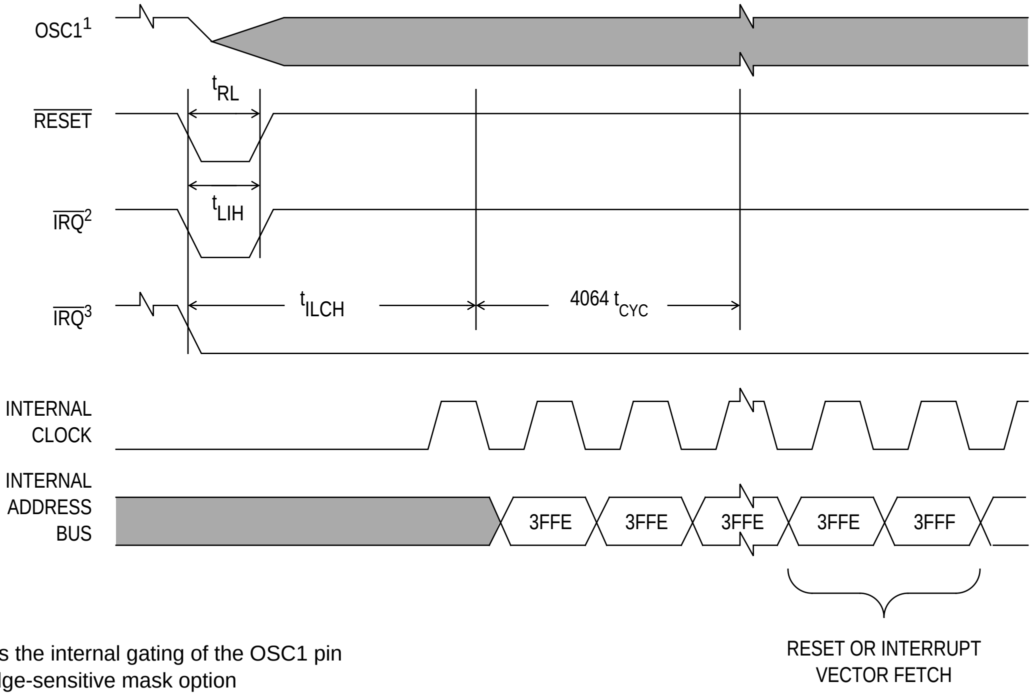
6.3 Stop Mode

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.

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

NOTE: *If an external interrupt is pending when stop mode is entered, then stop mode will be exited immediately.*



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

Figure 6-1. Stop Recovery Timing Diagram

NOTE: *The EIMSK bit of the carrier modulator transmitter MCSR (modulator control and status register) 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 Wait Mode

The WAIT instruction places the MCU in a low power-consumption mode, but wait mode consumes more power than 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 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 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 wait mode.

NOTE: *For minimum current consumption, the phase-locked loop (PLL) should be disabled or turned off before entering wait mode.*

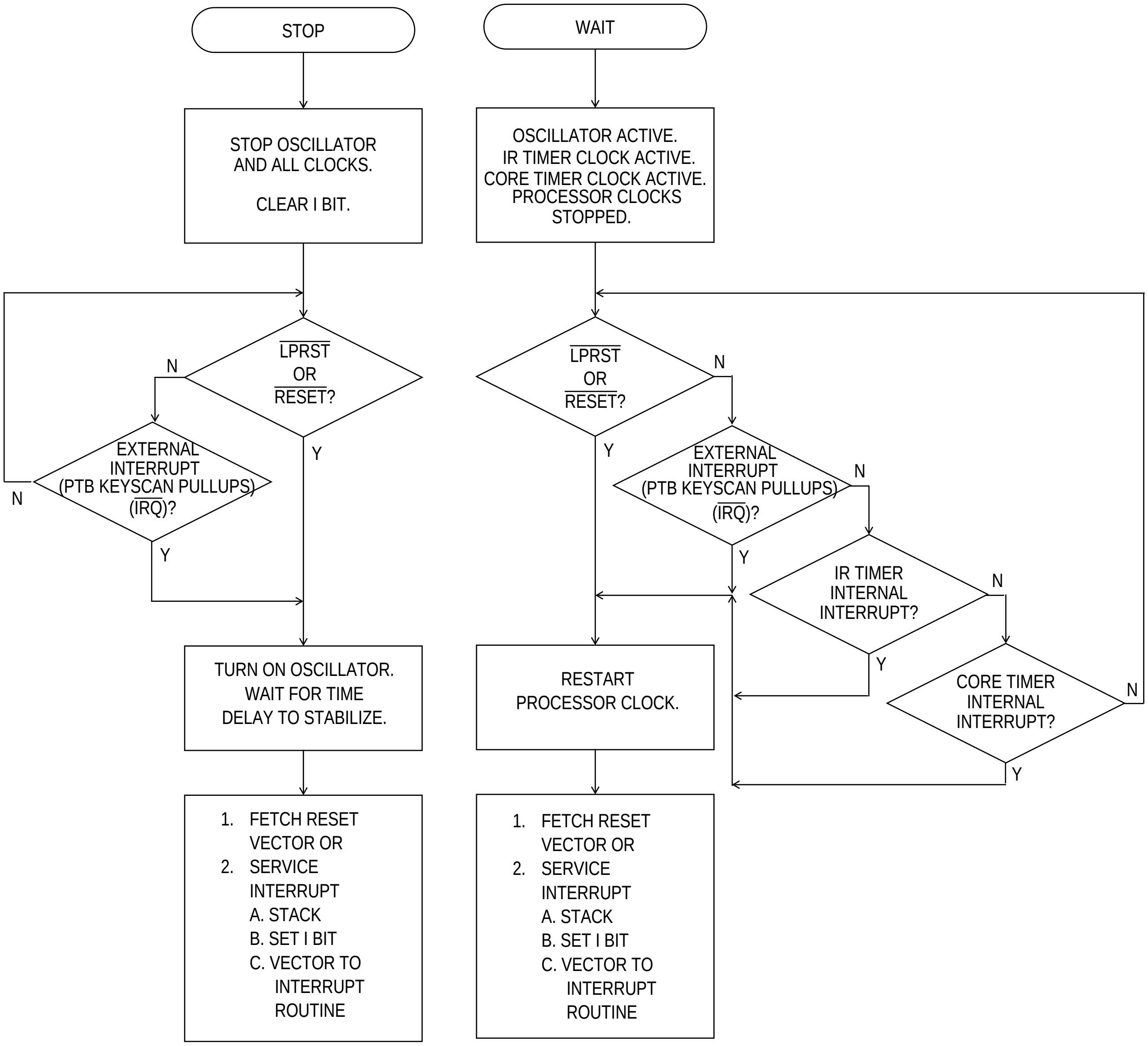


Figure 6-2. Stop/Wait Flowchart

6.5 Low-Power Reset

Low-power reset mode is entered when a logic 0 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 1 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. Parallel Input/Output (I/O)

7.1 Contents

7.2 Introduction

In user mode, 18 lines are arranged as one 2-bit and two 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 1 to the corresponding data direction register.*

Four extra I/O ports are available on higher pin count packages. Consult factory for availability.

7.3 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 1 to a DDR bit sets the corresponding port bit to output mode.

7.4 Port B

Port B is an 8-bit bidirectional port 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 1 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 1 or the I bit should be set in the condition code register to prevent an interrupt from occurring.

NOTE: *When a port B pin is configured as an output, its corresponding keyscan interrupt is disabled, regardless of its mask option.*

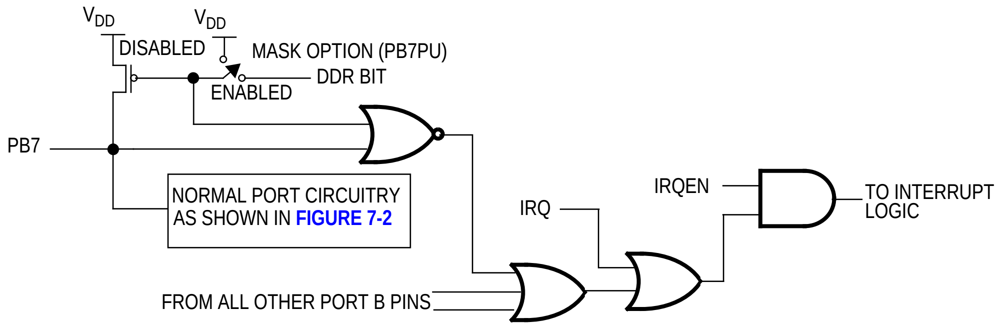


Figure 7-1. Port B Pullup Option

7.5 Port C

Port C is a 6-bit bidirectional port (PC0–PC1 and PC4–PC7) which does not share any of its pins with other subsystems. The port C data register is at $0002 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 1 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 two bits of port C are bonded out in 28-pin packages for the MC68HC05RC17, although port C is truly a 6-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.*

7.6 Input/Output (I/O) 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 1. A pin is configured as an input if its corresponding DDR bit is cleared to a logic 0.

