HC05E1GRS/D
REV. 2.0

68HC05E1

General Release Specification

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.

Table of Contents

Section 1. General Description

Section 2. Operating Modes

Section 3. CPU Core

Section 4. Input/Output Ports

Section 5. Memory

Section 6. Timer, Phase-Locked Loop, and Custom Periodic Interrupt

Section 7. Electrical Specifications

Section 8. Mechanical Specifications

Table of Contents

List of Figures

List of Figures

List of Tables

List of Tables

Section 1. General Description

1.1 Contents

1.2 Introduction

The MC68HC05E1 is a low-cost introduction to the M68HC05 Family of microcontrollers (MCUs). The HC05 CPU core has been enhanced with a 15-stage multi-functional timer and programmable phase-locked loop. The MCU is available in a 28-pin package, and has two 8-bit I/O ports and one 4-bit I/O port. The 8 Kbyte memory map includes 368 bytes of RAM and 4096 bytes of user ROM.

1.3 Features

Features of the MC68HC05E1 include:

- Low cost
- HC05 Core
- 28-pin package
- On-Chip Oscillator (Crystal or Ceramic Resonator)
- Phase-Locked Loop (PLL) Synthesizer with Programmable Speed
- 4112 Bytes of User ROM (including 16 Bytes of User Vectors)
- 368 Bytes of On-Chip RAM
- 15-Stage Multi-functional Timer with Programmable Input
- Real Time Interrupt Circuit
- COP Watchdog Timer Mask Option
- Custom Periodic Interrupt Circuit
- 20 Bidirectional I/O Lines
- Single-Chip Mode
- Self-Check Mode
- Power Saving STOP and WAIT Modes
- Edge-Sensitive or Edge- and Level-Sensitive Interrupt Trigger Mask Option
- STOP Instruction Disable Mask Option
- Illegal Address Reset

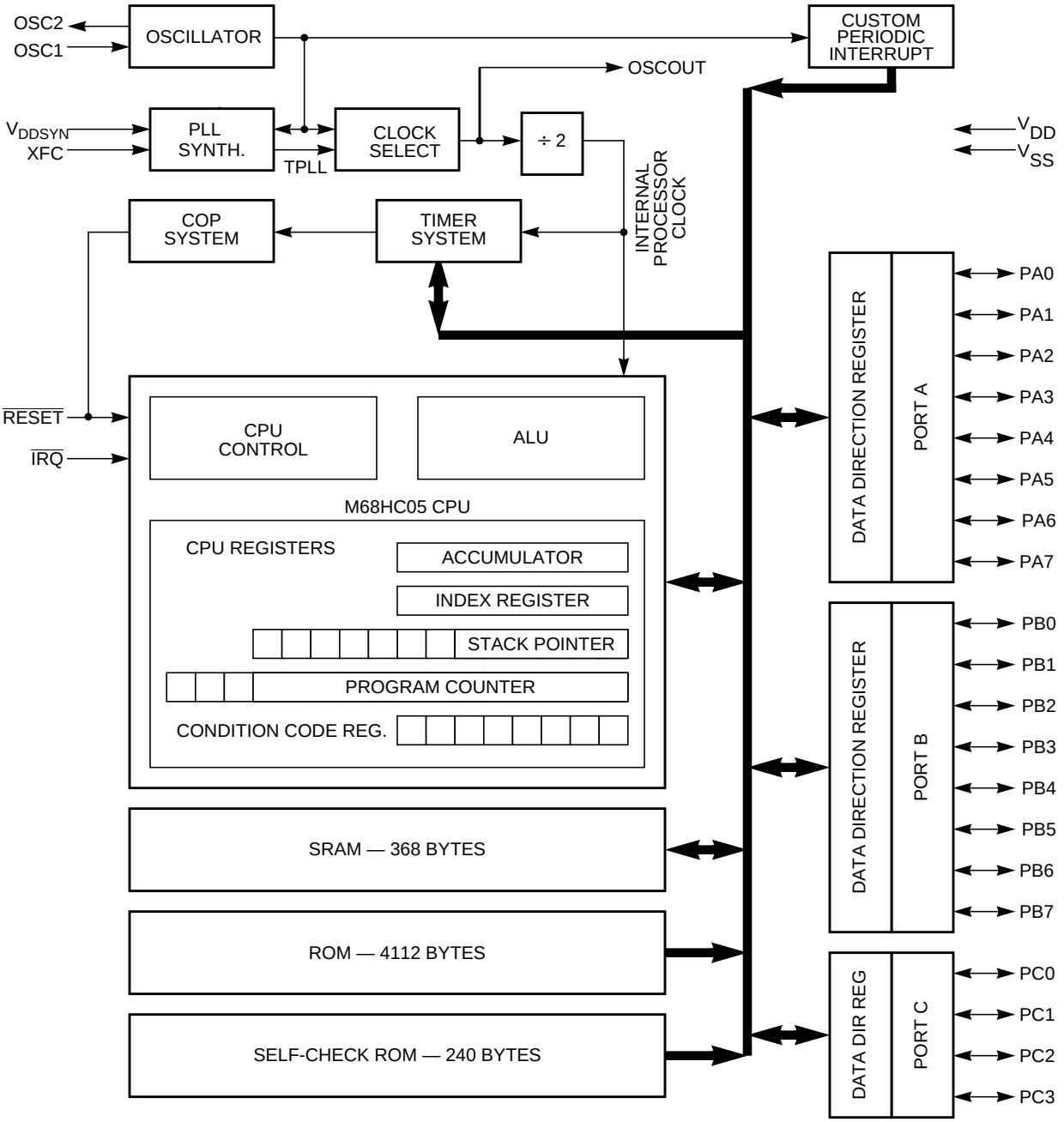


Figure 1-1. Block Diagram of the MC68HC05E1

1.4 Mask Options

There are four mask options on the MC68HC05E1: STOP instruction (enable/disable), $\overline{IRQ}$ (Edge-sensitive only or Edge- and level-sensitive), COP Watchdog Timer (enable/disable), and CPI Rate (1 second, 0.5 second, or 0.25 second).

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

1.5 Functional Pin Description

1.5.1 V_{DD} and V_{SS}

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

1.5.2 $\overline{IRQ}$ (Maskable Interrupt Request)

This pin has a programmable option that provides two different choices of interrupt triggering sensitivity. The options are:

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

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

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

The $\overline{IRQ}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. Refer to **3.7 Interrupts** for more detail.

NOTE: *The voltage on this pin affects the mode of operation. See* ***Section 2. Operating Modes****.*

1.5.3 OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit which can optionally drive a Phase-Locked Loop clock. A crystal, a ceramic resonator, or an external signal connects to these pins providing a system clock. The oscillator frequency is two times the internal bus rate if the PLL is not used.

1.5.3.1 Crystal

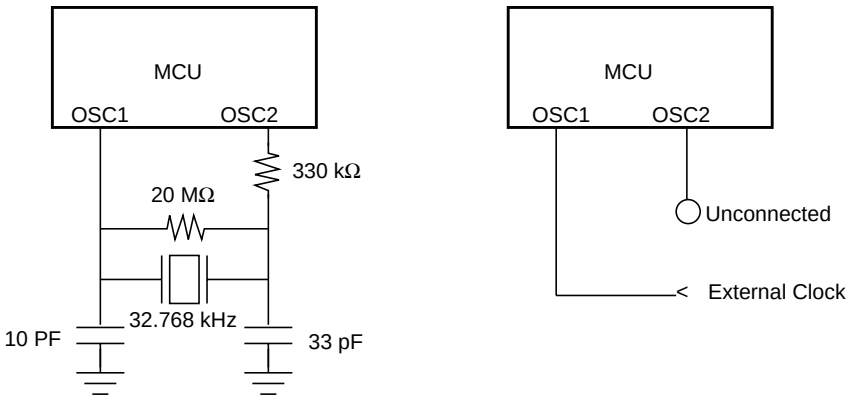
Figure 1-2 shows the recommended circuit for using a crystal. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time.

1.5.3.2 Ceramic Resonator

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. **Figure 1-2** shows the recommended circuit for using a ceramic resonator. The manufacturer of the particular ceramic resonator being considered should be consulted for specific information.

1.5.3.3 External Clock

An external clock should be applied to the OSC1 input with the OSC2 pin not connected. See **Figure 1-2**. This setup can be used if the user does not wish to run the CPU with a 32.768 KHz crystal or the PLL frequencies are not suitable for the application.



(a) Crystal/Ceramic Resonator Oscillator Connections

(b) External Clock Source Connections

Figure 1-2. Oscillator Connections

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

This active low pin is used to reset the MCU to a known start-up state by pulling $\overline{\text{RESET}}$ low. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. See **3.6 Resets**.

1.5.5 PA0-PA7

These eight I/O lines comprise port A. The state of any pin is software programmable and all port A lines are configured as input during power-on or reset. See **4.6 Input/Output Programmingf**.

1.5.6 PB0-PB7

These eight I/O lines comprise port B. The state of any pin is software programmable and all port B lines are configured as input during power-on or reset. See **4.6 Input/Output Programmingf**.

1.5.7 PC0-PC3

These four I/O lines comprise port C. The state of any pin is software programmable and all port C lines are configured as input during power-on or reset. See **4.6 Input/Output Programmingf**.

1.5.8 XFC

This pin provides a means for connecting an external filter capacitor to the synthesizer phase-locked loop filter. See **6.4 Phase-Locked Loop Synthesizer** for additional information concerning this capacitor.

1.5.9 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 and I/O ports should be tied to an appropriate logic level (either V_{DD} or V_{SS}). Although the I/O ports of the MC68HC05E1 do not require termination, it is recommended to reduce the possibility of static damage.*

General Description

Section 2. Operating Modes

2.1 Contents

2.2 Introduction

The MCU has 3 modes of operation: Single-chip mode, Self-Check Mode, and Test Mode. **Table 2-1** shows the conditions required to go into each mode.

Table 2-1. Operating Mode Conditions

$\overline{\text{RESET}}$	$\overline{\text{IRQ}}$	PB1	Mode
	V_{SS}–V_{DD}	V_{SS}–V_{DD}	Single Chip
	V_{TST}	V_{DD}	Self Check
	V_{TST}	V_{SS}	Factory Test

$V_{TST} = 2 \times V_{DD}$

2.3 Single-Chip Mode

In single-chip mode, the address and data buses are not available externally, but there are two 8-bit I/O ports and one 4-bit I/O port. This mode allows the MCU to function as a self-contained microcontroller, with maximum use of the pins for on-chip peripheral functions. All address and data activity occurs within the MCU. Single-Chip Mode is entered on the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}$ pin is within normal operating range.