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

Table 7-1. I/O Pin Functions

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

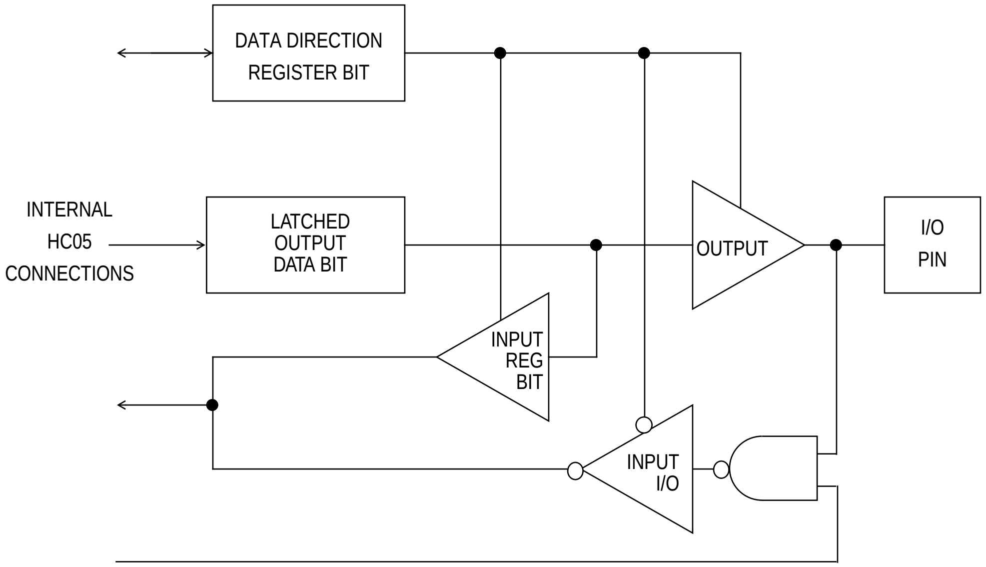


Figure 7-2. I/O Circuitry

Section 8. Core Timer

8.1 Contents

8.2 Introduction

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.

As seen in **Figure 8-1**, 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.

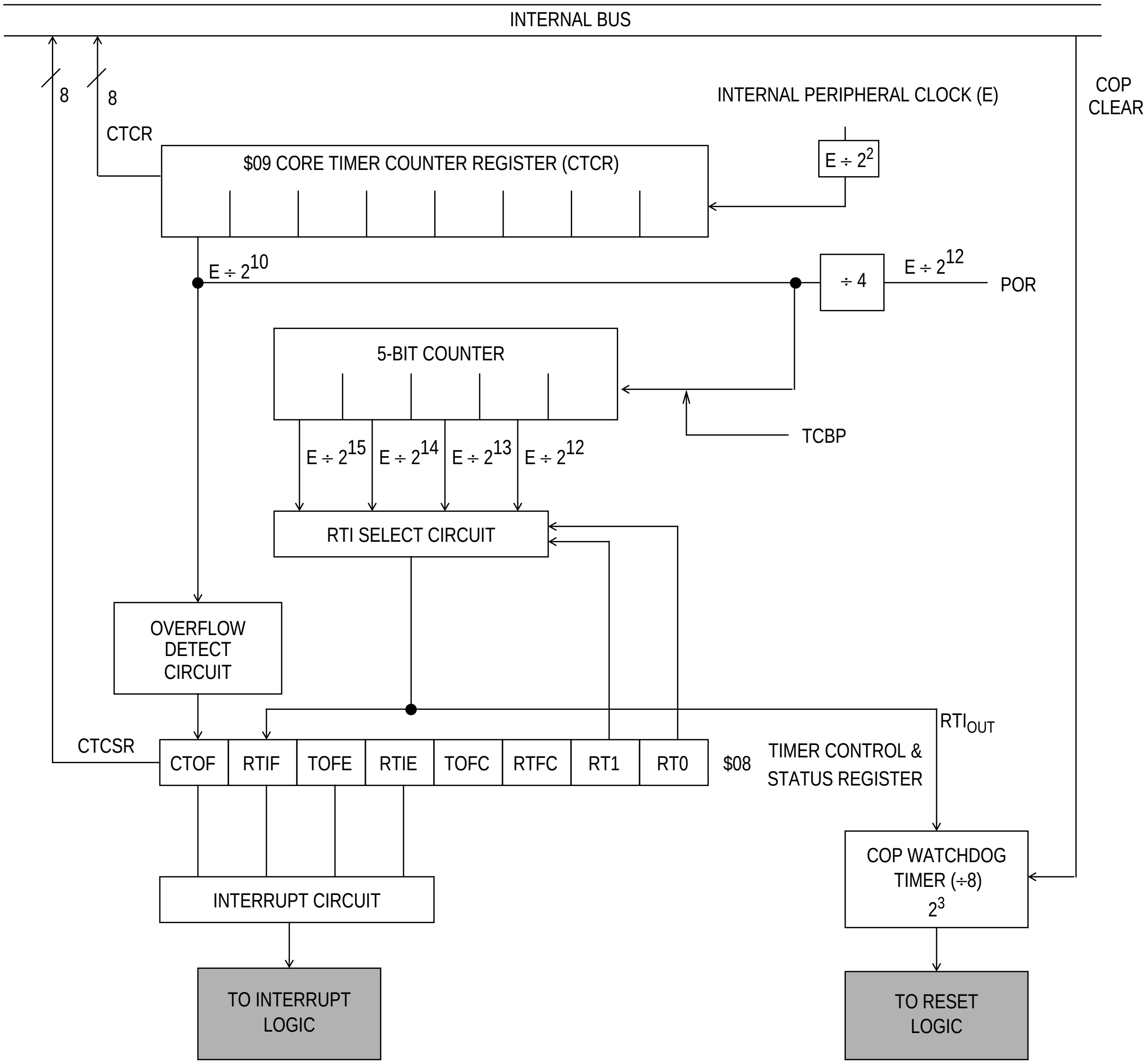


Figure 8-1. Core Timer Block Diagram

8.3 Core Timer Control and Status Register

The CTCSR contains the timer interrupt flag, the timer interrupt enable bits, and the real-time interrupt rate select bits. **Figure 8-2** shows the value of each bit in the CTCSR when coming out of reset.

Address: $0008

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

▨ = Unimplemented

Figure 8-2. Core Timer Control and Status Register (CTCSR)

CTOF — Core Timer Overflow

CTOF is a read-only status bit set when the 8-bit ripple counter rolls over from $FF to $00. Clearing the CTOF is done by writing a 1 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 3-stage divider and a one-of-four selector. The clock frequency that drives the RTI circuit is E/2^{12} (or E ÷ 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 1 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 1 is written to this bit, CTOF is cleared. Writing a 0 has no effect on the CTOF bit. This bit always reads as 0.

RTFC — Real-Time Interrupt Flag Clear

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

RT1 and RT0 — Real-Time Interrupt Rate Select

These two bits select one of four taps from the real-time interrupt circuit. Refer to **Table 8-1**. 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 8-1. RTI and COP Rates at 4.096-MHz Oscillator

RTI Rate 2.048-MHz Bus		RT1 and 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

8.4 Core Timer Counter Register

The timer counter register is a read-only register that contains the current value of the 8-bit ripple counter at the beginning of the timer chain. This counter is clocked by the CPU clock (E/4) and can be used for various functions, including a software input capture. Extended time periods can be attained using the TOF function to increment a temporary RAM storage location, thereby simulating a 16-bit (or more) counter.

Address: $0009

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	D7	D6	D5	D4	D3	D2	D1	D0
Write:	▨	▨	▨	▨	▨	▨	▨	▨
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 8-3. Core Timer Counter Register (CTCR)

The power-on cycle clears the entire counter chain and begins clocking the counter. After 4064 cycles, the power-on reset circuit is released, which again clears the counter chain and allows the device to come out of reset. At this point, if $\overline{\text{RESET}}$ is not asserted, the timer starts counting up from zero and normal device operation begins. When $\overline{\text{RESET}}$ is asserted any time during operation (other than POR and low-power reset), the counter chain is cleared.

8.5 Computer Operating Properly (COP) Reset

The COP watchdog timer function is implemented on this device by using the output of the RTI circuit and further dividing it by eight. The minimum COP reset rates are listed in **Table 8-1**. If the COP circuit times out, an internal reset is generated and the normal reset vector is fetched. Preventing a COP time out or clearing the COP is accomplished by writing a 0 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.

8.6 Timer During Wait Mode

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

Section 9. Carrier Modulator Transmitter (CMT)

9.1 Contents

9.2 Introduction

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.

9.3 Overview

The module consists of carrier generator, modulator, and transmitter output blocks. The block diagram is shown in **Figure 9-1**.

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 **9.4 Carrier Generator**.

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

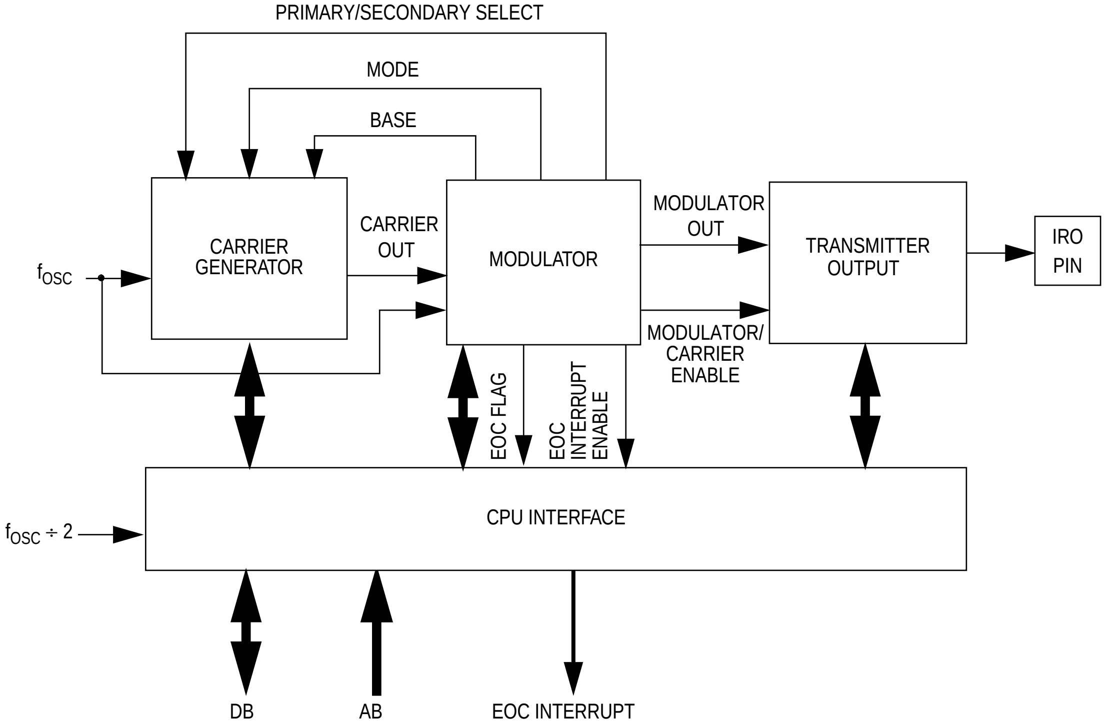


Figure 9-1. Carrier Modulator Transmitter Module Block Diagram

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 **9.5 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 **9.4.2 Carrier Generator Data Registers**.

9.4 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 9-2**.

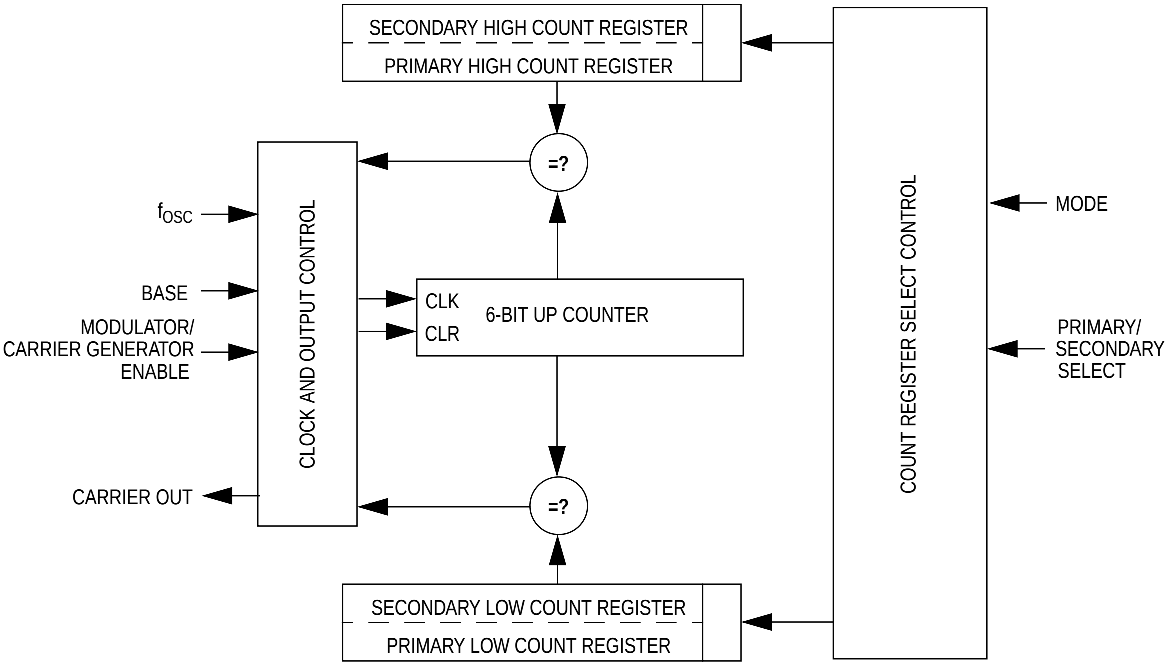


Figure 9-2. Carrier Generator Block Diagram

9.4.1 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.

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

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

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

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

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

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.

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

9.4.2 Carrier Generator Data Registers

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.

CHR1 Address: $0010

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

U = Unaffected

CLR1 Address: $0011

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

U = Unaffected

CHR2 Address: $0012

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

U = Unaffected

CLR2 Address: $0013

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

U = Unaffected

Figure 9-3. Carrier Data Register (CHR1, CLR1, CHR2, and CLR2)

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 **9.5.1 Time Mode**), this register pair is always selected. When operating in FSK mode (see **9.5.2 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 nonzero 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 **9.5.1 Time Mode**), this register pair is never selected. When operating in FSK mode (see **9.5.2 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 nonzero 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.*

9.5 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.

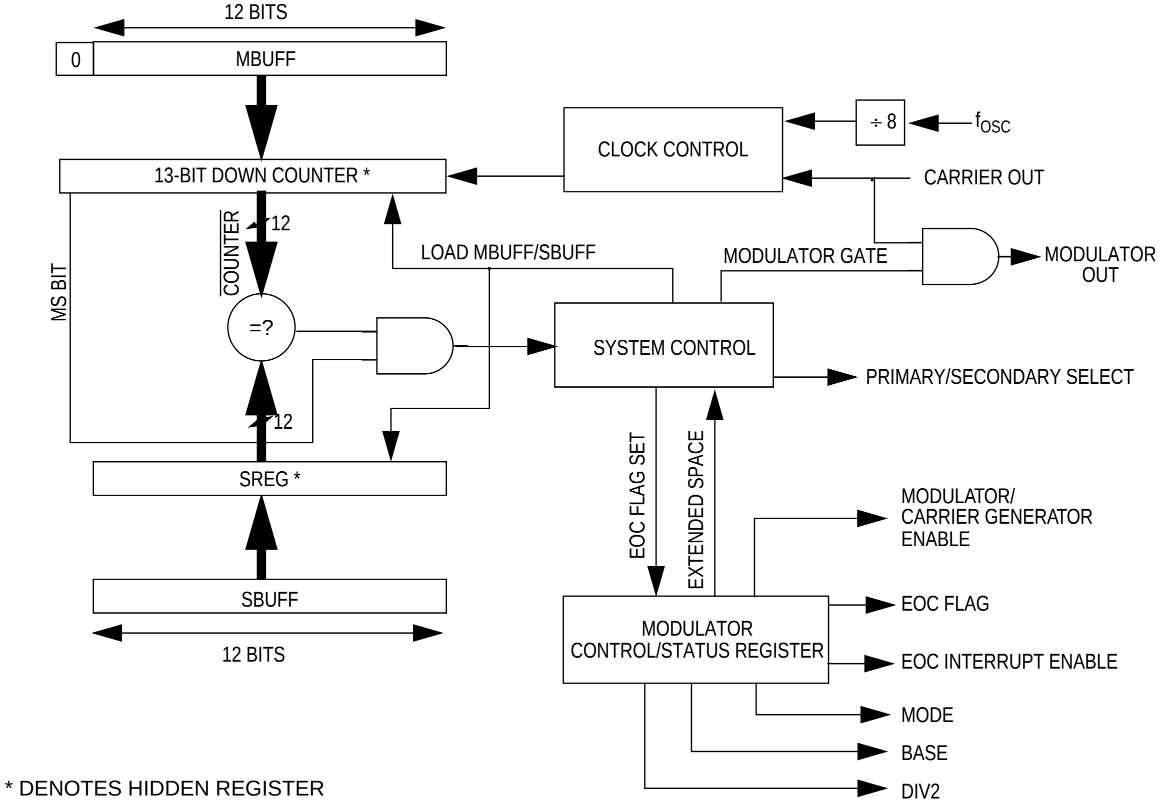


Figure 9-4. Modulator Block Diagram

9.5.1 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 1 for the duration of the mark period and at a logic 0 for the duration of a space period. See **Figure 9-5**.

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.