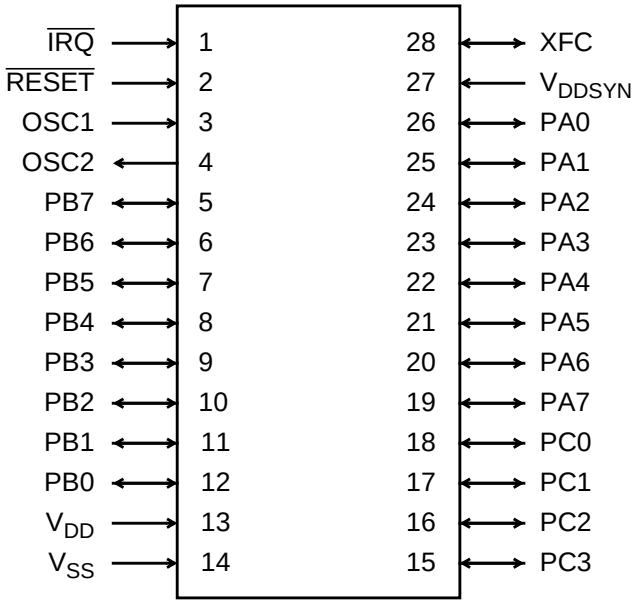


Figure 2-1. Single-Chip Mode Pinout of the MC68HC05E1

2.4 Self-Check Mode

The Self-Check Mode provides an internal check to determine if the device is functional. See **Figure 2-2**.

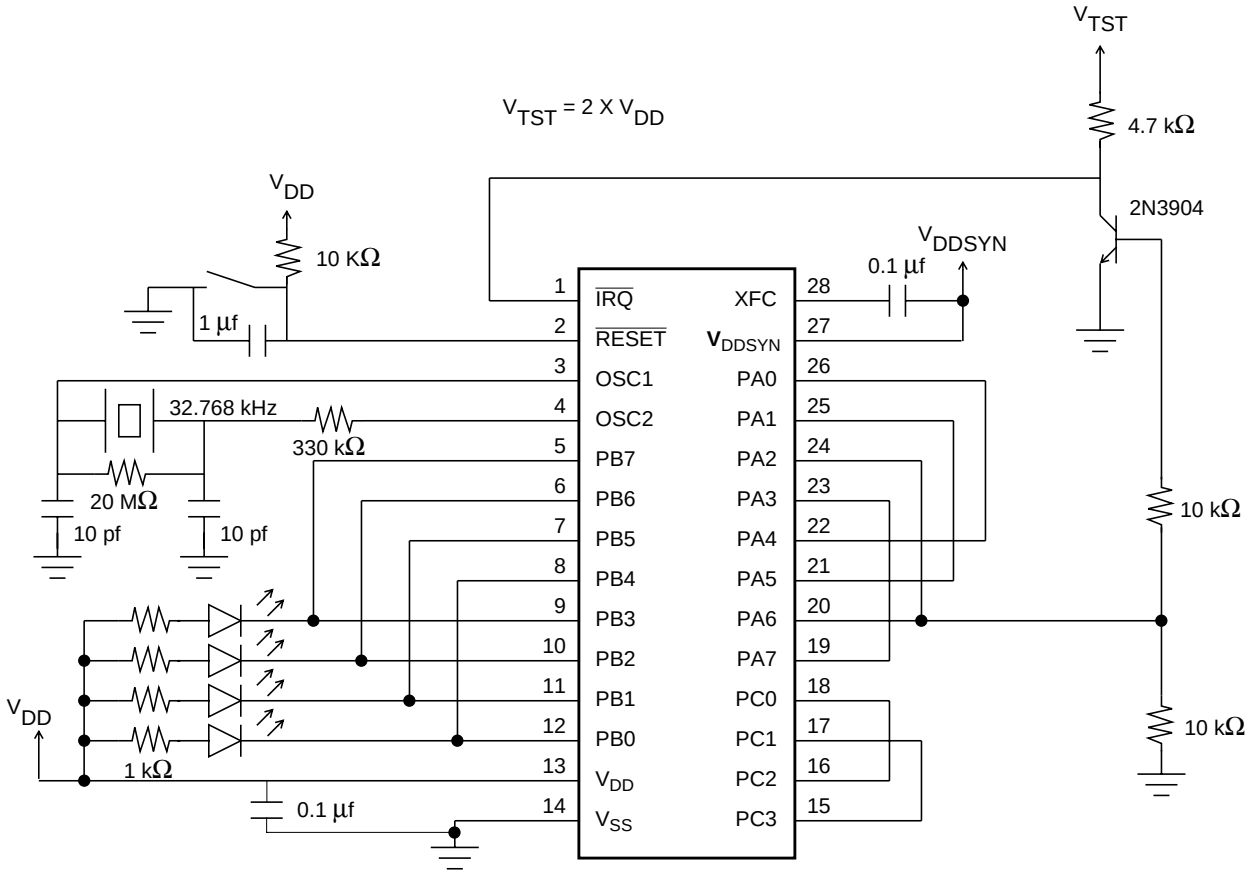


Figure 2-2. Self-Check Circuit Schematic Diagram

The Self-Check Mode is entered on the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}$ pin is at V_{TST}, and the PB1 pin is at logic one. $\overline{\text{RESET}}$ must be held low for 4064 cycles after POR, or for a time t_{RL} for any other reset. After reset, the PLL is turned on (f_{op} = 1.049 MHz) and the following tests are performed automatically:

1. I/O – Functionally exercises ports A, B, and C
2. RAM – Counter test for each page zero RAM byte
3. Timer/CPI – Tracks counter register and checks TOF and RTIF flags
4. ROM – Exclusive OR with odd ones parity result
5. Interrupts –Tests external interrupts, RTI and CPI

Self-check results (using the LEDs as monitors) are shown in **Table 2-2**. The Self-Check program resides at ROM location $1F00 to $1FEF. The following subroutines are available to user programs and do not require any external hardware.

2.4.1 Timer Test Subroutine

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set.

This subroutine is called at location $1F9B. Because the timer is free running and has only a divide-by-four prescaler, each timer count cannot be tested. The test sets RTIE and CPIE and reads the timer once every 3 counts (12 cycles) to check for correct counting. The test tracks the counter until the timer wraps around, setting the TOF bit in the Timer Control and Status register. The routine then waits for RTIF=1 and CPIF = 1 before returning with the RTI and the CPI pending. RAM location $0095 is overwritten. Upon return to the user's program, A=0 if the test passed.

2.4.2 ROM Checksum Subroutine

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set.

This subroutine is called at location $1FD1. A short routine is set up and executed in RAM to compute a checksum of the entire ROM pattern. The checksum byte is computed by Motorola and is located in the Self-Check ROM. Upon return to the user's program, X=0. If the test passed, A=0. RAM locations $0090 through $0093 are overwritten.

Table 2-2. Self-Check Results

PA3	PA2	PA1	PA0	Remarks
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad ROM
1	1	0	1	Bad Interrupts or $\overline{IRQ}$ request
Flashing				Good Device
All Others				Bad Device

0 indicates LED is on; 1 indicates LED is off.

2.4.3 Additional Self-Check Routines

The Self-Check ROM contains additional programs to facilitate testing and characterization of the device. **Figure 2-3** shows the program flow in the Self-Check ROM. These programs are used in Self-Check Mode (PB1=1). On power-up, the device goes into Self-Check Mode on the rising edge of $\overline{RESET}$ if the $\overline{IRQ}$ pin is at V_{TST}, and the PB1 pin is at logic one. The values of PB0:PB3 after power-up determine which routine is executed from the Self-Check ROM. Only the Self-Check routine is intended for customer use.

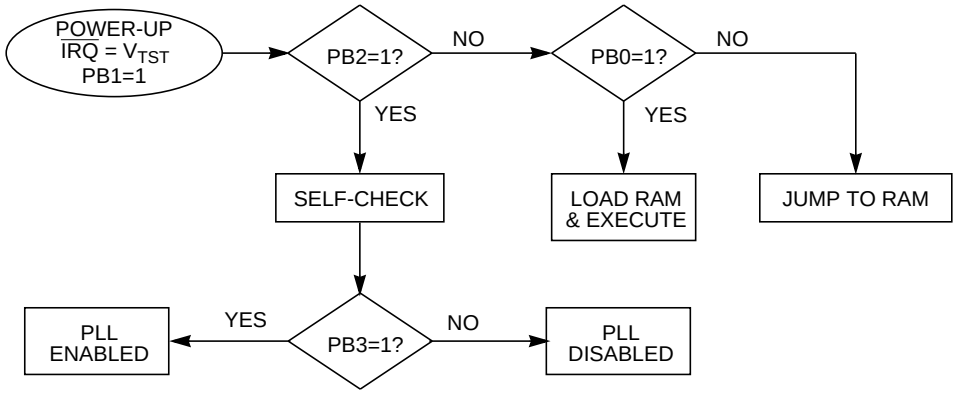


Figure 2-3. Self-Check Mode Flowchart

2.4.3.1 Self-Check PLL Disabled

If PB2=1 and PB3=0, the self-check routine is run without turning on the PLL. This allows the self-check program to run at any frequency, as determined by the value of the crystal oscillator in the self-check circuit.

2.4.3.2 Jump to RAM

This routine is executed if PB2=0, PB1=1, and PB0=0.

This routine jumps to the starting address of the RAM. This is used after a program has been placed in the RAM. This feature is useful for production testing where single-chip timing or port functionality is needed.

2.4.3.3 Load RAM

This routine is entered if PB2=0, PB1=1, and PB0=1.

The **`ldram`** routine does a parallel download of a program into port A using $\overline{\text{IRQ}}$ (data ready) and PC0 (data acknowledge) to synchronize the download with the host system. When $\overline{\text{IRQ}}$ (data ready) goes low, PC0 (data acknowledge) is deasserted and a byte of data is loaded from port A to RAM starting at location $100. After the byte is stored in RAM, PC0 is asserted as an active low data acknowledge signal to the host. The first byte downloaded must contain the total number of bytes to be downloaded (program length +1). When the download is complete, the program in RAM is executed.

Section 3. CPU Core

3.1 Contents

3.2 Introduction

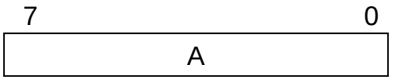
This section describes the CPU core.

3.3 Registers

The MCU contains the registers described in the following paragraphs.

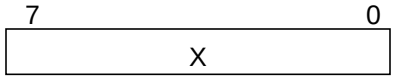
3.3.1 Accumulator (A)

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



3.3.2 Index Register (X)

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



3.3.3 Condition Code Register (CCR)

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

3.3.3.1 Half Carry (H)

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

3.3.3.2 Interrupt (I)

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

3.3.3.3 Negative (N)

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

3.3.3.4 Zero (Z)

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

3.3.3.5 Carry/Borrow (C)

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

3.3.4 Stack Pointer (SP)

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

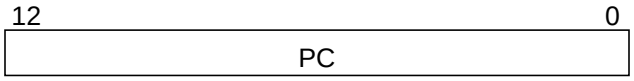
When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce and address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are 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.

12					7		0
0	0	0	0	0	1	1	SP

3.3.5 Program Counter (PC)

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

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



3.4 Instruction Set

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. For more information on the instruction set, refer to the *M6805 Family User's Manual* (M6805UM/AD2) or the *MC68HC05C4 Data Sheet* (MC68HC05C4/D).