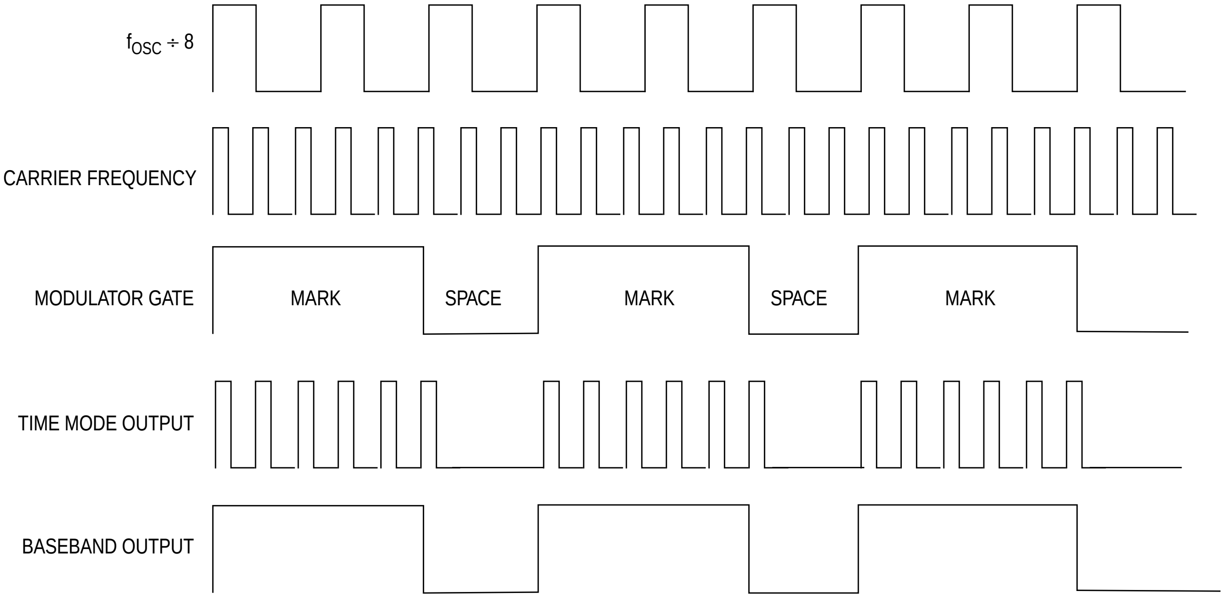


Figure 9-5. CMT Operation in Time Mode

9.5.2 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 automatically can 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.

9.5.3 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:

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

Where f_{cg} is the frequency output from the carrier generator. For an example of extended space operation, see **Figure 9-6**.

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

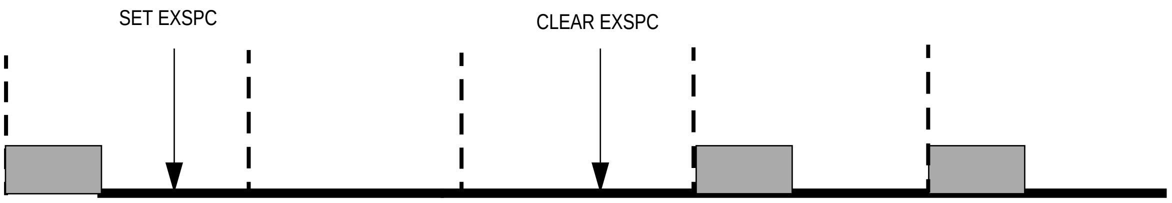


Figure 9-6. Extended Space Operation

9.5.3.1 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.

9.5.3.2 Modulator Control and Status Register

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.

Address: $0014

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

▨ = Unimplemented

Figure 9-7. 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 **9.5.3 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 **9.5.1 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 **9.5.1 Time Mode**. For a description of CMT operation in FSK mode, see **9.5.2 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.

9.5.4 Modulator Period Data Registers

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.

MDR1 Address: $0015

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	MB11	MB10	MB9	MB8	SB11	SB10	SB9	SB8
Reset:	Unaffected by Reset							

MDR2 Address: $0016

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Reset:	Unaffected by Reset							

MDR3 Address: $0017

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SB7	SB6	SB5	SB4	SB3	SB2	SB1	SB0
Reset:	Unaffected by Reset							

Figure 9-8. Modulator Data Registers (MDR1, MDR2, and MDR3)

Section 10. Phase-Locked Loop (PLL) Synthesizer

10.1 Contents

10.2 Introduction

The phase-locked loop (PLL) consists of a variable bandwidth loop filter, a voltage controlled oscillator (VCO), a feedback frequency divider, and a digital phase detector. The PLL requires an external loop filter capacitor (typically 0.1 µF) connected between XFC and V_{DDSYN}. This capacitor should be located as close to the chip as possible to minimize noise. V_{DDSYN} is the supply source for the PLL and should be bypassed to minimize noise. The V_{DDSYN} bypass cap should be as close as possible to the chip. V_{DDSYN} should be at the same potential as V_{DD}.

The phase detector compares the frequency and phase of the feedback frequency (t_{FB}) and the crystal oscillator reference frequency (t_{REF}) and generates the output, PCOMP, as shown in **Figure 10-1**. The output waveform is then integrated and amplified. The resultant dc voltage is applied to the voltage controlled oscillator. The output of the VCO is divided by a variable frequency divider of 128, 64, 32, or 16 to provide the feedback frequency for the phase detector.

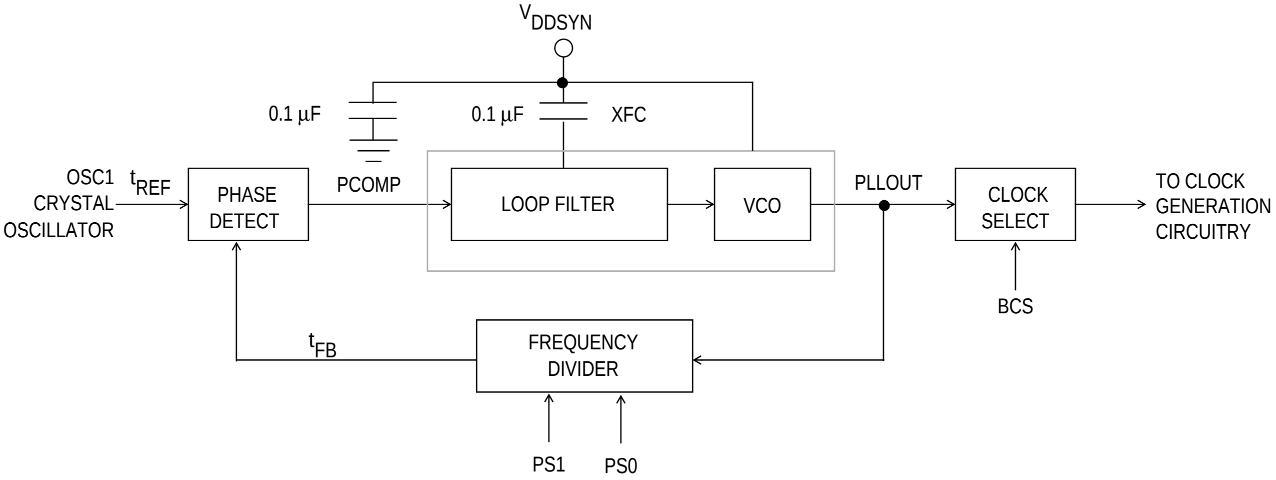


Figure 10-1. PLL Circuit

To change PLL frequencies, follow this 6-step procedure:

1. Clear BCS to enable the low frequency bus rate
2. Clear PLLON to disable the PLL and select manual high bandwidth
3. Select the speed using PS1 and PS0
4. Set PLLON to enable the PLL
5. Wait a time of 90% t_{PLLS} for the PLL frequency to stabilize and select manual low bandwidth, wait another 10% t_{PLLS}

NOTE: *Typically, t_{PLLS} equals 10 ms.*

6. Set BCS to switch to the high-frequency bus rate

The user cannot switch among the high speeds with the BCS bit set. Following the procedure above will prevent possible bursts of high frequency operation during the re-configuration of the PLL.

Whenever the PLL is first enabled, the wide bandwidth mode is used. This enables the PLL frequency to ramp up quickly. When the output frequency is near the desired frequency, the filter is switched to the narrow bandwidth mode to make the final frequency more stable.

10.3 Phase-Locked Loop Control Register

This read/write register contains the control bits that select the PLL frequency and enable/disable the synthesizer.

Address: $0007

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	0	BCS	0	BWC	PLLON	VCOTST	PS1	PS0
Reset:	0	0	0	0	1	1	0	1

Figure 10-2. Phase-Locked Loop Control Register (PLLCR)

BCS — Bus Clock Select

When this bit is set, the output of the PLL is used to generate the internal processor clock. When clear, the internal bus clock is driven by the crystal (OSC1÷2). Once BCS has been changed, it may take up to 1.5 OSC1 cycles + 1.5 PLLOUT cycles to make the transition. During the transition, the clock select output will be held low and all CPU and timer activity will cease until the transition is complete. Before setting BCS, allow at least a time of t_{PLLS} after PLLON is set. This bit should not be set unless the PLLON bit is already set on a previous instruction. Reset clears this bit.

BWC — Bandwidth Control

This bit selects high bandwidth control when set and low bandwidth control when clear. The low bandwidth driver is always enabled, so this bit determines whether the high bandwidth driver is on or off. When the PLL is turned on, the BWC bit should be set to 1 for a time 90% t_{PLLS} to allow the PLL time to acquire a frequency close to the desired frequency. The BWC bit should then be cleared and software should delay for a time, 10% t_{PLLS}, to allow the PLL time to make the final adjustments. The PLL clock cannot be used (BCS bit set). Although it is NOT prohibited in hardware, the BCS bit should not be set unless the BWC bit is cleared and the proper delay times have been followed. The PLL will generate a lower jitter clock when the BWC bit is cleared. Reset clears this bit.

PLLON — PLL On

This bit activates the synthesizer circuit without connecting it to the control circuit. This allows the synthesizer to stabilize before it can drive the CPU clocks. When this bit is cleared, the PLL is shut off and the BCS bit cannot be set. (Setting the BCS bit would engage the disabled PLL onto the bus). Reset sets this bit.

NOTE: *For minimum current consumption, disable the PLL module before entering wait mode.*

NOTE: *The PLLON bit should not be cleared unless the BCS bit has been cleared on a previous write to the register.*

VCOTST — VCO Test

NOTE: *This bit is intended for use by Motorola. This bit cannot be cleared in user mode.*

PS1 and PS0 — PLL Synthesizer Speed Select

These two bits select one of four taps from the PLL to drive the CPU clocks. These bits are used in conjunction with PLLON and BCS bits in the PLL control register. Reset clears PS1 and sets PS0, choosing a bus clock frequency of 524 kHz using an external crystal of 32.768 kHz.

CAUTION: *This bit should not be modified if BCS and PLLON in the PLLCR are both at a logic high.*

Table 10-1. PS1 and PS0 Speed Selects with 32.768-kHz Crystal

PS1	PS0	CPU Bus Clock Frequency (f_{OP})	
0	0	524 kHz	
0	1	1.049 MHz	Reset Condition
1	0	2.097 MHz	For 3.0 V ≤ V_{DD} ≤ 5.5 V
1	1	4.194 MHz	Do Not Select

NOTE: *For the MC68HC705RC17, the 4.194-MHz bus clock frequency should* ***not*** *be selected. The 2.097-MHz bus clock frequency should not be selected when running the part below* $V_{DD} = 3.0$ *V.*

10.4 Operation During Stop Mode

The PLL is switched to low-frequency bus rate and is turned off temporarily when STOP is executed. Coming out of stop mode with an external IRQ, the PLL is turned on with the same configuration it had before going into STOP with the exception of BCS which is reset. Otherwise, the PLL control register is in the reset condition.

10.5 Noise Immunity

The MCU should be insulated as much as possible from noise in the system. These steps are recommend to help prevent problems due to noise injection.

1. The application environment should be designed so that the MCU is not near signal traces which switch often, such as a clock signal.
2. The oscillator circuit for the MCU should be placed as close as possible to the OSC1 and OSC2 pins on the MCU.
3. To minimize noise, all power pins should be filtered by using bypass capacitors placed as close as possible to the MCU.

See the application note *Designing for Electromagnetic Compatibility (EMC) with HCMOS Microcontrollers*, available through the Motorola Literature Distribution Center, document number AN1050/D.

Section 11. EPROM

11.1 Contents

11.2 Introduction

This section describes erasable programmable read-only memory (EPROM) programming.

11.3 EPROM

The user EPROM consists of 15,920 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 $0020 to $00AF, $0100 to $0170, and from $3FB0 to $3FEF.

Ten of the user vectors, $3FF6–$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.

11.4 Bootloader

This program (contained in an on-chip boot ROM) handles copying of user code from an external EPROM into the on-chip EPROM. The bootloader function does not have to be done from an external EPROM, but can be done from a host.

11.4.1 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 that is driving EPROM data into the MCU. The programmer/host interface is shown in **Figure 11-1**.

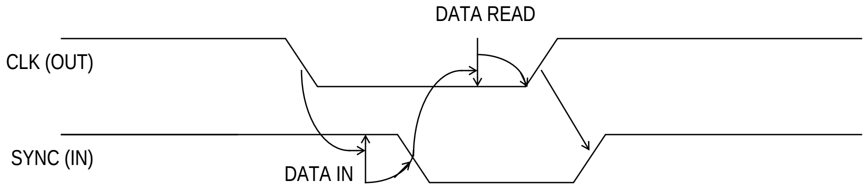


Figure 11-1. Programmer Interface to Host

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 11-1**.

Table 11-1. Bootloader Functions

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

A program flow for the bootloader software included in the MC68HC705RC17 boot ROM is shown in **Figure 11-3**. The bootloader programming board shown in **Figure 11-4** 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.*

11.4.2 Programming Register

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}.

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	R	R	R	R	R	LATCH	R	EPGM
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 11-2. Programming Register (PROG)

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

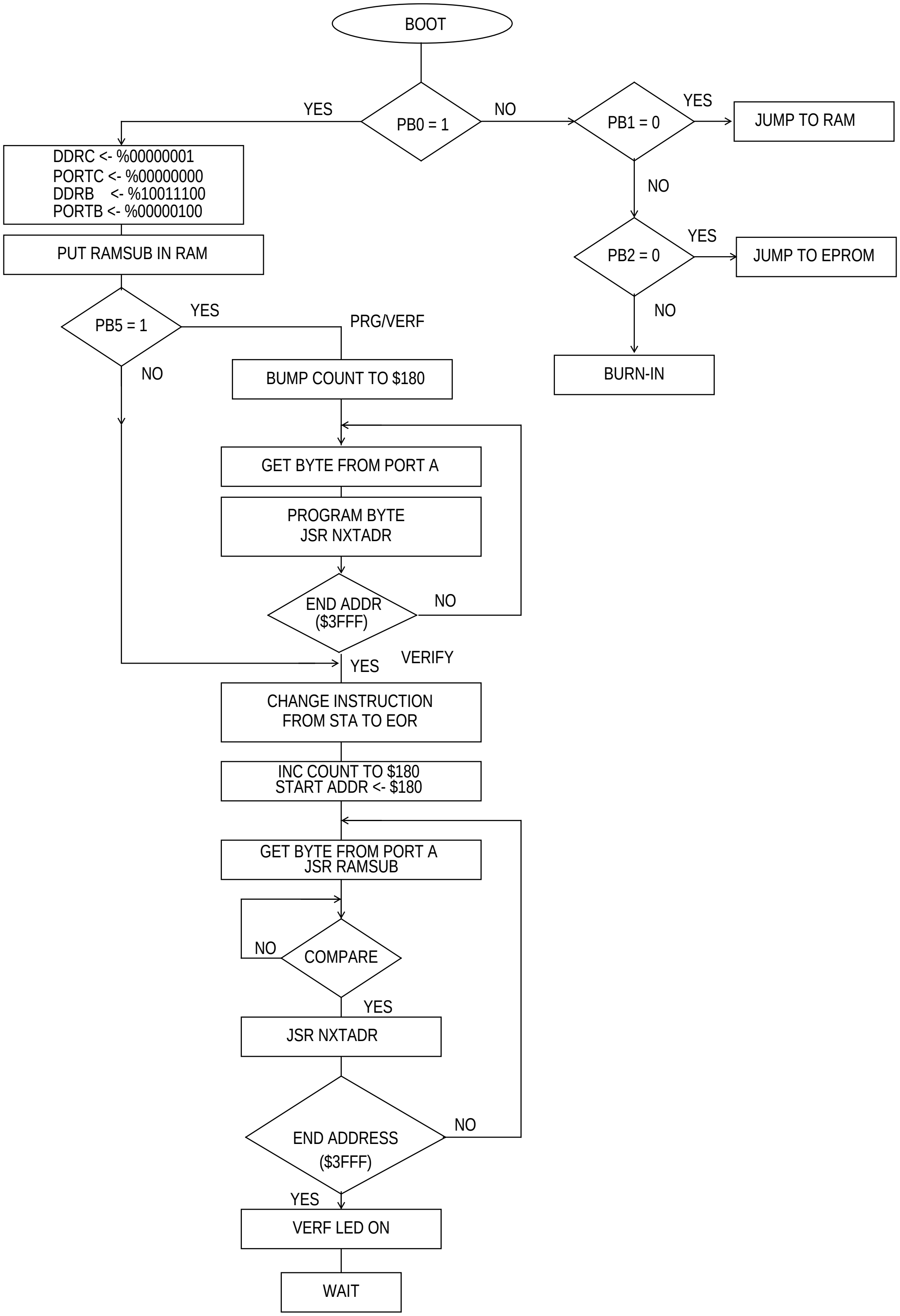
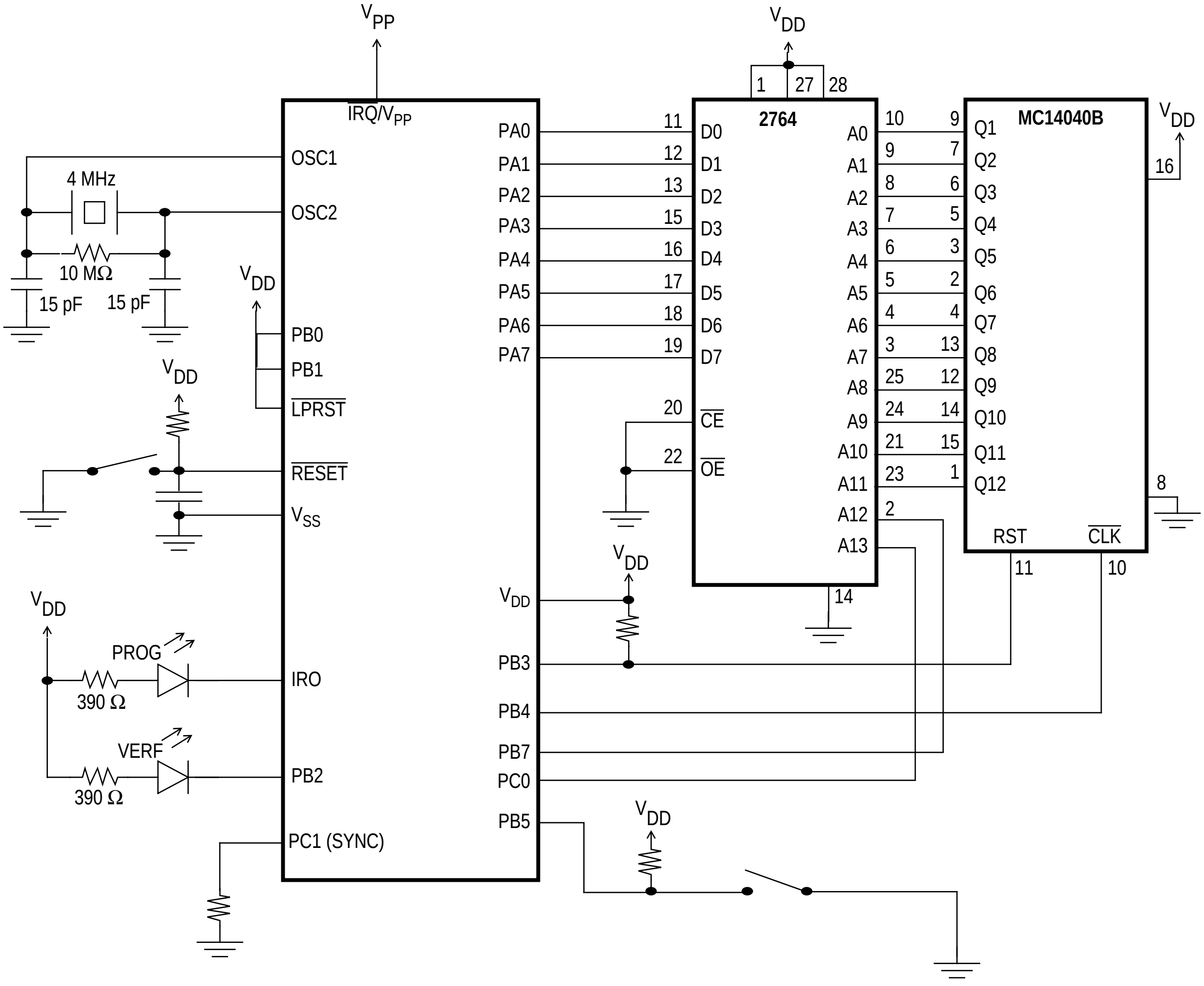