3.4.1 Register/Memory Instructions

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR
Multiply	MUL

3.4.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Do not use these read-modify-write instructions on write-only locations. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST

3.4.3 Branch Instructions

This set of instruction branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

3.4.4 Bit Manipulation Instructions

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. These instructions are also read-modify-write instructions. Do not bit manipulate write-only locations. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n = 0. . .7)
Branch if bit n is Clear	BRCLR n (n = 0. . .7)
Set Bit n	BSET n (n = 0. . .7)
Clear Bit n	BCLR n (n = 0. . .7)

3.4.5 Control Instructions

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

3.5 Addressing Modes

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term “effective address” (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

3.5.1 Immediate

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

3.5.2 Direct

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

3.5.3 Extended

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction

uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

3.5.4 Re;atove

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed offset byte (which is the last byte of the instruction) is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from -128 to +127 from the address of the next opcode. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

3.5.5 Indexed, No Offset

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

3.5.6 Indexed, 8-Bit Offset

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510. $1FE is the last location which can be accessed in this way.

3.5.7 Indexed, 16-Bit Offset

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

3.5.8 Bit Set/Clear

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

3.5.9 Bit Test and Branch

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from -128 to +127 from the address of the next opcode. The state of the tested bit is also transferred to the carry bit of the condition code register.

3.5.10 Inherent

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register and/or accumulator as well as the control instructions with no other arguments are included in this mode. These instructions are one byte long.

3.6 Resets

The MCU can be reset three ways: by the initial power-on reset function, by an active low input to the $\overline{\text{RESET}}$ pin, by a COP watchdog-timer reset, and by the ILADR bit being set in the test register.

3.6.1 Power-On Reset (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{cyc}) oscillator stabilization delay after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is low at the end of this 4064 cycle delay, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ goes high.

3.6.2 $\overline{\text{RESET}}$ Pin

The MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period of one and one-half machine cycles (t_{cyc}). $\overline{\text{RESET}}$ is an input-only pin and will not indicate when an internal reset has occurred.

3.6.3 Computer Operating Properly (COP) Reset

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 timeout, an internal reset is generated to reset the MCU. Because the internal reset signal is used, the MCU

comes out of a COP reset in the same operating mode it was in when the COP time-out was generated.

The COP reset function is enabled or disabled by a mask option.

Refer to **6.3.2 Computer Operating Properly (COP) Watchdog Reset**, for more information on the COP Watchdog timer.

3.6.4 Illegal Address Reset

When an opcode fetch occurs from an address which is not implemented in the RAM ($0090–$01FF) or ROM ($0F00–$1FFF), the part is automatically reset.

3.7 Interrupts

The MCU can be interrupted four different ways: the three maskable hardware interrupts ($\overline{\text{IRQ}}$, timer, and CPI) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete.

NOTE: *The current instruction is the one already fetched and being operated on.*

When the current instruction is complete, the processor checks all pending hardware interrupts. If interrupts are not masked (CCR I bit clear) and the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state.

Table 3-1. Vector Address for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	Reset	RESET	$1FFE–$1FFF
N/A	N/A	Software	SWI	$1FFC–$1FFD
N/A	N/A	External Interrupt	RQ	$1FFA–$1FFB
TCSR	TOF	Timer Overflow	TIMER	$1FF8–$1FF9
	RTIF	Real Time Interrupt	IMER	$1FF8–$1FF9
CPICSR	CPIF	Custom Periodic Interrupt	CPI	$1FF6–$1FF7

3.7.1 Hardware Controlled Interrupt Sequence

The following three functions ($\overline{\text{RESET}}$, STOP, and WAIT) are not in the strictest sense an interrupt; however, they are acted upon in a similar manner. See **Figure 3-1** and **Figure 3-2**. A discussion is provided below.

1. $\overline{\text{RESET}}$ - A low input on the $\overline{\text{RESET}}$ input pin causes the program to vector to its starting address which is specified by the contents of memory locations $1FFE and $1FFF. The I bit in the condition code register is also set. Much of the MCU is configured to a known state during this type of reset as previously described in **3.6 Resets**.
2. STOP - The STOP instruction causes the oscillator to be turned off and the processor to “sleep” until an external interrupt ($\overline{\text{IRQ}}$) or reset occurs.
3. WAIT - The WAIT instruction causes all processor clocks to stop, but leaves the timer clock running. This “rest” state of the processor can be cleared by reset, an external interrupt ($\overline{\text{IRQ}}$), or Timer interrupt. There are no special wait vectors for these individual interrupts.

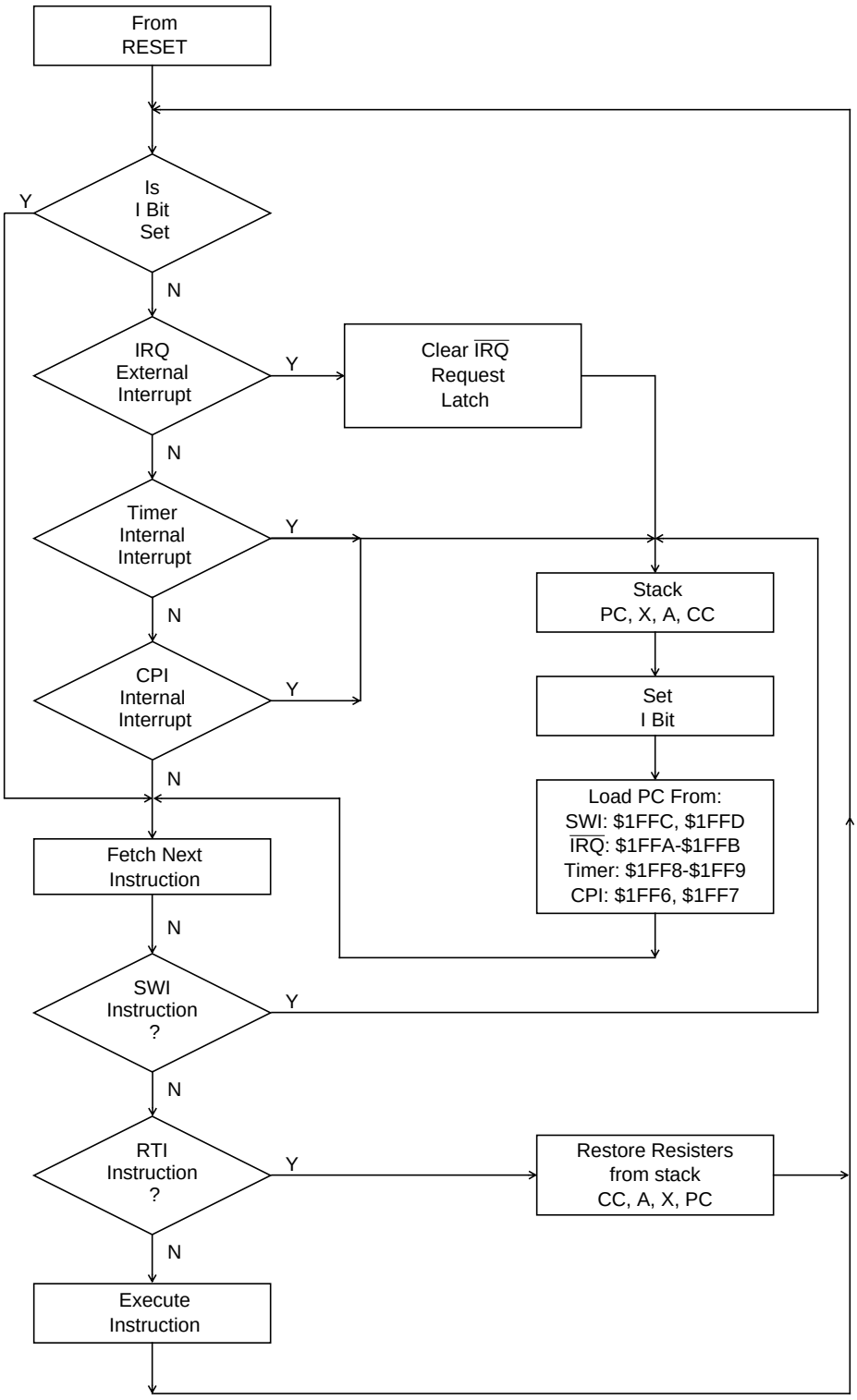


Figure 3-1. Interrupt Processing Flowchart

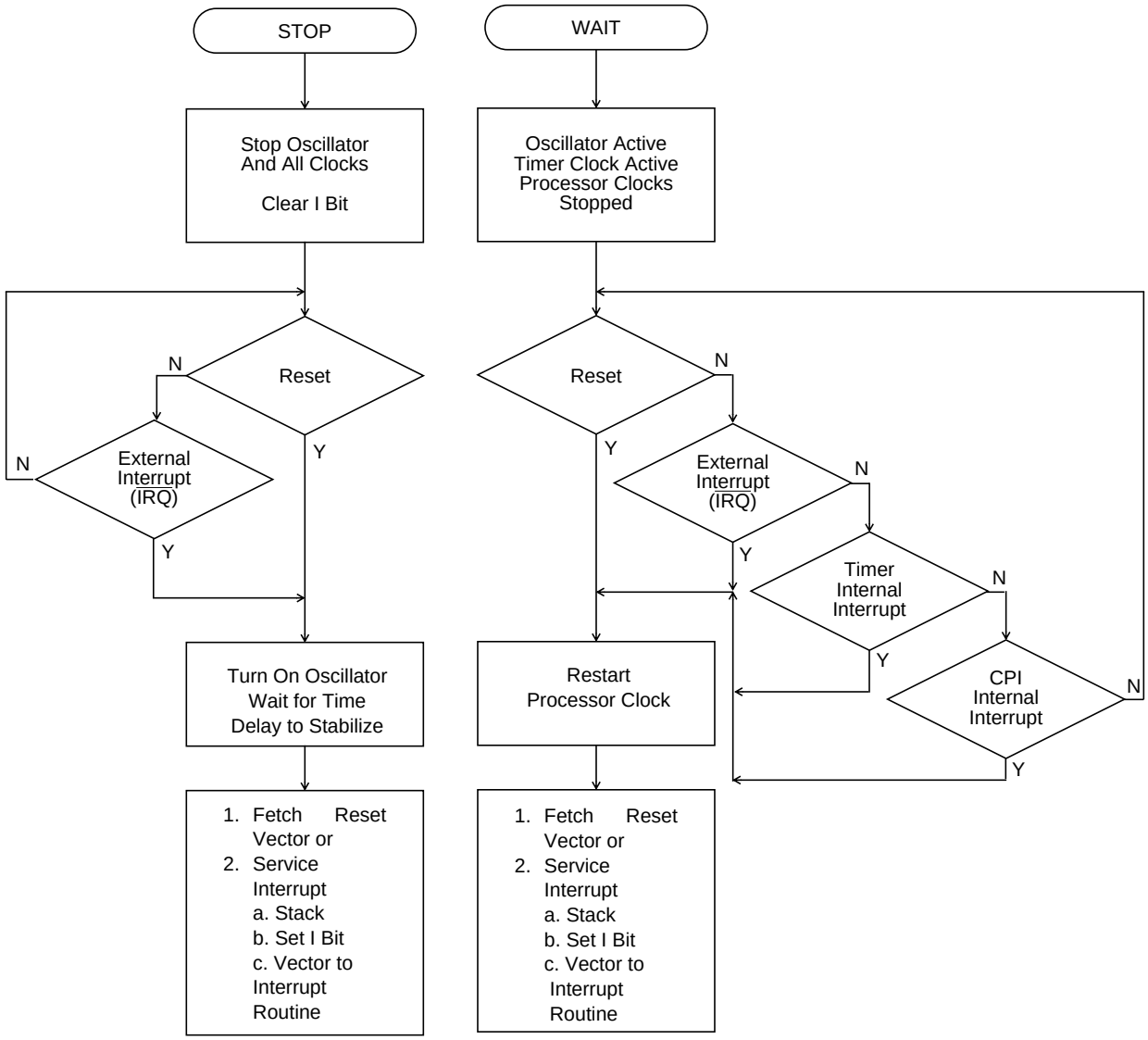


Figure 3-2. STOP/WAIT Flowcharts

3.7.2 Software Interrupt (SWI)

The SWI is an executable instruction and a non-maskable interrupt: it is executed regardless of the state of the I bit in the CCR. If the I bit is zero (interrupts enabled), SWI executes after interrupts which were pending when the SWI was fetched, but before interrupts generated after the SWI was fetched. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD.

3.7.3 External Interrupt

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 $\overline{\text{IRQ}}$. It is then synchronized internally and serviced by the interrupt service routine located at the address specified by the contents of $1FFA and $1FFB.

Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive-only trigger is available as a mask option.

NOTE: *The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I bit is cleared.*

3.7.4 Timer Interrupt

There are two different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags and enable bits are located in the Timer Control and Status Register (TCSR). Either of these interrupts will vector to the same interrupt service routine, located at the address specified by the contents of memory location $1FF8 and $1FF9. See **6.3.1 Timer Control and Status Register (TCSR) $08**.

3.7.5 Custom Peirodic Interrupt (CPI)

The CPI flag and enable bits are located in the CPI Control and Status Register (CPICSR). A CPI interrupt will vector to the interrupt service routine located at the address specified by the contents of memory location $1FF6 and $1FF7. See **6.5 Custom Periodic Interrupt**.

3.8 Low-Power Modes

3.8.1 STOP

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 (and COP Watchdog timer) operation.

The I bit in the CCR is cleared to enable external interrupts. All other registers, including the remaining bits in the TCSR, and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or $\overline{\text{RESET}}$.

The STOP instruction can be disabled by a mask option. When disabled, the STOP instruction is executed as a NOP.

See **6.6 Operation During STOP Mode**.

3.8.2 WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the timer remains active. An interrupt from the timer can cause the MCU to exit the WAIT mode.

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

See **6.7 Operation During WAIT Mode**.

3.8.3 Data-Retention Mode

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0Vdc. This is called the data-retention mode where the data is held, but the device is not guaranteed to operate. RESET must be held low during data-retention mode.

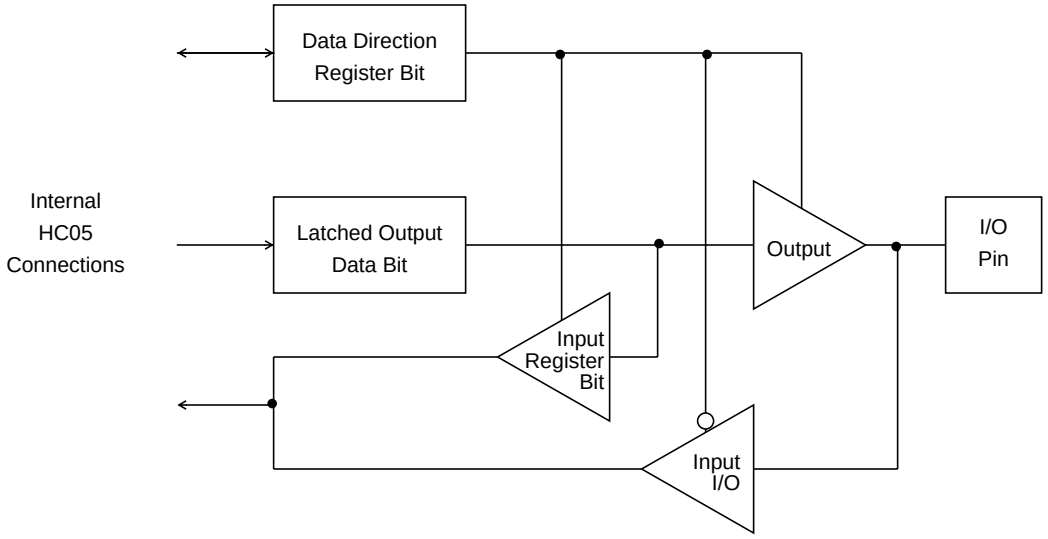


Figure 3-3. Port I/O Circuitry

CPU Core

Section 4. Input/Output Ports

4.1 Contents

4.2 Introduction

In single-chip mode there will be 20 lines arranged as two 8-bit I/O port and one 4-bit I/O port. These ports are programmable as either inputs or outputs under software control of the data direction registers.

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

4.3 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 registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode.

4.4 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 registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode.

4.5 Port C

Port C is a 4-bit bidirectional port 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 registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode.

4.6 Input/Output Programmingf

Ports A, B and C may be programmed as an input or an output 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 port has an associated DDR. Any port A, port B or port C pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero.

At power-on or reset, all DDRs are cleared, which configures all port A, B, and C 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. See **Table 4-1** and **Figure 4-1**.

Table 4-1. I/O Pin Functions

R/$\overline{W}$*	DDR	I/O Pin Function
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 of 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 jmode. The output data latch is read.

*R/$\overline{W}$ is an internal signal.

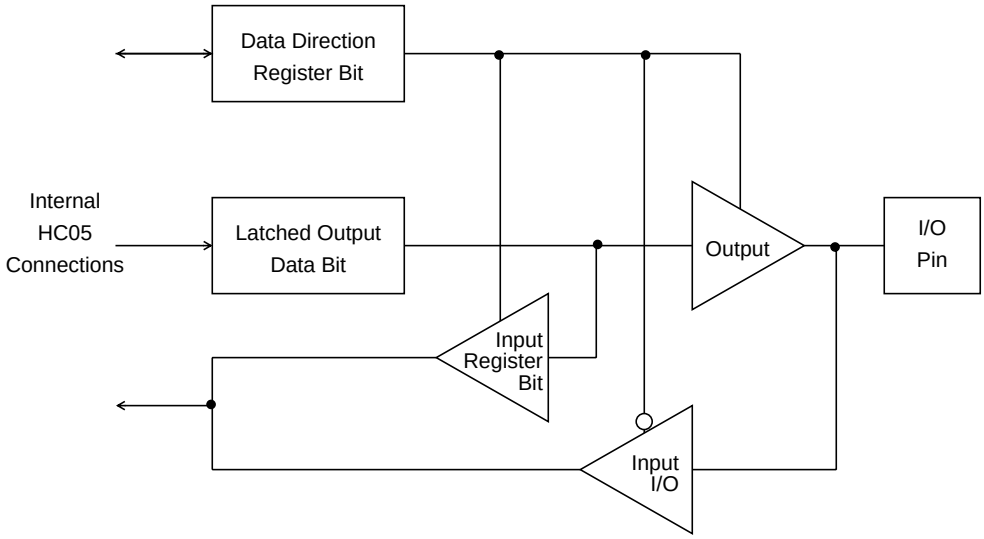


Figure 4-1. Port I/O Circuitry

Section 5. Memory

5.1 Contents

5.2 Introduction

The MC68HC05E1 has an 8K byte memory map, consisting of user ROM, user RAM, Self-Check ROM, Control Registers, and I/O. See **Figure 5-1** and **Figure 5-2**.

5.3 ROM

4096 bytes of user ROM are located from $0F00 to $1EFF, with 16 additional bytes of user vectors from $1FF0 to $1FFF. The Self-Check ROM and vectors are located from $1F00 to $1FEF.

5.4 RAM

The user RAM consists of 368 bytes from location $0090 to $01FF including the stack area. The stack begins at address $00FF. The stack pointer can access 64 bytes of RAM from $00FF to $00C0. 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.