Figure 11-3. MC68HC705RC17 Bootloader Flowchart



NOTE:
All resistors are 10 kΩ unless specified otherwise.

Figure 11-4. MC68HC705RC17 Programming Circuit

11.4.3 Mask Option Registers (MOR1 and MOR2)

The mask option registers contain programmable EPROM bits to control mask options. The MOR register is latched at the end of reset and refreshed periodically depending on how often the EPROM is read.

Address: $3FF0

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	▨	▨	▨	▨	SECUR	IRQ	STOP	COP
Write:								
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 11-5. Mask Option Register 1 (MOR1)

Address: $3FF1

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PB7PU	PB6PU	PB5PU	PB4PU	PB3PU	PB2PU	PB1PU	PB0PU
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 11-6. Mask Option Register 2 (MOR2)

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.*

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.

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

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.

NOTE: *The port B keyscan interrupt sensitivity will match that of the IRQ sensitivity. (See* ***4.7 External Interrupt (IRQ/Port B Keyscan)*** *for more information.)*

Section 12. Instruction Set

12.1 Contents

12.2 Introduction

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

12.3 Addressing Modes

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

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

12.3.1 Inherent

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

12.3.2 Immediate

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

12.3.3 Direct

Direct instructions can access any of the first 256 memory locations with two bytes. The first byte is the opcode, and the second is the low byte of the operand address. In direct addressing, the CPU automatically uses $00 as the high byte of the operand address.

12.3.4 Extended

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

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

12.3.5 Indexed, No Offset

Indexed instructions with no offset are 1-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the effective address of the operand. The CPU automatically uses $00 as the high byte, so these instructions can address locations $0000–$00FF.

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

12.3.6 Indexed, 8-Bit Offset

Indexed, 8-bit offset instructions are 2-byte instructions that can access data with variable addresses within the first 511 memory locations. The CPU adds the unsigned byte in the index register to the unsigned byte following the opcode. The sum is the effective address of the operand. These instructions can access locations $0000–$01FE.

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

12.3.7 Indexed,16-Bit Offset

Indexed, 16-bit offset instructions are 3-byte instructions that can access data with variable addresses at any location in memory. The CPU adds the unsigned byte in the index register to the two unsigned bytes following the opcode. The sum is the effective address of the operand. The first byte after the opcode is the high byte of the 16-bit offset; the second byte is the low byte of the offset.

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

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

12.3.8 Relative

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

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

12.4 Instruction Types

The MCU instructions fall into the following five categories:

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

12.4.1 Register/Memory Instructions

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

Table 12-1. 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

12.4.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to the memory location or to the register.

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

Table 12-2. Read-Modify-Write Instructions

Instruction	Mnemonic
Arithmetic Shift Left (Same as LSL)	ASL
Arithmetic Shift Right	ASR
Bit Clear	BCLR(1)
Bit Set	BSET(1)
Clear Register	CLR
Complement (One's Complement)	COM
Decrement	DEC
Increment	INC
Logical Shift Left (Same as ASL)	LSL
Logical Shift Right	LSR
Negate (Two's Complement)	NEG
Rotate Left through Carry Bit	ROL
Rotate Right through Carry Bit	ROR
Test for Negative or Zero	TST(2)

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

12.4.3 Jump/Branch Instructions

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

The BRCLR and BRSET instructions cause a branch based on the state of any readable bit in the first 256 memory locations. These 3-byte instructions use a combination of direct addressing and relative addressing. The direct address of the byte to be tested is in the byte following the opcode. The third byte is the signed offset byte. The CPU finds the effective branch destination by adding the third byte to the program counter if the specified bit tests true. The bit to be tested and its condition (set or clear) is part of the opcode. The span of branching is from –128 to +127 from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register.

Table 12-3. 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

12.4.4 Bit Manipulation Instructions

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

Table 12-4. Bit Manipulation Instructions

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

12.4.5 Control Instructions

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

Table 12-5. 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

12.5 Instruction Set Summary

Table 12-6. Instruction Set Summary

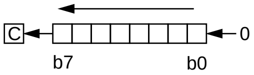
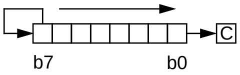
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕×	—	↕×	↕×	↕×	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X	Add without Carry	A ← (A) + (M)	↕×	—	↕×	↕	↕	IMM DIR EXT IX2 IX1 IX	AB BB CB DB EB FB	ii dd hh ll ee ff ff	2 3 4 5 4 3
AND #*opr* AND *opr* AN*D opr* AND *opr*,X AND *opr*,X AND ,X	Logical AND	A ← (A) ∧ (M)	—	—	↕×	↕	—	IMM DIR EXT IX2 IX1 IX	A4 B4 C4 D4 E4 F4	ii dd hh ll ee ff ff	2 3 4 5 4 3
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X	Arithmetic Shift Left (Same as LSL)		—	—	↕×	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff	5 3 3 6 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR ,X	Arithmetic Shift Right		—	—	↕×	↕	↕	DIR INH INH IX1 IX	37 47 57 67 77	dd ff	5 3 3 6 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BCLR *n opr*	Clear Bit n	Mn ← 0	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? Z = 1	—	—	—	—	—	REL	27	rr	3
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BIH *rel*	Branch if IRQ Pin High	PC ← (PC) + 2 + *rel* ? IRQ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if IRQ Pin Low	PC ← (PC) + 2 + *rel* ? IRQ = 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	2 3 4 5 4 3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? C ∨ Z = 1	—	—	—	—	—	REL	23	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? I = 0	—	—	—	—	—	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? N = 1	—	—	—	—	—	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? I = 1	—	—	—	—	—	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? Z = 0	—	—	—	—	—	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? N = 0	—	—	—	—	—	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel* ? 1 = 1	—	—	—	—	—	REL	20	rr	3
BRCLR *n opr rel*	Branch if Bit n Clear	PC ← (PC) + 2 + *rel* ? Mn = 0	—	—	—	—	↕×	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3
BRSET *n opr rel*	Branch if Bit n Set	PC ← (PC) + 2 + *rel* ? Mn = 1	—	—	—	—	↕×	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BSET *n opr*	Set Bit n	Mn ← 1	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	—	—	—	—	—	REL	AD	rr	6
CLC	Clear Carry Bit	C ← 0	—	—	—	—	0	INH	98		2
CLI	Clear Interrupt Mask	I ← 0	—	0	—	—	—	INH	9A		2

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
CLR *opr* CLRA CLRX CLR *opr*,X CLR ,X	Clear Byte	M ← $00 A ← $00 X ← $00 M ← $00 M ← $00	—	—	0	1	—	DIR INH INH IX1 IX	3F 4F 5F 6F 7F	dd ff 	5 3 3 6 5
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X	Compare Accumulator with Memory Byte	(A) – (M)	—	—	↕×	↕	↕	IMM DIR EXT IX2 IX1 IX	A1 B1 C1 D1 E1 F1	ii dd hh ll ee ff ff 	2 3 4 5 4 3
COM *opr* COMA COMX COM *opr*,X COM ,X	Complement Byte (One's Complement)	$M \leftarrow (\overline{M}) = \$FF - (M)$ $A \leftarrow (\overline{A}) = \$FF - (A)$ $X \leftarrow (\overline{X}) = \$FF - (X)$ $M \leftarrow (\overline{M}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$	—	—	↕×	↕×	1	DIR INH INH IX1 IX	33 43 53 63 73	dd ff 	5 3 3 6 5
CPX #*opr* CPX *opr* CPX *opr* CPX *opr*,X CPX *opr*,X CPX ,X	Compare Index Register with Memory Byte	(X) – (M)	—	—	↕×	↕×	↕×	IMM DIR EXT IX2 IX1 IX	A3 B3 C3 D3 E3 F3	ii dd hh ll ee ff ff 	2 3 4 5 4 3
DEC *opr* DECA DECX DEC *opr*,X DEC ,X	Decrement Byte	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1	—	—	↕×	↕×	—	DIR INH INH IX1 IX	3A 4A 5A 6A 7A	dd ff 	5 3 3 6 5
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X	EXCLUSIVE OR Accumulator with Memory Byte	A ← (A) ⊕ (M)	—	—	↕×	↕	—	IMM DIR EXT IX2 IX1 IX	A8 B8 C8 D8 E8 F8	ii dd hh ll ee ff ff 	2 3 4 5 4 3
INC *opr* INCA INCX INC *opr*,X INC ,X	Increment Byte	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1	—	—	↕×	↕×	—	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff 	5 3 3 6 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Unconditional Jump	PC ← Jump Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff 	2 3 4 3 2

Table 12-6. Instruction Set Summary (Continued)

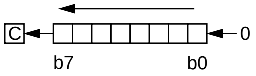
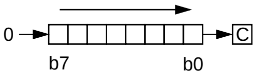
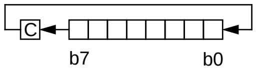
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Effective Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	5 6 7 6 5
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X	Load Accumulator with Memory Byte	A ← (M)	—	—	↕×	↕	—	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff	2 3 4 5 4 3
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X	Load Index Register with Memory Byte	X ← (M)	—	—	↕×	↕×	—	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff	2 3 4 5 4 3
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X	Logical Shift Left (Same as ASL)		—	—	↕×	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff	5 3 3 6 5
LSR *opr* LSRA LSRX LSR *opr*,X LSR ,X	Logical Shift Right		—	—	0	↕	↕	DIR INH INH IX1 IX	34 44 54 64 74	dd ff	5 3 3 6 5
MUL	Unsigned Multiply	X : A ← (X) × (A)	0	—	—	—	0	INH	42		11
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (Two's Complement)	M ← –(M) = \$00 – (M) A ← –(A) = \$00 – (A) X ← –(X) = \$00 – (X) M ← –(M) = \$00 – (M) M ← –(M) = \$00 – (M)	—	—	↕×	↕	↕	DIR INH INH IX1 IX	30 40 50 60 70	dd ff	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	↕×	↕	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	↕×	↕	↕	DIR INH INH IX1 IX	39 49 59 69 79	dd ff	5 3 3 6 5

Table 12-6. Instruction Set Summary (Continued)

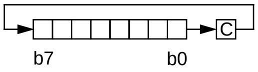
Source Form	Operation	Description	Effect on CCR H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕×	↕	↕	DIR INH INH IX1 IX	36 46 56 66 76	dd ff 	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕×	↕	↕	↕	↕	INH	80		9
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	—	—	—	—	—	INH	81		6
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	↕×	↕	↕	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff 	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X	Store Accumulator in Memory	M ← (A)	—	—	↕×	↕	—	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff 	4 5 6 5 4
STOP	Stop Oscillator and Enable IRQ Pin		—	0	—	—	—	INH	8E		2
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X	Store Index Register In Memory	M ← (X)	—	—	↕×	↕	—	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff 	4 5 6 5 4
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X	Subtract Memory Byte from Accumulator	A ← (A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff 	2 3 4 5 4 3
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		10
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
TST *opr*	Test Memory Byte for Negative or Zero	(M) – $00	—	—	↕	↕	—	DIR	3D	dd	4
TSTA								INH	4D		3
TSTX								INH	5D		3
TST *opr*,X								IX1	6D	ff	5
TST ,X								IX	7D		4
TXA	Transfer Index Register to Accumulator	A ← (X)	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	0×	—	—	—	INH	8F		2

A	Accumulator
C	Carry/borrow flag
CCR	Condition code register
dd	Direct address of operand
dd rr	Direct address of operand and relative offset of branch instruction
DIR	Direct addressing mode
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing
EXT	Extended addressing mode
ff	Offset byte in indexed, 8-bit offset addressing
H	Half-carry flag
hh ll	High and low bytes of operand address in extended addressing
I	Interrupt mask
ii	Immediate operand byte
IMM	Immediate addressing mode
INH	Inherent addressing mode
IX	Indexed, no offset addressing mode
IX1	Indexed, 8-bit offset addressing mode
IX2	Indexed, 16-bit offset addressing mode
M	Memory location
N	Negative flag
n	Any bit

opr	Operand (one or two bytes)
PC	Program counter
PCH	Program counter high byte
PCL	Program counter low byte
REL	Relative addressing mode
rel	Relative program counter offset byte
rr	Relative program counter offset byte
SP	Stack pointer
X	Index register
Z	Zero flag
#	Immediate value
∧	Logical AND
∨	Logical OR
⊕	Logical EXCLUSIVE OR
( )	Contents of
–( )	Negation (two's complement)
←	Loaded with
?	If
:	Concatenated with
↕	Set or cleared
—	Not affected

12.6 Opcode Map

See **Table 12-7**.

Table 12-7. Opcode Map

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

INH = Inherent
IMM = Immediate
DIR = Direct
EXT = Extended

REL = Relative
IX = Indexed, No Offset
IX1 = Indexed, 8-Bit Offset
IX2 = Indexed, 16-Bit Offset

	MSB / LSB	0	MSB of Opcode in Hexadecimal
LSB of Opcode in Hexadecimal	0	5 BRSET0 3 DIR	Number of Cycles Opcode Mnemonic Number of Bytes/Addressing Mode

Section 13. Electrical Specifications

13.1 Contents

13.2 Introduction

This section contains the electrical and timing specifications.

13.3 Maximum Ratings

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

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

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

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to* ***13.6 DC Electrical Characteristics (5.0 Vdc)*** and ***13.7 DC Electrical Characteristics (3.3 Vdc)*** *for guaranteed operating conditions.*

13.4 Operating Range

Characteristic	Symbol	Value	Unit
Operating Temperature Range MC68HC05RC17 (Standard)	T_A	T_L to T_H 0 to +70	°C

13.5 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic Dual In-Line Package		60	
Small Outline Intergrated Circuit Package		60	

13.6 DC Electrical Characteristics (5.0 Vdc)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage					
I_{Load} = 10.0 µA	V_{OL}	—	—	0.1	V
I_{Load} = –10.0 µA	V_{OH}	V_{DD} –0.1	—	—	
Output High Voltage					
(I_{Load} –2.0 mA) Port A, Port B, Port C (1–7)	V_{OH}	V_{DD} –0.8	V_{DD} –0.2	—	V
(I_{Load} –15.0 mA) IRO		V_{DD} –0.7	V_{DD} –0.2	—	
(I_{Load} –4.0 mA) Port C (Bit 0)		V_{DD} –0.8	V_{DD} –0.2	—	
Output Low Voltage					
(I_{Load} = 3.0 mA) Port A, Port B, Port C (1–7)	V_{OL}	—	0.2	0.4	V
(I_{Load} = 25.0 mA) IRO		—	0.2	0.8	
(I_{Load} = 20.0 mA) Port C (Bit 0)		—	0.2	0.4	
Input High Voltage					
Port A, Port B, Port C, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, $\overline{\text{LPRST}}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage					
Port A, Port B, Port C, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, $\overline{\text{LPRST}}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Input Hysteresis ($\overline{\text{RESET}}$)	V_{HYST}	0.8	0.9	1	V
EPROM Programming Voltage	V_{PP}	14.5	—	15.5	V
Supply Current (see Notes)					
Run (f_{OP} = 2.1 MHz)	I_{DD}	—	TBD	TBD	mA
Wait with PLL Enabled (f_{OP} = 2.1 MHz)		—	TBD	TBD	mA
Wait with PLL Disabled (f_{OP} = 16.384 kHz)		—	TBD	TBD	µA
Stop					
25 ^{o}C		—	0.4	15	µA
0 ^{o}C to +70 ^{o}C		—	0.4	30	µA
I/O Ports Hi-Z Leakage Current					
Port A, Port B, Port C	I_{OZ}	–10	—	10	µA
Input Current					
$\overline{\text{RESET}}$, $\overline{\text{LPRST}}$, $\overline{\text{IRQ}}$, OSC1	I_{In}	–1	—	1	µA
PB0–PB7 with Pullups Enabled (V_{IN} = 0.2 x V_{DD})		–100	–330	–700	
PB0–PB7 with Pullups Enabled (V_{IN} = 0.7 x V_{DD})		–50	–120	–300	
Capacitance					
Ports (as Input or Output)	C_{Out}	—	—	12	pF
$\overline{\text{RESET}}$, $\overline{\text{LPRST}}$, $\overline{\text{IRQ}}$	C_{INT}	—	—	8	

NOTES:

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0 °C to +70 °C, unless otherwise noted
2. All values shown reflect average measurements.
3. All current measurements represent the summation of current through V_{DD} and V_{DDSYN} supply pins.
4. Typical values at midpoint of voltage range, 25 ^{o}C only
5. To minimize current consumption in wait mode, disable the PLL before executing the WAIT instruction. Internal bus speed will be that of the 32.768-kHz external frequency.
6. Wait I_{DD}: only core timer active
7. Run (with PLL enabled) I_{DD}, wait I_{DD} (with PLL enabled): 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. Wait I_{DD} (with PLL disabled): Measured using external square wave clock source (f_{OSC} = 33 kHz).
8. Wait, stop I_{DD}: Port A and port C configured as inputs; port B configured as outputs; V_{IL} = 0.2 V; V_{IH} = V_{DD} –0.2 V
9. Stop I_{DD} is measured with OSC1 = V_{SS}.
10. Wait I_{DD} is affected linearly by the OSC2 capacitance.
11. Pullups are designed to be capable of pulling to V_{IH} within 10 µs for a 100 pF, 4-kΩ load.