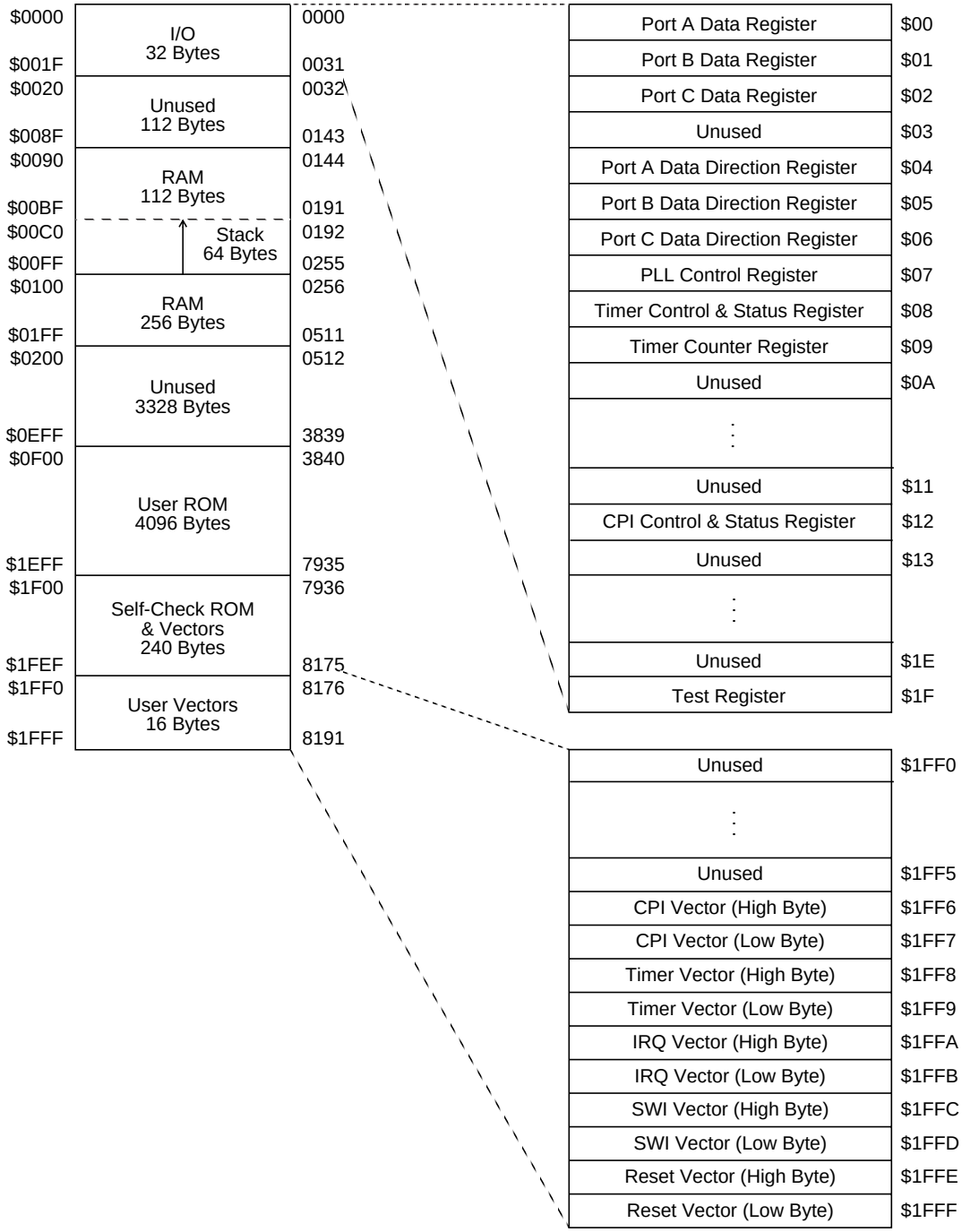


Figure 5-1. The 8 Kbyte Memory Map of the MC68HC05E1

ADDRESS $00 TO $001F	DATA							
	7	6	5	4	3	2	1	0
$00 PORT A DATA								
$01 PORT B DATA								
$02 PORT C DATA								
$03 UNUSED	0	0	0	0				
$04 PORT A DDR	—	—	—	—	—	—	—	—
$05 PORT B DDR								
$06 PORT C DDR								
$07 PLL CONTROL REG	0	BCS	AUTO	BWC	PLLON	VCOTST	PS1	PS0
$08 TIMER CONTROL & STATUS REG	TOF	RTIF	TOFE	RTIE	0	0	RT1	RT0
$09 TIMER COUNTER REG								
$0A UNUSED	—	—	—	—	—	—	—	—
$0B UNUSED	—	—	—	—	—	—	—	—
$0C UNUSED	—	—	—	—	—	—	—	—
$0D UNUSED	—	—	—	—	—	—	—	—
$0E UNUSED	—	—	—	—	—	—	—	—
$0F UNUSED	—	—	—	—	—	—	—	—
$10 UNUSED	—	—	—	—	—	—	—	—
$11 UNUSED	—	—	—	—	—	—	—	—
$12 CPI CONTROL &STATUS REG	—	CPIF	—	CPIE	—	—	—	—
$13 UNUSED	—	—	—	—	—	—	—	—
$14 UNUSED	—	—	—	—	—	—	—	—
$15 UNUSED	—	—	—	—	—	—	—	—
$16 UNUSED	—	—	—	—	—	—	—	—
$17 UNUSED	—	—	—	—	—	—	—	—
$18 UNUSED	—	—	—	—	—	—	—	—
$19 UNUSED	—	—	—	—	—	—	—	—
$1A UNUSED	—	—	—	—	—	—	—	—
$1B UNUSED	—	—	—	—	—	—	—	—
$1C UNUSED	—	—	—	—	—	—	—	—
$1D UNUSED	—	—	—	—	—	—	—	—
$1E UNUSED	—	—	—	—	—	—	—	—
$1F UNUSED	—	—	—	—	—	—	—	—

Figure 5-2. Input/Output (I/O) Registers

Section 6. Timer, Phase-Locked Loop, and Custom Periodic Interrupt

6.1 Contents

6.2 Introduction

This section describes the timer, phase-locked loop (PLL), and custom periodic interrupt (CPI).

6.3 Timer

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

As seen in **Figure 6-1**, the Timer is driven by the output of the clock select circuit (as determined by the value of BCS in the PLLCR) then a fixed divide by four prescaler. This signal drives an 8-bit ripple counter. The value of this 8-bit ripple counter can be read by the CPU at any time by accessing the Timer Counter Register (TCR) at address $09. A timer overflow function is implemented on the last stage of this counter, giving a possible interrupt at the rate of $f_{op}/1024$. Two additional stages produce the POR function at $f_{op}/4064$. The Timer Counter Bypass circuitry (available only in Test Mode) is at this point in the timer chain. This circuit is followed by two more stages, with the resulting clock ($f_{op}/16384$) driving the Real Time Interrupt circuit. The RTI circuit consists of three divider stages with a 1 of 4 selector. The output of the RTI circuit is further divided by eight to drive the mask optional COP Watchdog Timer circuit. The RTI rate selector bits, and the RTI and TOF enable bits and flags are located in the Timer Control and Status Register at location $08.

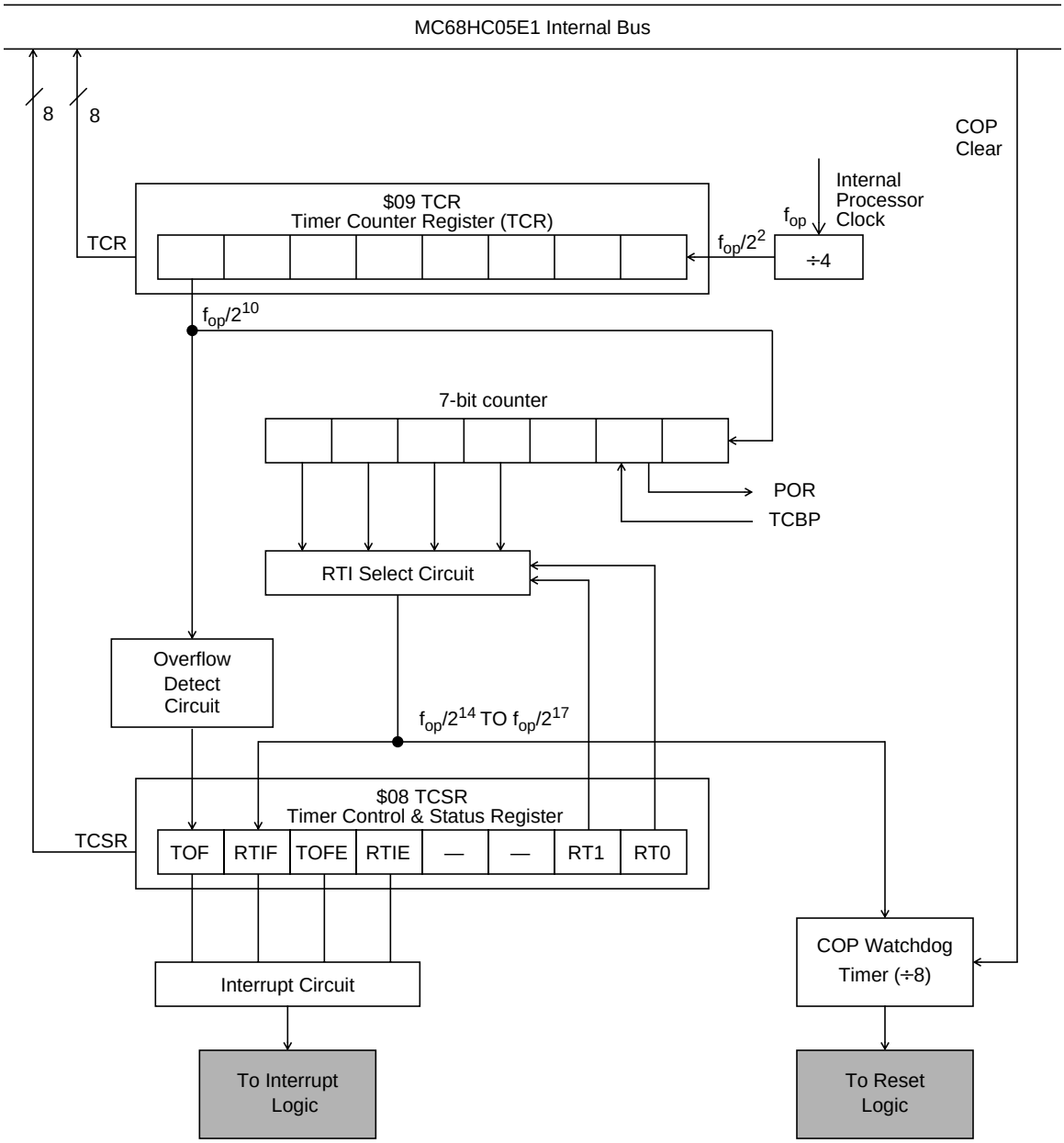


Figure 6-1. Timer Block Diagram

6.3.1 Timer Control and Status Register (TCSR) $08

The TCSR contains the timer interrupt flag, the timer interrupt enable bits, and the real time interrupt rate select bits. **Figure 6-2** shows the value of each bit in the TCSR when coming out of reset.

$08	TOF	RTIF	TOFE	RTIE	0	0	RT1	RT0
RESET:	0	0	0	0	0	0	1	1

Figure 6-2. Timer Control and Status Register (TCSR)

TOF — Timer Over Flow

TOF is a clearable, read-only status bit and is set when the 8-bit ripple counter rolls over from $FF to $00. A CPU interrupt request will be generated if TOFE is set. Clearing the TOF is done by writing a ’0’ to it. Writing a ’1’ to TOF has no effect on the bit’s value. Reset clears TOF.

RTIF — Real Time Interrupt Flag

The Real Time Interrupt circuit consists of a three stage divider and a 1 of 4 selector. The clock frequency that drives the RTI circuit is $f_{op}/2^{13}$ (or $f_{op}/8192$) with three additional divider stages giving a maximum interrupt period of 4 seconds at a crystal frequency of 32.768 kHz. RTIF is a clearable, read-only status bit and is set when the output of the chosen (1 of 4 selection) stage goes active. A CPU interrupt request will be generated if RTIE is set. Clearing the RTIF is done by writing a ’0’ to it. Writing a ’1’ to RTIF has no effect on this bit. Reset clears RTIF.

TOFE — Timer Over Flow Enable

When this bit is set, a CPU interrupt request is generated when the TOF 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.