13.7 DC Electrical Characteristics (3.3 Vdc)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage					
I_{Load} = 10.0 µA	V_{OL}	—	—	0.1	V
I_{Load} = –10.0 µA	V_{OH}	V_{DD} –0.1	—	—	
Output High Voltage					
(I_{Load} –1.6 mA) Port A, Port B, Port C (1–7)	V_{OH}	V_{DD} –0.3	V_{DD} –0.1	—	V
(I_{Load} –10 mA) IRO		V_{DD} –0.7	V_{DD} –0.1	—	
(I_{Load} –1.2 mA) Port C (Bit 0)		V_{DD} –0.3	V_{DD} –0.1	—	
Output Low Voltage					
(I_{Load} = 1.0 mA) Port A, Port B, Port C (1–7)	V_{OL}	—	0.1	0.3	V
(I_{Load} = 8.0 mA) IRO		—	0.1	0.8	
(I_{Load} = 7.0 mA) Port C (Bit 0)		—	0.1	0.3	
Input High Voltage Port A, Port B, Port C, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, $\overline{\text{LPRST}}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage Port A, Port B, Port C, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, $\overline{\text{LPRST}}$, OSC1	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
EPROM Programming Voltage	V_{PP}	14.5	—	15.5	V
Supply Current (see Notes)					
Run (f_{OP} = 2.1 MHz)	I_{DD}	—	TBD	TBD	mA
Wait with PLL Enabled (f_{OP} = 2.1 MHz)		—	TBD	TBD	mA
Wait with PLL Disabled (f_{OP} = 16.384 kHz)		—	TBD	TBD	µA
Stop					
25 ^{o}C		—	0.15	4.0	µA
0 ^{o}C to +70 ^{o}C		—	0.15	8.0	µA
I/O Ports Hi-Z Leakage Current Port A, Port B, Port C	I_{OZ}	–6	—	6	µA
Input Current					
$\overline{\text{RESET}}$, $\overline{\text{LPRST}}$, $\overline{\text{IRQ}}$, OSC1	I_{In}	–0.6	—	0.6	µA
PB0–PB7 with Pullups Enabled (V_{IN} = 0.3 x V_{DD})		–25	–80	–105	
PB0–PB7 with Pullups Enabled (V_{IN} = 0.7 x V_{DD})		–15	–50	–65	
Capacitance					
Ports (as Input or Output)	C_{Out}	—	—	12	pF
$\overline{\text{RESET}}$, $\overline{\text{LPRST}}$, $\overline{\text{IRQ}}$	C_{INT}	—	—	8	

NOTES:

1. V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0 °C to +70 °C, unless otherwise noted
2. All values shown reflect average measurements.
3. Typical values at midpoint of voltage range, 25 ^{o}C only
4. All current measurements represent the summation of current through V_{DD} and V_{DDSYN} supply pins.
5. To minimize current consumption in wait mode, disable the PLL before executing the WAIT instruction. Internal bus speed will be that of the 32.768-kHz external frequency.
6. Wait I_{DD}: only core timer active
7. Run (with PLL enabled) I_{DD}, wait I_{DD} (with PLL enabled): 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. Wait I_{DD} (with PLL disabled): Measured using external square wave clock source (f_{osc} = 33 kHz).
8. Wait, stop I_{DD}: Port A and port C configured as inputs, port B configured as outputs; V_{IL} = 0.2 V; V_{IH} = V_{DD} –0.2 V
9. Stop I_{DD} is measured with OSC1 = V_{SS}.
10. Wait I_{DD} is affected linearly by the OSC2 capacitance.
11. Pullups are designed to be capable of pulling to V_{IH} within 10 µs for a 100 pF, 4-kΩ load.

13.8 Control Timing (3.3 Vdc and 5.0 Vdc)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Assumes use of 32.768-kHz Crystal Only	f_{OSC}	32.768	32.768	kHz
Internal Operating Frequency Assumes use of 32.768-kHz Crystal Only $f_{OP} = f_{OSC} \div 2$ if PLL is NOT Driving Internal Bus f_{OP} = 32.768 kHz x N, where N = 16, 32, or 64 (Selectable) and PLL is Driving Internal Bus	f_{OP}	— 0.524	16.384 2.097	kHz MHz
Cycle Time Assumes f_{OP} = 2.097 MHz	t_{CYC}	480	—	ns
Crystal Oscillator Startup Time Assumes f_{OP} = 2.097 MHz)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) Assumes f_{OP} = 2.097 MHz)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width Assumes f_{OP} = 2.097 MHz)	t_{RL}	1.5	—	t_{CYC}
Interrupt Pulse Width Low (Edge-Triggered) Assumes f_{OP} = 2.097 MHz)	t_{ILIH}	125	—	ns
Interrupt Pulse Period Assumes f_{OP} = 2.097 MHz)	t_{ILIL}	Note 2	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns
EPROM Byte Programming Time	t_{EPGM}	—	10.0	ms

NOTES:

1. V_{DD} = 3.0 to 5.5 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0 °C to +70 °C, unless otherwise noted
2. The minimum period, t_{ILIL}, should not be less than the number of cycle times it takes to execute the interrupt service routine plus 2T t_{CYC}.

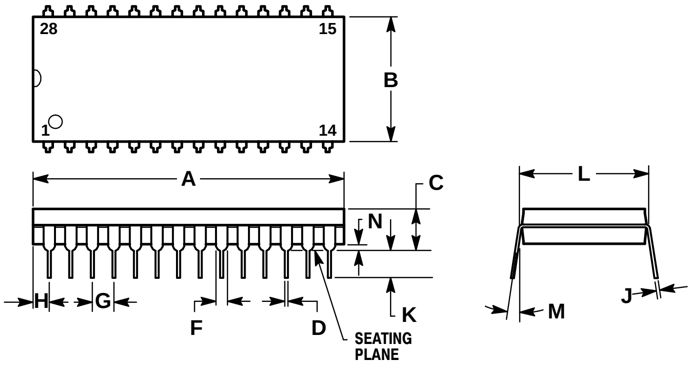
Section 14. Mechanical Specifications

14.1 Contents

14.2 Introduction

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

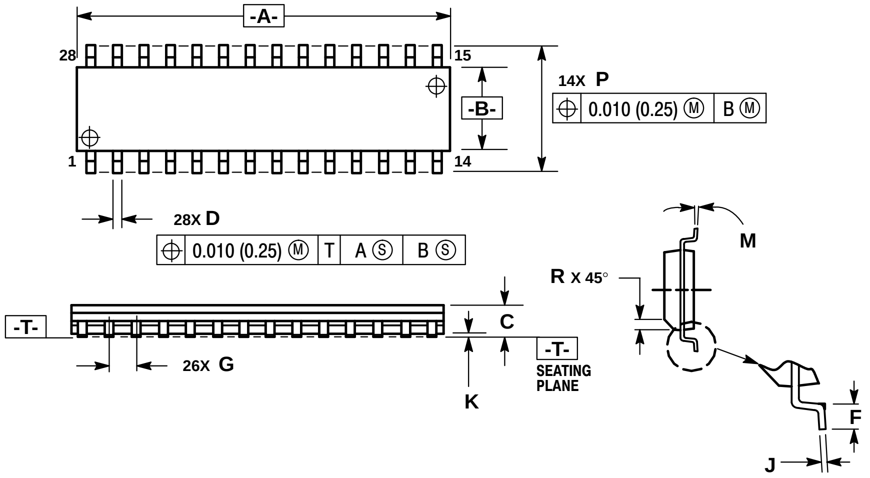
14.3 28-Pin Plastic Dual In-Line Package (Case 710-02)



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.

DIM	MILLIMETERS		INCHES	
	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

14.4 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 15. Ordering Information

15.1 Contents

15.2 Introduction

This section contains ordering instructions for the MC68HC705RC17.

15.3 Ordering Information

Table 15-1 provides information in determing order numbers.

Table 15-1. MC Order Numbers

Package Type	Operating Temperature Range	MC Order Number
28-Pin Plastic Dual In-Line Package (DIP)	0 to 70 °C	MC68HC705RC17P
28-Pin Small Outline Integrated Circuit Package (SOIC)	0 to 70 °C	MC68HC705RC17DW

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

How to reach us:
USA/EUROPE/Locations Not Listed: Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado 80217. 1-800-441-2447 or 303-675-2140
Mfax™: RMFAX0@email.sps.mot.com – TOUCHTONE 602-244-6609, US & Canada ONLY 1-800-774-1848
INTERNET: http://motorola.com/sps
JAPAN: Nippon Motorola Ltd. SPD, Strategic Planning Office 4-32-1, Nishi-Gotanda Shinagawa-ku, Tokyo 141, Japan. 81-3-5487-8488
ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

Mfax is a trademark of Motorola, Inc.

MOTOROLA

© Motorola, Inc., 1997