RT1:RT0 — Real Time Interrupt Rate Select

These two bits select one of four taps from the Real Time Interrupt circuit.**Table 6-1** shows the available interrupt rates with several f_{op} values. Reset sets these RT0 and RT1, selecting the lowest periodic rate and therefore 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 6-1. RTI Rates

RT1:RT0	RTI RATES AT f_{OP} FREQUENCY SPECIFIED:					
	16.384 kHz	**524 kHz**	**1.049 MHz**	**2.097 MHz**	**4.194 MHz**	**f_{OP}**
00	1 s	31.3 ms	15.6 ms	7.8 ms	3.9 ms	2^{14} f_{op}
01	2 s	62.5 ms	31.3 ms	15.6 ms	7.8 ms	2^{15} f_{op}
10	4 s	125 ms	62.5 ms	31.3 ms	15.6 ms	2^{16} f_{op}
11	8 s	250 ms	125.1 ms	62.5 ms	31.3 ms	2^{17} f_{op}

NOTE: *In rare instances, clearing any of the timer control and status register (TCSR) flag or enable bits could result in vectoring to the reset vector rather than the timer interrupt vector if the correct precautions are not followed. Do not clear any of the timer flags or enable bits (i.e., TOF, TOFE, RTI, and RTIF) with bit manipulation instructions.*

Example:

CLEARING TIMER OVERFLOW FLAG (TOF) BIT

```
SEI     SEI NOT REQUIRED IF USED WITHIN
        TIMER INTERRUPT ROUTINE.
LDA     #$73
AND     $TCSR
OR      #$40        MASK RTIF BIT
STA     $TCSR
CLI                 DO NOT USE CLI IF THIS CODE
                    SEGMENT IF USED WITHIN TIMER
                    INTERRUPT ROUTINE
```

CLEARING TIMER OVERFLOW ENABLE (TOFE) BIT

```
SEI
LDA     #$D3
AND     $TCSR
OR      #$C0        MASK RTIF & TOF
STA     $TCSR
CLI                 DO NOT USE CLI IF THIS CODE
                    SEGMENT IF USED WITHIN TIMER
                    INTERRUPT ROUTINE
```

6.3.2 Computer Operating Properly (COP) Watchdog Reset

The COP watchdog timer function is implemented on this device by using the output of the RTI circuit and further dividing it by eight. The minimum COP reset rates are listed in **Table 6-2**. If the COP circuit times out, an internal reset is generated and the normal reset vector is fetched. Preventing a COP time-out is done by writing a ’0’ to bit 0 of address $1FF0. When the COP is cleared, only the final divide by eight stage (output of the RTI) is cleared.

This function is a mask option.

Table 6-2. COP Reset Times

RT1:RT0	RTI RATES AT f_{OP} FREQUENCY SPECIFIED:					
	16.384 kHz	524 kHz	1.049 MHz	2.097 MHz	4.194 MHz	f_{OP}
00	7 s	218.8 ms	109.4 ms	54.7 ms	27.3 ms	7 x (RTI Rate)
01	14 s	437.5 ms	218.8 ms	109.4 ms	54.7 ms	7 x (RTI Rate)
10	28 s	875.0 ms	437.5 ms	218.8 ms	109.4 ms	7 x (RTI Rate)
11	56 s	1.75 s	875.0 ms	437.5 ms	218.8 ms	7 x (RTI Rate)

6.3.3 Timer Control Register (TCR) $09

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

Figure 6-3. Timer Counter Register

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

6.4 Phase-Locked Loop Synthesizer

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 uf) 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.

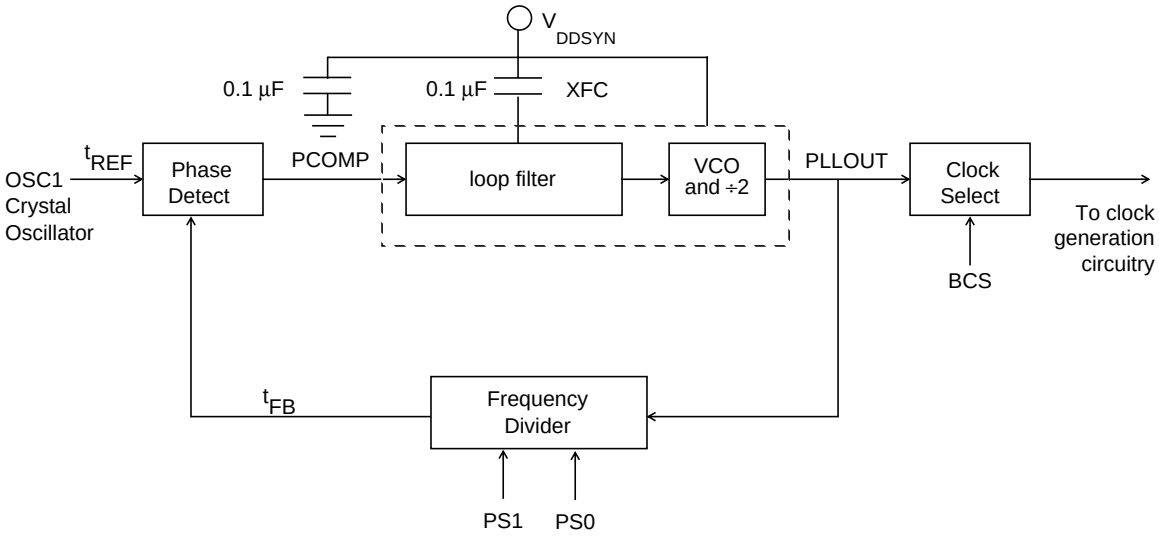


Figure 6-4. PLL Circuit

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 6-4**. The output wave-form 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 256, 128, 64, or 32 to provide the feedback frequency for the phase detector.

To change PLL frequencies, follow the procedure outlined below:

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},
6. Set BCS to switch to the high frequency bus rate.

The user should not 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.

The PLL loop filter has two bandwidths which are automatically selected by the PLL if AUTO=1. 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. **The use of automatic bandwidth is not recommended at this time.** Manual bandwidth control can be done by clearing AUTO in the PLLCR and setting the appropriate value for BWC.

6.4.1 Phase-Locked Loop Control Register (PLLCR) $07

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

$07	0	BCS	AUTO	BWC	PLLON	VCOTST	PS1	PS0
RESET:	0	0	1	0	1	1	0	1

Figure 6-5. Phase-Locked Loop Control Register

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. Reset clears this bit.

AUTO

When set, this bit selects the automatic bandwidth circuitry in the Phase detect block. When clear, manual bandwidth control is selected. Reset sets this bit.

NOTE: *The use of automatic bandwidth is not recommended at this time.*

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. Bandwidth control is under manual control only when the AUTO bit is clear. When the AUTO bit is set, BWC acts as a read-only status bit to indicate which mode has been selected by the internal circuit. On PLL start-up in automatic mode (AUTO=1), the high bandwidth driver is enabled (BWC=1) by internal circuitry until the PLL has locked onto

the specified frequency. The high bandwidth driver is then disabled and BWC is cleared by internal circuitry. Reset clears this bit. Table 6-3

Table 6-3. Loop Filter Bandwidth Control

AUTO	BWC	VCOTST	HIGH BANDWIDTH	LOW BANDWIDTH
0	0	0	OFF	OFF
0	0	1	OFF	ON
0	1	0	ON	OFF
0	1	1	ON	ON
1	X	1	AUTO	ON

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. Reset sets this bit.

NOTE: *PLLON should not be cleared while using the PLL to drive the internal processor clock, i.e. when BCS is high. If the internal processor clock is driven by the PLL, clearing the PLLON bit would cause the internal processor clock to stop. Exercise caution when using these bits.*

VCOTST — VCO Test

This bit is used to isolate the loop filter from the VCO in order to facilitate testing. When clear, the low bandwidth mode of the PLL filter is disabled. When set, the loop filter operates as indicated by the values of AUTO and BWC. This bit is always set when AUTO=1 as security when running in automatic mode. Reset sets this bit.

NOTE: *This bit is intended for use by Motorola to test and characterize the PLL. The user should always have this bit set to 1.*

PS1: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. These bits should not be written if BCS in the PLLCR is at a logic high. Reset clears PS1 and sets PS0, choosing a bus clock frequency of 1.049 MHz.

Table 6-4. PS1 and PS0 Speed Selects with 32.768 kHz Crystal

PA1:PS0	CPU BUS CLOCK FREQUENCY (f_{OP})	
0 0	524 kHz	
0 1	1.049 MHz	Reset Condition
1 0	2.097 MHz	See Note below
1 1	4.194 MHz	See Note below

NOTE: *For the standard MC68HC05E1, the 4.194 MHz bus clock frequency should never be selected, and the 2.097 MHz bus clock frequency should not be selected when running the part below V_{DD} = 4.5 V. For the high speed MC68HSC05E1, the 4.194 MHz bus clock frequency should not be selected when running the part below V_{DD} = 4.5 V.*

6.4.2 Operation During STOP Mode

The PLL is switched to low frequency bus rate and is temporarily turned off 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.

6.4.3 Noise Immunity

The MCU should be insulated as much as possible from noise in the system. We recommend the following steps be taken 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, and
3. All power pins should be filtered (to minimize noise on these signals) 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.

6.5 Custom Periodic Interrupt

The custom periodic interrupt is mask programmable to a 0.25 second, 0.5 second, or 1 second interrupt. The interrupt is generated from the 32 kHz OSC1 input by a 15-bit counter. This interrupt is under the control of the Custom Periodic Interrupt Control and Status Register located at $12.

6.5.1 Custom Periodic Interrupt Control and Status Register (CPICSR) $12

The CPICSR contains the CPI flag and enable bits. **Figure 6-6** shows the location of these bits and their values after reset.

$12	0	CPIF	0	CPIE	0	0	0	0
RESET:	0	0	0	0	0	0	0	0

Figure 6-6. Custom Periodic Interrupt Control and Status Register (CPICSR)

CPIF — Custom Periodic Interrupt Flag

CPIF is a clearable, read-only status bit and is set when the 15-bit counter changes from $7FFF to $0000. A CPU interrupt request will be generated if CPIE is set. Clearing the CPIF is done by writing a ’0’ to it. Writing a ’1’ to CPIF has no effect on the bit’s value. Reset clears CPIF.

CPIE — Custom Periodic Interrupt Enable

When this bit is cleared, the counter is cleared and CPI interrupts are disabled. When this bit is set, the counter starts from $0000 and a CPU interrupt request is generated when the CPIF bit is set. Reset clears this bit.

6.6 Operation During STOP Mode

The timer system is cleared and the CPI counter is halted when going into STOP mode. When STOP is exited by an external interrupt or an external RESET, the internal oscillator will resume, followed by a 4064 internal processor oscillator stabilization delay. The timer system counter is then cleared and operation resumes. The CPI will continue counting once the oscillator resumes and does not wait for the oscillator to stabilize.

6.7 Operation During WAIT Mode

The CPU clock halts during the WAIT mode, but the timer and CPI remain active. If interrupts are enabled, a timer interrupt or custom periodic interrupt will cause the processor to exit the WAIT mode.

Section 7. Electrical Specifications

7.1 Contents

7.2 Introduction

This section provides parametric information for the MC68HC05E1.

7.3 Maximum Ratings

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

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

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

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to* ***7.6 5.0-Volt DC Electrical Characteristics*** *and* ***7.7 3.3-Volt DC Electrical Characteristics*** *for guaranteed operating conditions.*

7.4 Operating Range

Characteristic	Symbol	Value	Unit
Operating Temperature Range MC68HC05E1P (Standard) MC68HC05E1CP (Extended) MC68HC05E1MP (Automotive)	T_A	T_L to T_H 0 to +70 –40 to +85 –40 to +125	°C

7.5 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic DIP SOIC	θ_{JA}	 60 60	°C/W

7.6 5.0-Volt DC Electrical Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage I_{Load} = 10.0 µA I_{Load} = –10.0 µA	 V_{OL} V_{OH}	 — V_{DD} –0.1	 — —	 0.1 —	V
Output High Voltage (I_{Load} = –0.8 mA) PA0–PA7, PB0–PB7, PC0–PC3	V_{OH}	V_{DD} –0.8	—	—	V
Output Low Voltage (I_{Load} = 1.6 mA) PA0–PA7, PB0–PB7, PC0–PC3	V_{OL}	—	—	0.40	V
Input High Voltage PA0–PA7, PB0–PB7, PD0–PD3, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0–PA7, PB0–PB7, PD0–PD3, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
XFC Wide Bandwidth Source Sink	 I_{OH} I_{OL}	 –50 50	 –100 100	 — —	µA
XFC Narrow Bandwidth Source Sink	 I_{OH} I_{OL}	 –1 1	 –2 2	 — —	µA
Supply Current (see Notes) Run f_{osc} = 32.768 kHz, f_{OP} =16.384 kHz f_{osc} = 4.2 MHz, f_{OP} = 2.1 MHz Wait f_{osc} = 32.768 kHz, f_{OP} =16.384 kHz f_{osc} = 4.2 MHz, f_{OP} = 2.1 MHz Stop (PLL off) 25 °C –40 °C to +85 °C (Extended)	 I_{DD} 	 — — — — — —	 100 3.5 60 0.8 2 —	 160 5.0 100 1.2 50 180	 µA mA µA mA µA µA
I/O Ports Hi-Z Leakage Current PB0–PB7, PC0–PC3, PA0–PA7	I_{OZ}	—	—	10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1	I_{IN}	—	—	1	µA
Capacitance Ports (As Input or Output) $\overline{RESET}$, $\overline{IRQ}$	 C_{OUT} C_{IN}	 — —	 — —	 12 8	pF

NOTES:

1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = 0 °C to 70 °C, unless otherwise noted
2. All values shown reflect average measurements at midpoint of voltage range at 25 °C.
3. Wait I_{DD}: Only timer and CPI systems active
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source, all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 Vdc, V_{IH} = V_{DD} –0.2 Vdc.
6. Stop I_{DD} is measured with OSC1 = V_{SS}.
7. Standard temperature range is 0 °C to 70 °C. Extended temperature range (–40 °C to 85 °C) is available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

7.7 3.3-Volt DC Electrical Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage I_{Load} = 10.0 µA I_{Load} = –10.0 µA	 V_{OL} V_{OH}	 — V_{DD} –0.1	 — —	 0.1 —	V
Output High Voltage (I_{Load} = –0.2 mA) PA0–PA7, PB0–PB7, PC0–PC3	V_{OH}	V_{DD} –0.3	—	—	V
Output Low Voltage (I_{Load} = 0.4 mA) PA0–PA7, PB0–PB7, PC0–PC3	V_{OL}	—	—	0.30	V
Input High Voltage PA0–PA7, PB0–PB7, PD0–PD3, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0–PA7, PB0–PB7, PD0–PD3, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
XFC Wide Bandwidth Source Sink	 I_{OH} I_{OL}	 –25 25	 –50 50	 — —	µA
XFC Narrow Bandwidth Source Sink	 I_{OH} I_{OL}	 –-0.5 0.5	 –1 1	 — —	µA
Supply Current (see Notes) Run f_{osc} = 32.768 kHz, f_{OP} =16.384 kHz f_{osc} = 2.1 MHz, f_{OP} = 1.0 MHz Wait f_{osc} = 32.768 kHz, f_{OP} =16.384 kHz f_{osc} = 2.1 MHz, f_{OP} = 1.0 MHz Stop (PLL off) 25 °C –40 °C to +85 °C (Extended)	 I_{DD} 	 — — — — — —	 60 1.5 30 0.3 1 —	 90 2.0 50 0.3 30 120	 µA mA µA mA µA µA
I/O Ports Hi-Z Leakage Current PB0–PB7, PC0–PC3, PA0–PA7	I_{OZ}	—	—	10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1	I_{IN}	—	—	1	µA
Capacitance Ports (As Input or Output) $\overline{RESET}$, $\overline{IRQ}$	 C_{OUT} C_{IN}	 — —	 — —	 12 8	pF

NOTES:

1. V_{DD} = 3.3 Vdc ±10%, V_{SS} = 0 Vdc, T_A = 0 °C to 70 °C, unless otherwise noted
2. All values shown reflect average measurements at midpoint of voltage range at 25 °C.
3. Wait I_{DD}: Only timer and CPI systems active
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source, all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 Vdc, V_{IH} = V_{DD} –0.2 Vdc.
6. Stop I_{DD} is measured with OSC1 = V_{SS}.
7. Standard temperature range is 0 °C to 70 °C. Extended temperature range (–40 °C to 85 °C) is available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

7.8 5.0-Volt Control Timing

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Oscillator Option External Clock Option	f_{osc}	 — DC	 32.768 4.2	 kHz MHz
Internal Operating Frequency Crystal Oscillator ($f_{OSC} \div 2$) External Clock ($f_{OSC} \div 2$)	f_{op}	 — DC	 16.384 2.1	 kHz MHz
Cycle Time	t_{cyc}	480	—	ns
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see **Figure 7-1**)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (see **Figure 7-1**)	t_{ILIL}	Note 2	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns
PLL Startup Stabilization Time	t_{PLLS}	50	—	ms

NOTES:
1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0 °C to 70 °C, unless otherwise note
2. The minimum period, t_{ILIL}, should not be less than the number of cycles it takes to execute the interrupt service routine plus 19 t_{cyc}.

7.9 3.3-Volt Control Timing

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Oscillator Option External Clock Option	f_{osc}	 — DC	 32.768 2.1	 kHz MHz
Internal Operating Frequency Crystal Oscillator ($f_{OSC} \div 2$) External Clock ($f_{OSC} \div 2$)	f_{op}	 — DC	 16.384 1.0	 kHz MHz
Cycle Time	t_{cyc}	1000	—	ns
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see **Figure 7-1**)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (see **Figure 7-1**)	t_{ILIL}	Note 2	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns
PLL Startup Stabilization Time	t_{PLLS}	100	—	ms

NOTES:
1. V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0 °C to 70 °C, unless otherwise note
2. The minimum period, t_{ILIL}, should not be less than the number of cycles it takes to execute the interrupt service routine plus 19 t_{cyc}.

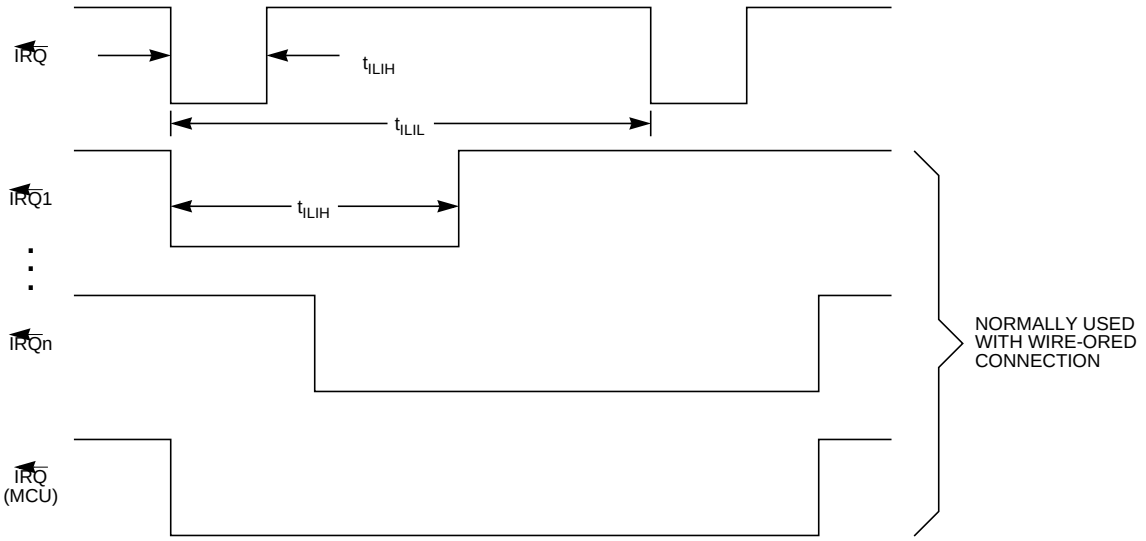
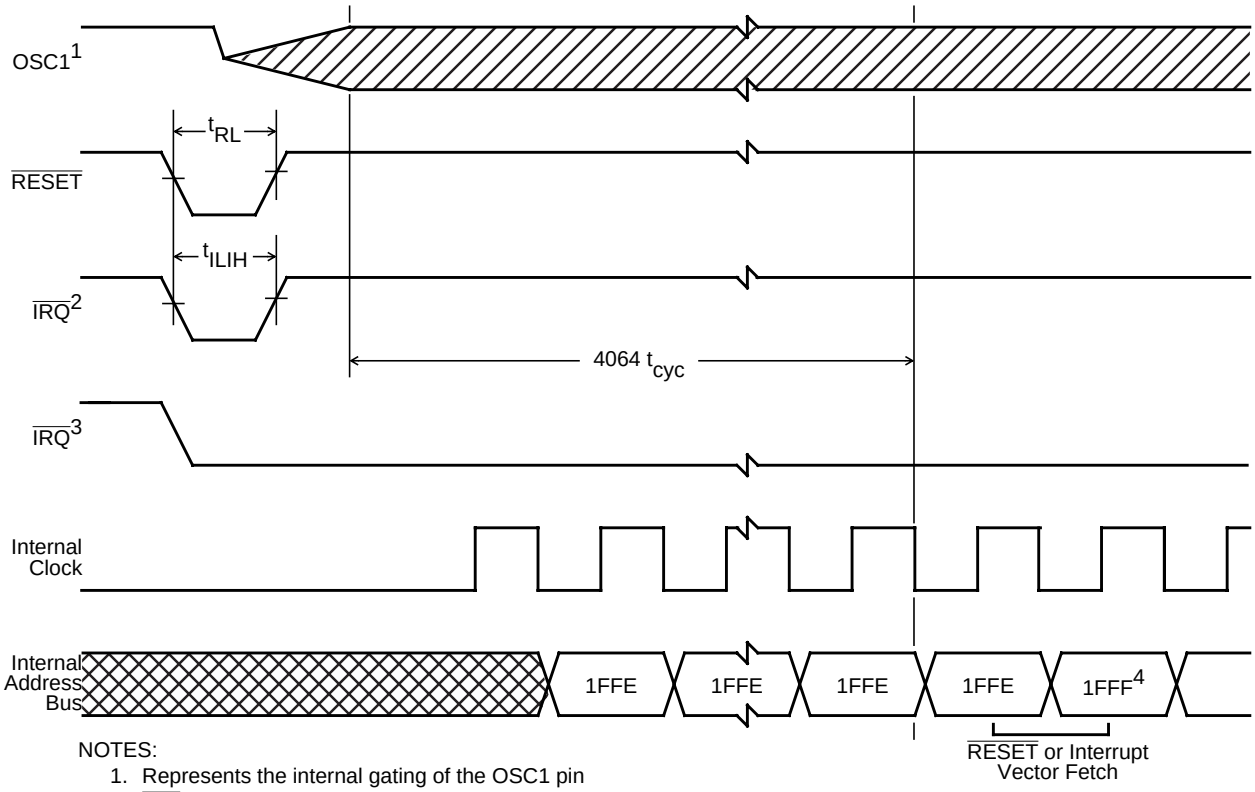
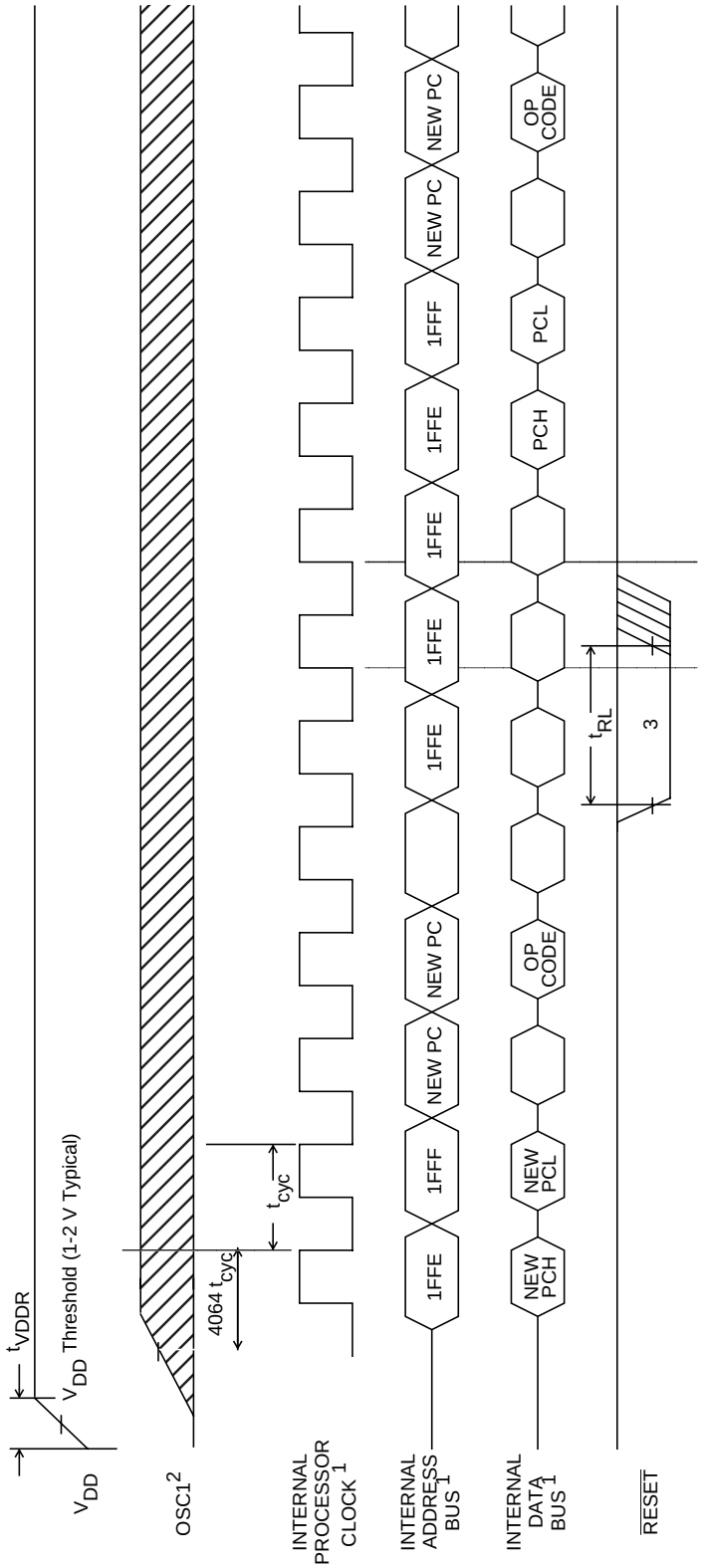


Figure 7-1. External Interrupt Mode Diagram



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

Figure 7-2. Stop Recovery Timing Diagram



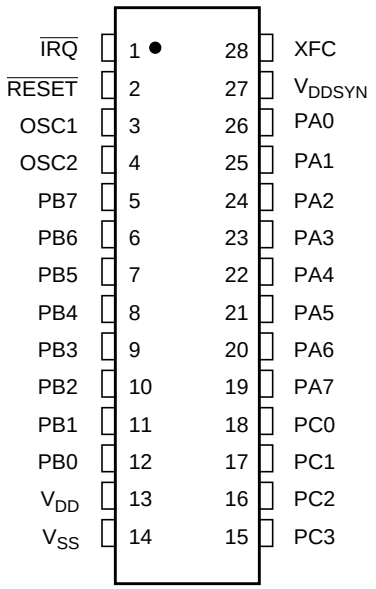
NOTES:
1. Internal timing signal and bus information not available externally.
2. OSC1 line is not meant to represent frequency. It is only used to represent time.
3. The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.

Figure 7-3. Power-On Reset and RESET

Section 8. Mechanical Specifications

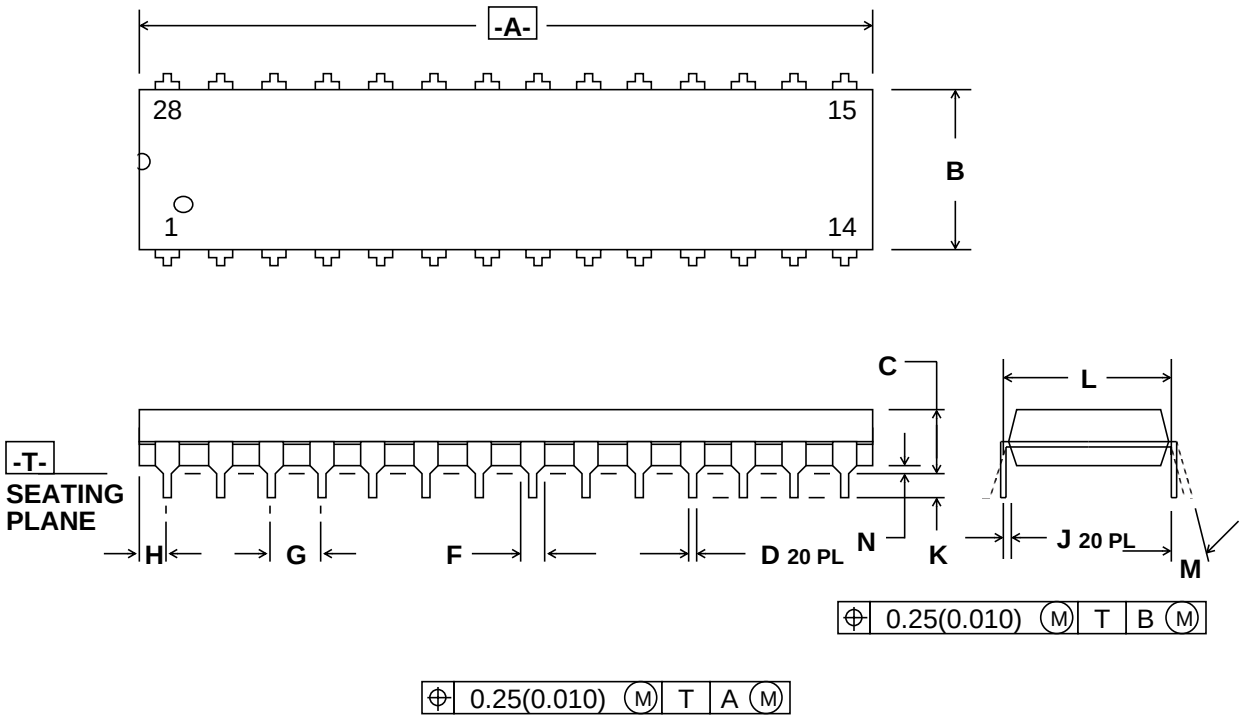
8.1 Contents

8.2 Mechnical Data



8.3 Package Dimensions

8.3.1 P Suffix, Plastic DIP, Case # 710-02

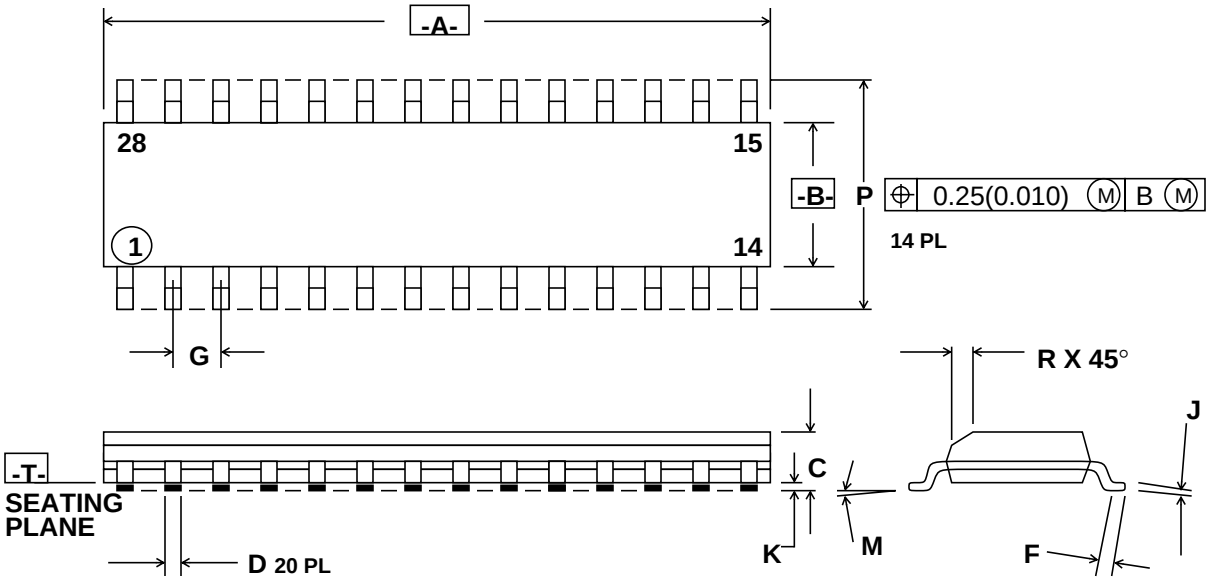


NOTES

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

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.2	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

8.3.2 DW Suffix, SOIC, Case # 751F-02



NOTES

1. DIMENSIONS "A" AND "B" ARE DATUMS AND "T" IS A DATUM SURFACE.
2. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
3. CONTROLLING DIM; MILLIMETER.
4. DIMENSION A AND B DO NOT INCLUDE MLD PROTRUSION.
5. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
6. 751F-01 OBSOLETE, NEW STANDARD 751F-02.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	17.80	18.05	0.701	0.710
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	0.50	0.90	0.020	0.035
G	1.27 BSC		0.050BSC	
J	0.25	0.32	0010	0.012
K	0.10	0.25	0.004	0.009
M	0°	7°	0°	7°
P	10.05	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

Mechanical Specifications

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com
Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com
Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com
Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

RoHS-compliant and/or Pb- free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb- free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale.s Environmental Products program, go to http://www.freescale.com/epp.